



# **Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide**

## **Technical Note**

FPGA-TN-02163-2.8

February 2023

## Disclaimers

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

## Contents

1. Introduction .....	7
1.1. EFB Register Map .....	8
1.2. WISHBONE Bus Interface .....	8
1.3. WISHBONE Write Cycle .....	9
1.4. WISHBONE Read Cycle .....	10
1.5. WISHBONE Reset Cycle .....	12
2. Hardened I <sup>2</sup> C IP Cores .....	12
3. I <sup>2</sup> C Registers .....	12
4. Typical I <sup>2</sup> C Transactions .....	19
5. I <sup>2</sup> C Functional Waveforms .....	21
6. I <sup>2</sup> C Timing Diagram .....	25
7. I <sup>2</sup> C Simulation Model .....	25
8. Hardened SPI IP Core .....	29
9. SPI Registers .....	29
10. Typical SPI Transactions .....	39
11. SPI Functional Waveforms .....	40
12. SPI Timing Diagrams .....	41
13. SPI Simulation Model .....	43
14. Hardened Timer/Counter PWM .....	44
14.1. Timer/Counter Registers .....	44
15. Timer Counter Simulation Model .....	52
16. Flash Memory (UFM/Configuration) Access .....	53
16.1. Flash Memory (UFM/Configuration) Access Ports .....	53
16.2. Flash Memory (UFM/Configuration) Access through WISHBONE Slave Interface .....	54
16.3. Command and Data Transfers to Flash Memory (UFM/Configuration) Space .....	59
16.4. Command Summary by Application .....	59
16.5. Command Descriptions by Command Code .....	61
17. Interface to Configuration Flash .....	72
18. Command Framing .....	74
18.1. I <sup>2</sup> C Framing .....	74
18.2. SPI Framing .....	74
18.3. WISHBONE Framing .....	75
19. UFM Write and Read Examples .....	76
20. Flash Memory Erase and Program Performance .....	79
21. Erase/Program/Verify Time Calculation Example .....	79
Technical Support Assistance .....	80
Revision History .....	81

## Figures

Figure 1.1. Embedded Function Block (EFB) .....	7
Figure 1.2. WISHBONE Bus Interface Between the FPGA Core and the EFB Module .....	8
Figure 1.3. WISHBONE Bus Write Operation .....	10
Figure 1.4. WISHBONE Bus Read Operation .....	11
Figure 1.5. EFB WISHBONE Interface Reset .....	12
Figure 3.1. I2C Master Read/Write Example (via WISHBONE) .....	18
Figure 3.2. I2C Slave Read/Write Example (via WISHBONE) .....	19
Figure 4.1. Simple I2C Command (for example, ISC_ERASE) .....	19
Figure 4.2. I2C Command with Write Data (for example, LSC_PROG_INCR_NV) .....	20
Figure 4.3. I2C Command with Read Data (for example, LSC_READ_STATUS) .....	20
Figure 5.1. EFB Master – I2C Write .....	21
Figure 5.2. EFB Master – I2C Read .....	22
Figure 5.3. EFB Slave – I2C Write .....	23
Figure 5.4. EFB Slave – I2C Read .....	24
Figure 6.1. I2C Bit Transfer Timing .....	25
Figure 9.1. SPI Master Read/Write Example (via WISHBONE) – Production Silicon .....	36
Figure 9.2. SPI Master Read/Write Example (via WISHBONE) – R1 Silicon .....	37
Figure 9.3. SPI Slave Read/Write Example (via WISHBONE) – Production Silicon .....	38
Figure 10.1. Simple SPI Command (for example, ISC_ERASE) .....	39
Figure 10.2. SPI Command w/ Write Data (for example, LSC_PROG_INCR_NV) .....	39
Figure 10.3. SPI Command w/ Read Data (for example, LSC_READ_STATUS) .....	39
Figure 11.1. Fully Specified SPI Transaction (MachXO2 as SPI Master or Slave) .....	40
Figure 11.2. Minimally Specified SPI Transaction Example (MachXO2 as SPI Slave) .....	40
Figure 12.1. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0) .....	41
Figure 12.2. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0) .....	41
Figure 12.3. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1) .....	42
Figure 12.4. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1) .....	42
Figure 12.5. Slave SPI Dummy Byte Response (SPICR2[SDBRE]) Timing .....	42
Figure 16.1. Interfaces to the UFM/Configuration Sectors .....	54
Figure 17.1. Basic Configuration Flash Update Example .....	73
Figure 18.1. I2C Read Device ID Example .....	74
Figure 18.2. SSPI Read Device ID Example .....	75
Figure 18.3. WISHBONE Read Device ID Example (-1200 HC Device) .....	75

## Tables

Table 1.1. EFB Register Map .....	8
Table 1.2. WISHBONE Slave Interface Signals of the EFB Module .....	9
Table 3.1. I <sup>2</sup> C Registers .....	13
Table 3.2. I <sup>2</sup> C Control (Primary/Secondary) .....	13
Table 3.3. I <sup>2</sup> C Command (Pri/Sec) .....	14
Table 3.4. I <sup>2</sup> C Clock Prescale 0 (Primary/Secondary) .....	14
Table 3.5. I <sup>2</sup> C Register Clock Prescale 1 (Primary/Secondary) .....	14
Table 3.6. I <sup>2</sup> C Transmit Data Register (Primary/Secondary) .....	15
Table 3.7. I <sup>2</sup> C Status (Primary/Secondary) .....	15
Table 3.8. I <sup>2</sup> C General Call Data Register (Primary/Secondary) .....	16
Table 3.9. I <sup>2</sup> C Receive Data Register (Primary/Secondary) .....	16
Table 3.10. I <sup>2</sup> C Interrupt Status (Primary/Secondary) .....	16
Table 3.11. I <sup>2</sup> C Interrupt Enable (Primary/Secondary) .....	17
Table 7.1. I <sup>2</sup> C Primary Simulation Mode .....	25
Table 7.2. I <sup>2</sup> C Secondary Simulation Model .....	27
Table 9.1. SPI Registers .....	29
Table 9.2. SPI Control 0 .....	30
Table 9.3. SPI Control 1 .....	30
Table 9.4. SPI Control 2 .....	31
Table 9.5. SPI Clock Prescale .....	32
Table 9.6. SPI Master Chip Select .....	33
Table 9.7. SPI Transmit Data Register .....	33
Table 9.8. SPI Status .....	33
Table 9.9. SPI Receive Data Register .....	34
Table 9.10. SPI Interrupt Status .....	34
Table 9.11. SPI Interrupt Enable .....	35
Table 13.1. SPI Simulation Model .....	43
Table 14.1. Timer/Counter Registers .....	45
Table 14.2. Timer/Counter Control 0 .....	45
Table 14.3. Timer/Counter Control 1 .....	46
Table 14.4. Timer/Counter Set Top Counter Value 0 .....	47
Table 14.5. Timer/Counter Set Top Counter Value 1 .....	47
Table 14.6. Timer/Counter Set Compare Counter Value 0 .....	47
Table 14.7. Timer/Counter Set Compare Counter Value 1 .....	47
Table 14.8. Timer/Counter Control 2 .....	47
Table 14.9. Timer/Counter Counter Value 0 .....	48
Table 14.10. Timer/Counter Counter Value 1 .....	48
Table 14.11. Timer/Counter Current Top Counter Value 0 .....	48
Table 14.12. Timer/Counter Current Top Counter Value 1 .....	48
Table 14.13. Timer/Counter Current Compare Counter Value 0 .....	49
Table 14.14. Timer/Counter Current Compare Counter Value 1 .....	49
Table 14.15. Timer/Counter Current Capture Counter Value 0 .....	49
Table 14.16. Timer/Counter Current Capture Counter Value 1 .....	49
Table 14.17. Timer/Counter Status Register .....	50
Table 14.18. Timer/Counter Interrupt Status .....	50
Table 14.19. Timer/Counter Interrupt Enable .....	51
Table 15.1. Timer/Counter Simulation Mode .....	52
Table 16.1. WISHBONE to Flash Memory (CFG) Logic Registers .....	55
Table 16.2. Flash Memory (UFM/Configuration) Control .....	55
Table 16.3. Flash Memory (UFM/Configuration) Transmit Data .....	55
Table 16.4. Flash Memory (UFM/Configuration) Status .....	55
Table 16.5. Flash Memory (UFM/Configuration) Receive Data .....	56

Table 16.6. Flash Memory (UFM/Configuration) Interrupt Status .....	57
Table 16.7. Flash Memory (UFM/Configuration) Interrupt Enable .....	57
Table 16.8. Unused (Reserved) Register .....	58
Table 16.9. EFB Interrupt Source .....	58
Table 16.10. UFM (Sector 1) Commands .....	59
Table 16.11. Configuration Flash (Sector 0) Commands .....	60
Table 16.12. Non-Volatile Register (NVR) Commands .....	61
Table 16.13. Erase Flash (0x0E) .....	61
Table 16.14. Read TraceID Code (0x19) .....	62
Table 16.15. Disable Configuration Interface (0x26) .....	62
Table 16.16. Read Status Register (0x3C) .....	62
Table 16.17. Reset CFG Address (0x46) .....	63
Table 16.18. Reset UFM Address (0x47) .....	63
Table 16.19. Program DONE (0x5E) .....	63
Table 16.20. Program Configuration Flash (0x70) .....	63
Table 16.21. Read Configuration Flash (0x73) (SPI) .....	64
Table 16.22. Read Configuration Flash (0x73) (I <sup>2</sup> C/SPI) .....	64
Table 16.23. Read Configuration Flash (0x73) (WISHBONE) .....	64
Table 16.24. Enable Configuration Interface (Transparent) (0x74) .....	65
Table 16.25. Refresh (0x79) .....	65
Table 16.26. STANDBY (0x7D) .....	66
Table 16.27. Set Address (0xB4) .....	66
Table 16.28. Read USERCODE (0xC0) .....	67
Table 16.29. Program USERCODE (0xC2) .....	67
Table 16.30. Enable Configuration Interface (Offline) (0xC6) .....	67
Table 16.31. Program UFM (0xC9) .....	67
Table 16.32. Read UFM (0xCA) (SPI) .....	68
Table 16.33. Read UFM (0xCA) (I <sup>2</sup> C/SPI) .....	68
Table 16.34. Read UFM (0xCA) (WISHBONE) .....	68
Table 16.35. Erase UFM (0xCB) .....	69
Table 16.36. Program SECURITY (0xCE) .....	69
Table 16.37. Program SECURITY PLUS (0xCF) .....	69
Table 16.38. Read Device ID Code (0xE0) .....	70
Table 16.39. Device ID Table .....	70
Table 16.40. Verify Device ID Code (0xE2) .....	70
Table 16.41. Program Feature Row (0xE4) .....	70
Table 16.42. Read Feature Row (0xE7) .....	70
Table 16.43. Check Busy Flag (0xF0) .....	71
Table 16.44. Program FEABITs (0xF8) .....	71
Table 16.45. Read FEABITs (0xFB) .....	72
Table 16.46. Bypass (Null Operation) (0xFF) .....	72
Table 18.1. Command Framing Protocol, by Interface .....	74
Table 18.2. Command Framing Protocol, by Interface .....	74
Table 18.3. Command Framing Protocol, by Interface .....	75
Table 19.1. Write Two UFM Pages .....	76
Table 19.2. Read One UFM Page (All Devices, WISHBONE/SPI) .....	77
Table 19.3. Read Two UFM Pages (WISHBONE/SPI) .....	78
Table 20.1. Flash Memory (UFM/Configuration) Performance in MachXO2 Devices <sup>1</sup> .....	79
Table 21.1. E/P/V Calculation parameters .....	79

# 1. Introduction

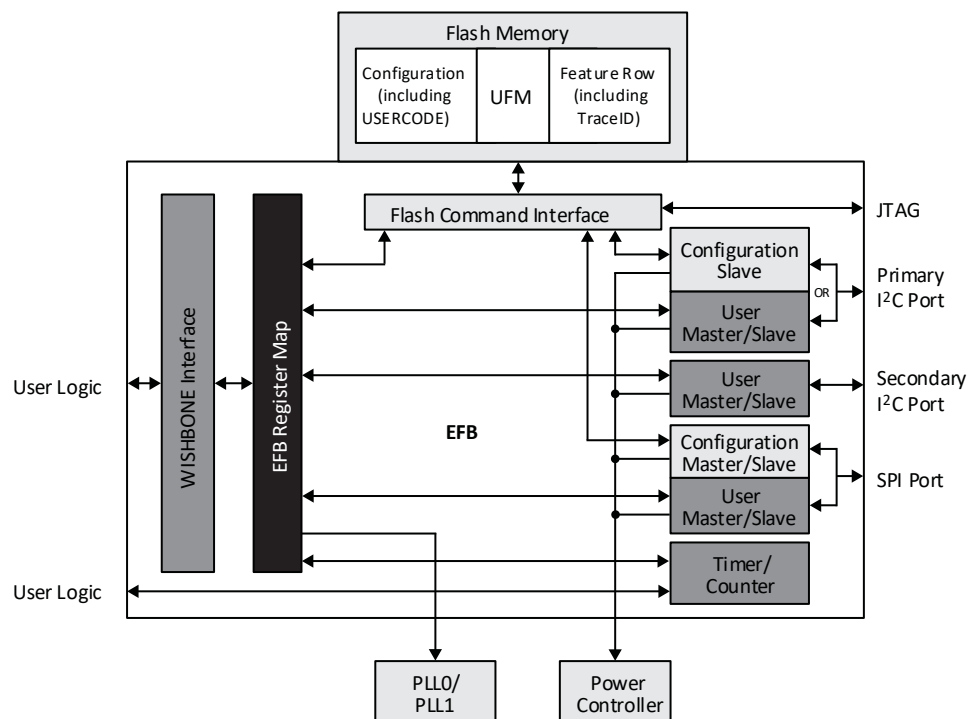
This reference guide supplements [Using User Flash Memory and Hardened Control Functions in MachXO2™ Devices Usage Guide \(FPGA-TN-02162\)](#) which explains the software usage. In this document you can find:

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

As an overview, the MachXO2 FPGA family combines a high-performance, low power, FPGA fabric with built-in, hardened control functions and on-chip User Flash Memory (UFM). 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 MachXO2 devices include an EFB module. The EFB block includes the following control functions:

- Two I2C Cores
- One SPI Core
- One 16-bit Timer/Counter
- Interface to Flash Memory which includes:
  - User Flash Memory for MachXO2-640 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)**

## 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 read/write operations. Each hardened function has dedicated 8-bit Data and Control registers, with the exception of the Flash Memory (UFM/Configuration), which are accessed through the same set of registers. [Table 1.1](#) documents the register map of the EFB module. The PLL registers are located in the MachXO2 PLL modules, but they are accessed through EFB WISHBONE read/write cycles.

**Table 1.1. EFB Register Map**

Address (Hex)	Hardened Function
0x00-0x1F	PLL0 Dynamic Access <sup>1</sup>
0x20-0x3F	PLL1 Dynamic Access <sup>1</sup>
0x40-0x49	I <sup>2</sup> C Primary
0x4A-0x53	I <sup>2</sup> C Secondary
0x54-0x5D	SPI
0x5E-0x6F	Timer/Counter
0x70-0x75	Flash Memory (UFM/Configuration)
0x76-0x77	EFB Interrupt Source

**Note:**

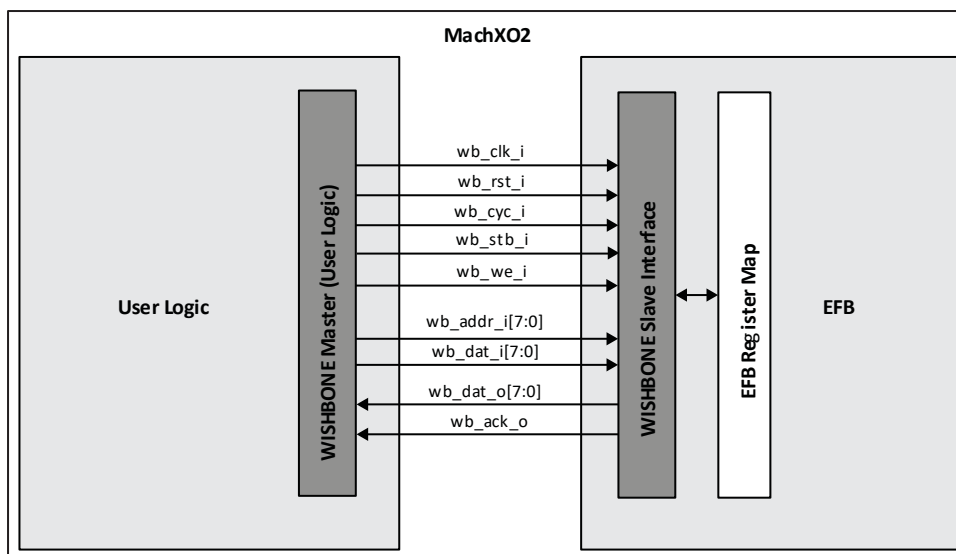
- There can be up to two PLLs in a MachXO2 device. PLL0 has an address range from 0x00 to 0x1F. PLL1 (if present) has an address range from 0x20 to 0x3F. Reference [MachXO2 sysCLOCK PLL Design and Usage Guide \(FPGA-TN-02157\)](#), for details on PLL configuration registers and recommended usage.

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

## 1.2. WISHBONE Bus Interface

The WISHBONE Bus in the MachXO2 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 Master interface to interact with the EFB WISHBONE slave 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 Slave 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 <sup>2</sup> C User Slave or Configuration Slave ports, the clock speed must be at least 7.5x the I <sup>2</sup> C bus speed (for example, >3.0 MHz when I <sup>2</sup> C rate = 400 kHz).
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 transactions.
wb_cyc_i	Input	1	Active-high signal, asserted by the WISHBONE master, indicates a valid bus cycle is present on the bus.
wb_stb_i	Input	1	Active-high strobe, input signal, indicating the WISHBONE slave is the target for the current transaction on the bus. The EFB module asserts an acknowledgment in response to the assertion of the strobe.
wb_we_i	Input	1	Level sensitive Write/Read control signal. Low indicates a Read operation, and High indicates a Write operation.
wb_adr_i	Input	8	8-bit wide address used to select a specific register from the register map of the EFB module.
wb_dat_i	Input	8	8-bit input data path used to write a byte of data to a specific register in the register map of the EFB module.
wb_dat_o	Output	8	8-bit output data path used to read a byte of data from a specific register in the register map of the EFB module.
wb_ack_o	Output	1	Active-high, transfer acknowledge signal asserted by the EFB module, indicating the requested transfer is acknowledged.

To interface to the EFB you must create a WISHBONE Master controller in the User Logic. In a multiple-Master configuration, the WISHBONE Master 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-Master bus configurations. If two Masters request the bus in the same cycle, only the outputs of the arbitration winner reach the Slave 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 Slave interface of the EFB module. Given that the EFB is a hardened block, these signals cannot be added by the user.
- The Slave 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 slave interface does not support the optional error and retry access termination signals. If the slave receives an access to an invalid address, it simply responds 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 Slave interface. During a single Write cycle, only one byte of data is written to the EFB block from the WISHBONE Master. A Write operation requires a minimum three clock cycles.

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

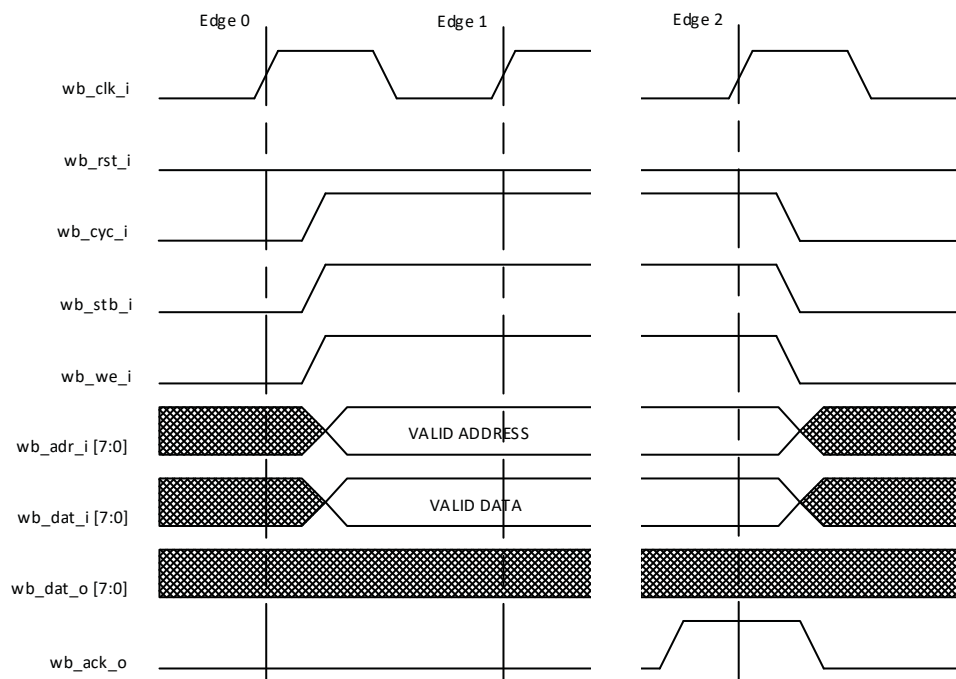
- The Master updates the address on the wb\_adr\_i[7:0] address lines.
- Updates the data written to the EFB block, wb\_dat\_i[7:0] data lines.
- Asserts the write enable wb\_we\_i signal, indicating a write cycle.
- Asserts the wb\_cyc\_i to indicate the start of the cycle.
- Asserts the wb\_stb\_i, selecting a specific slave module.

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

- The Slave decodes the address presented on the wb\_adr\_i[7:0] address lines.
- The Slave prepares to latch the data presented on the wb\_dat\_i[7:0] data lines.
- The Master 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 Slave asserts wb\_ack\_o signal.

The following occurs on clock Edge 2:

- The Slave latches the data presented on the wb\_dat\_i[7:0] data lines.
- The Master de-asserts the strobe signal, wb\_stb\_i, the cycle signal, wb\_cyc\_i, and the write enable signal, wb\_we\_i.
- The Slave de-asserts the acknowledge signal, wb\_ack\_o, in response to the Master 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 Slave interface. During a single Read cycle, only one byte of data is read from the EFB block by the WISHBONE master. A Read operation requires a minimum of three clock cycles.

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

- The Master 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 Slave module.

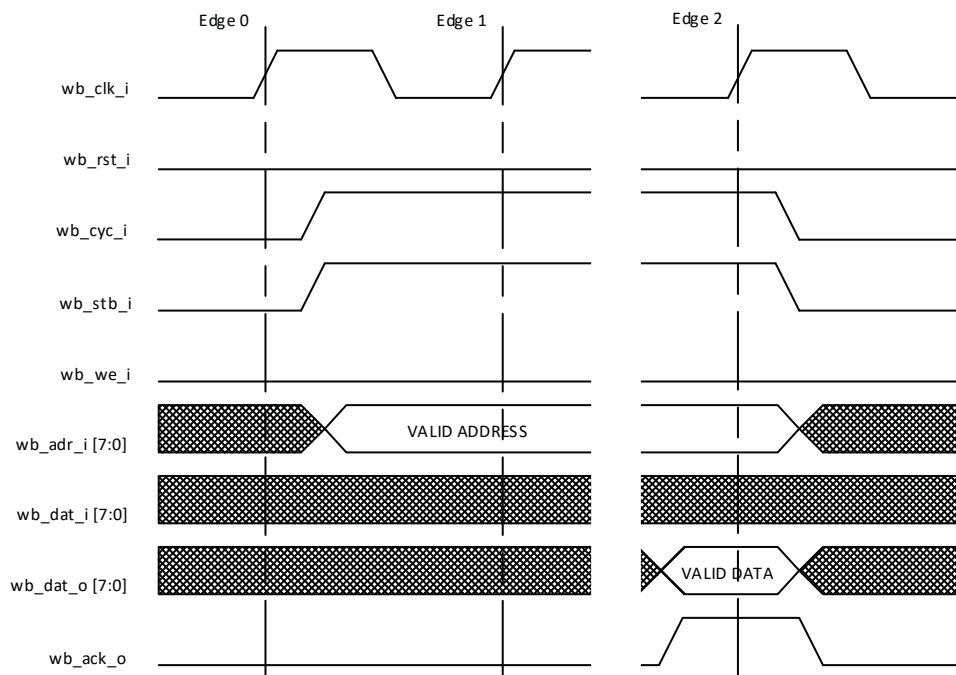
On clock Edge 1, the EFB WISHBONE slave decodes the input signals presented by the master. The following occurs during this cycle:

- The Slave decodes the address presented on the wb\_adr\_i[7:0] address lines.

- The Master prepares to latch the data presented on wb\_dat\_o[7:0] data lines from the EFB WISHBONE slave on the following clock edge.
- The Master 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 Slave presents valid data on the wb\_dat\_o[7:0] data lines.
- The Slave asserts wb\_ack\_o signal in response to the strobe, wb\_stb\_i signal.

The following occurs on clock Edge 2:

- The Master latches the data presented on the wb\_dat\_o[7:0] data lines.
- The Master de-asserts the strobe signal, wb\_stb\_i, and the cycle signal, wb\_cyc\_i.
- The Slave de-asserts the acknowledge signal, wb\_ack\_o, in response to the master de-assertion of the strobe signal.



**Figure 1.4. WISHBONE Bus Read Operation**

To avoid simulation mismatch in functional simulations, add a delay of 100 ps to wb\_cyc\_i and wb\_stb\_i assertion assignments. 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 only resets the WISHBONE interface logic. This signal does not affect the contents of any registers in the EFB register map. It only affects ongoing bus transactions.

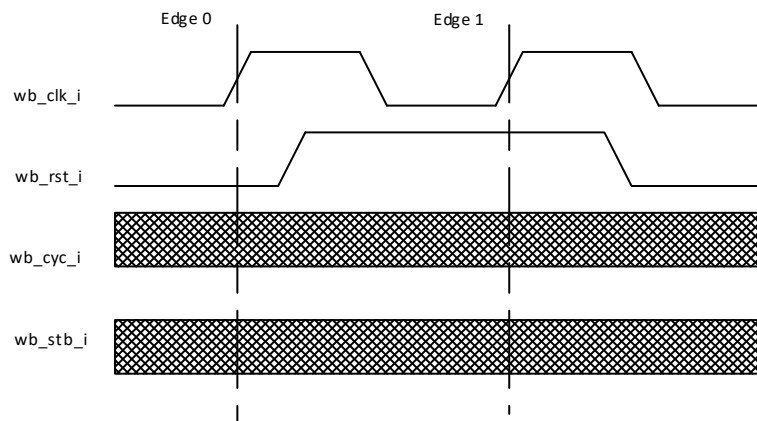


Figure 1.5. EFB WISHBONE Interface Reset

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

## 2. Hardened I<sup>2</sup>C IP Cores

I<sup>2</sup>C is a widely used two-wire serial bus for communication between devices on the same board. Every MachXO2 device contains two hardened I<sup>2</sup>C IP cores designated as the Primary and Secondary I<sup>2</sup>C IP cores. Either of the two cores can be operated as an I<sup>2</sup>C Master or as an I<sup>2</sup>C Slave. The difference between the two cores is that the Primary core has preassigned 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 I<sup>2</sup>C core can be used for accessing the User Flash Memory (UFM) and for programming the Configuration Flash. However, the Primary I<sup>2</sup>C port cannot be used for both UFM/Config access and user functions in the same design. When instantiating the Hardened I<sup>2</sup>C IP cores for Slave operations, the EFB '`wb_clk_i`' input must be connected to a valid clock source of at least 7.5x the I<sup>2</sup>C bus rate (for example, >3.0 MHz when I<sup>2</sup>C rate = 400 kHz).

## 3. I<sup>2</sup>C Registers

Both I<sup>2</sup>C cores communicate with the EFB WISHBONE interface through a set of control, command, status and data registers. Table 3.1 shows the register names and their functions. These registers are a subset of the EFB register map.

**Table 3.1. I<sup>2</sup>C Registers**

I <sup>2</sup> C Primary Register Name	I <sup>2</sup> C Secondary Register Name	Register Function	Address I <sup>2</sup> C Primary	Address I <sup>2</sup> 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 3.2. I<sup>2</sup>C 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 I<sup>2</sup>C core to reset.

I2CEN	I2C System Enable Bit – This bit enables the I2C core functions. If I2CEN is cleared, the 2C 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 slave 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 Slave 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 MachXO2 Slave Address is matched during standby/sleep mode. 0: Disable 1: Enable
SDA_DEL_SEL[1:0]	SDA Output Delay (Tdel) Selection (see <a href="#">Figure 9.1</a> ) 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 3.3. I<sup>2</sup>C 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	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	—	—

STA	Generate START (or Repeated START) condition (Master operation).
STO	Generate STOP condition (Master operation).
RD	Indicate Read from slave (Master operation).
WR	Indicate Write to slave (Master operation).
ACK	Acknowledge Option – when receiving, ACK transmission selection. 0: Send ACK 1: Send NACK
CKSDIS	Clock Stretching Disable. Clock stretching is not supported in this device. Please refer to Lattice Product Change Notice (PCN) #10A-13 for further information. This bit must be set to '1' for all writes to this register.

**Table 3.4. I<sup>2</sup>C 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 <sup>1</sup>	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:**

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

**Table 3.5. I<sup>2</sup>C Register 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[9:8]	
Default <sup>1</sup>	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	—	R/W	R/W

**Note:**

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

**I2C\_PRESCALE[9:0]** I<sup>2</sup>C Clock Prescale value. A write operation to I2CBR [9:8] causes an I2C core reset.  
The WISHBONE clock frequency is divided by (I2C\_PRESCALE\*4) to produce the Master I<sup>2</sup>C clock frequency supported by the I<sup>2</sup>C bus (50 kHz, 100 kHz, 400 kHz).

**Note:**

Different from transmitting a Master, the practical limit for Slave I2C bus speed support is (WISHBONE clock)/2048. For example, the maximum WISHBONE clock frequency to support a 50 kHz Slave I2C operation is 102 MHz.

**Note:**

The digital value is calculated by IPexpress™ when the I<sup>2</sup>C core is configured in the I<sup>2</sup>C tab of the EFB GUI. The calculation is based on the WISHBONE Clock Frequency and the I<sup>2</sup>C Frequency, both entered by the user. The digital value of the divider is programmed in the MachXO2 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 I<sup>2</sup>C bus frequencies.

**Table 3.6. I<sup>2</sup>C 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_Data[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] I<sup>2</sup>C 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 slave address, Bit 0 represents the Read/Write bit.

**Table 3.7. I<sup>2</sup>C Status (Primary/Secondary)**

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

**Note:**

- These bits exhibit 0.5 SCK period latency before valid in R1 devices. For more details on the R1 to Standard migration, refer to [Designing for Migration from MachXO2-1200-R1 to Standard \(Non-R1\) Devices \(FPGA-AN-02012\)](#).

**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 wakeup and before the first valid I<sup>2</sup>C 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** I<sup>2</sup>C Bus busy. The I<sup>2</sup>C 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 I<sup>2</sup>C SR bits be treated as valid indicators for a valid transfer.

1: I<sup>2</sup>C bus busy

0: I<sup>2</sup>C bus not busy

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

1: No acknowledge received

0: Acknowledge received

**SRW** Slave Read/Write. Indicates transmit or receive mode.

1: Master receiving/slave transmitting

0: Master transmitting/slave 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 6.1](#).

**ARBL** Arbitration Lost. The core has lost arbitration in Master 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 master/slave 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 master/slave mode and SRW status). Note: When acting as a transmitter (Master Write or Slave Read) a No Acknowledge received also 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 0: Normal
HGC	Hardware General Call Received. A hardware general call has been received in slave mode. The corresponding command byte is available in the General Call Data Register. This bit is capable of generating an interrupt. 1: General Call Received in slave mode 0: Normal

**Table 3.8. I<sup>2</sup>C 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] I<sup>2</sup>C General Call Data. This register holds the second (command) byte of the General Call transaction on the I2C bus.

**Table 3.9. I<sup>2</sup>C 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] I<sup>2</sup>C 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 3.10. I<sup>2</sup>C Interrupt Status (Primary/Secondary)**

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



IRQARBL	<p>Interrupt Status for Arbitration Lost.</p> <p>When enabled, indicates ARBL is asserted. Write a '1' to this bit to clear the interrupt.</p> <p>1: Arbitration Lost Interrupt</p> <p>0: No interrupt</p>
IRQTRRDY	<p>Interrupt Status for Transmitter or Receiver Ready.</p> <p>When enabled, indicates TRRDY is asserted. Write a '1' to this bit to clear the interrupt.</p> <p>1: Transmitter or Receiver Ready Interrupt</p> <p>0: No interrupt</p>
IRQTROE	<p>Interrupt Status for Transmitter/Receiver Overrun or NACK received.</p> <p>When enabled, indicates TROE is asserted. Write a '1' to this bit to clear the interrupt.</p> <p>1: Transmitter or Receiver Overrun or NACK received Interrupt</p> <p>0: No interrupt</p>
IRQHGC	<p>Interrupt Status for Hardware General Call Received.</p> <p>When enabled, indicates HGC is asserted. Write a '1' to this bit to clear the interrupt.</p> <p>1: General Call Received in slave mode Interrupt</p> <p>0: No interrupt</p>

**Table 3.11. I<sup>2</sup>C Interrupt Enable (Primary/Secondary)**

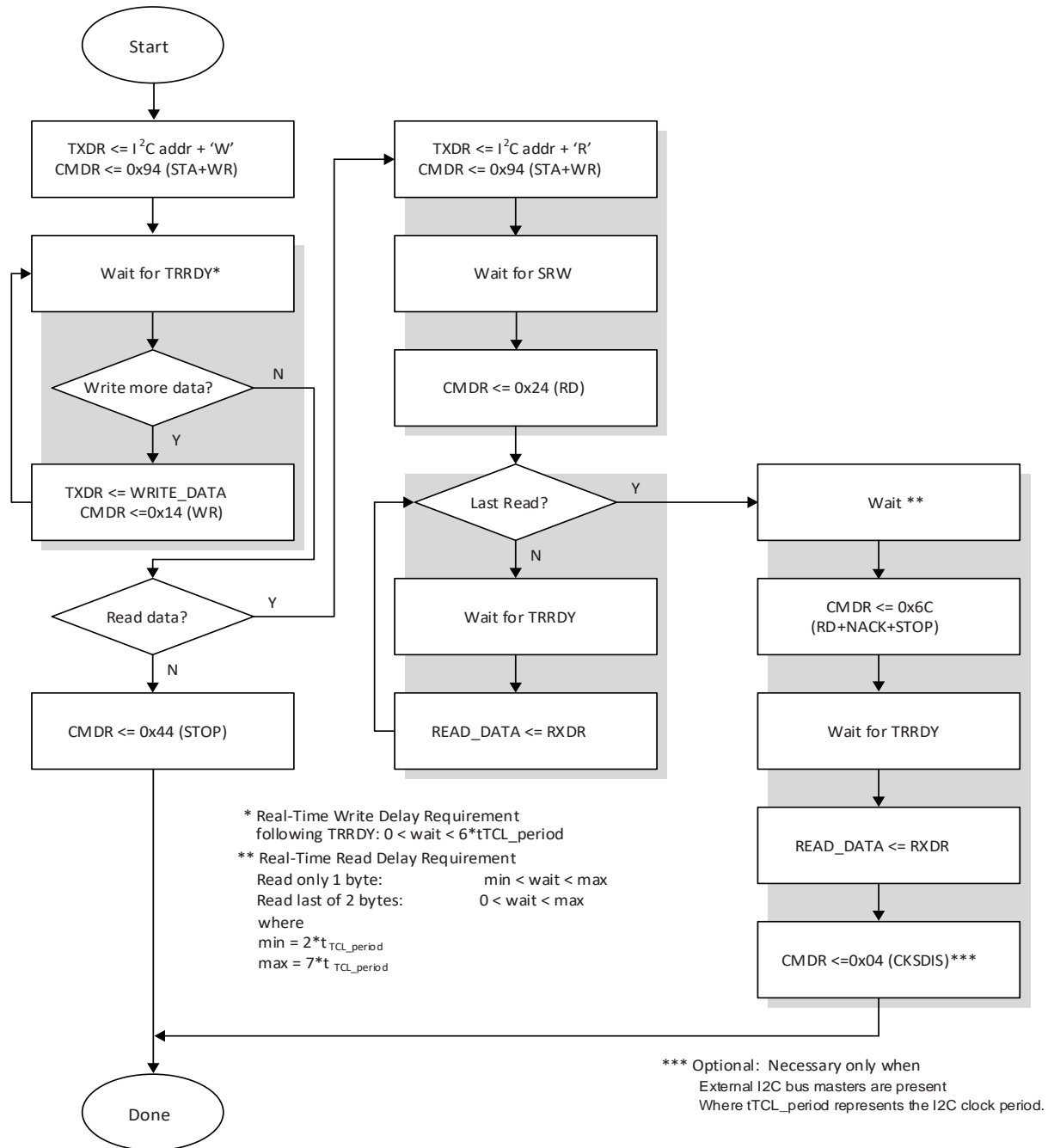
I2C_1_IRQEN / I2C_2_IRQEN						0x49/0x53		
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)				IRQARBL	IRQTRRDY	IRQTROE	IRQHGC
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	R/W	R/W	R/W	R/W

IRQARBL	<p>Interrupt Enable for Arbitration Lost</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>
IRQTRRDY	<p>Interrupt Enable for Transmitter or Receiver Ready</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>
IRQTROE	<p>Interrupt Enable for Transmitter/Receiver Overrun or NACK Received</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>
IRQHGC	<p>Interrupt Enable for Hardware General Call Received</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>

Figure 3.1 shows a flow diagram for controlling Master I<sup>2</sup>C reads and writes initiated via the WISHBONE interface.

The following sequence is for the Primary I<sup>2</sup>C but the same sequence applies to the Secondary I<sup>2</sup>C.

Note:  $t_{TCL}$  refers to the I<sup>2</sup>C clock period.



**Figure 3.1. I²C Master Read/Write Example (via WISHBONE)**

Figure 3.2 shows a flow diagram for reading and writing from an I<sup>2</sup>C Slave device via the WISHBONE interface. The following sequence is for the Primary I<sup>2</sup>C but the same sequence applies to the Secondary I<sup>2</sup>C.

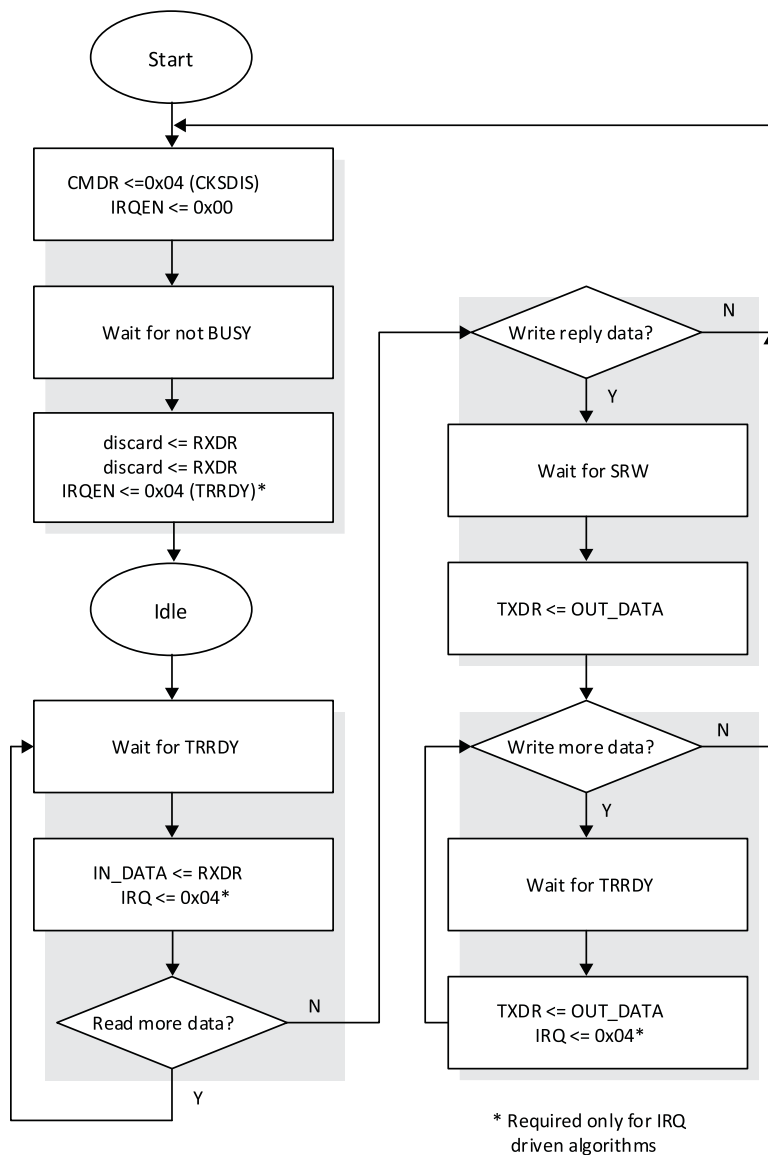


Figure 3.2. I<sup>2</sup>C Slave Read/Write Example (via WISHBONE)

## 4. Typical I<sup>2</sup>C Transactions

Figure 4.1, Figure 4.2, and Figure 4.3 illustrate typical User I<sup>2</sup>C bus protocol transactions that are supported by the Master and Slave flows shown in Figure 3.1 and Figure 3.2. Additionally, the figures below reference typical sysConfig Configuration commands structures.

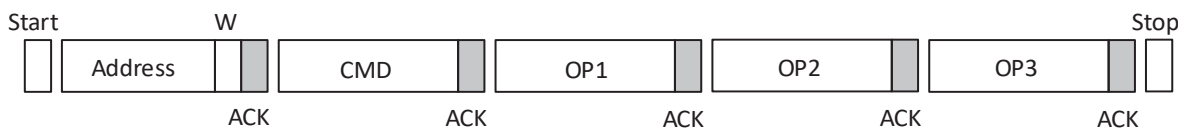


Figure 4.1. Simple I<sup>2</sup>C Command (for example, ISC\_ERASE)

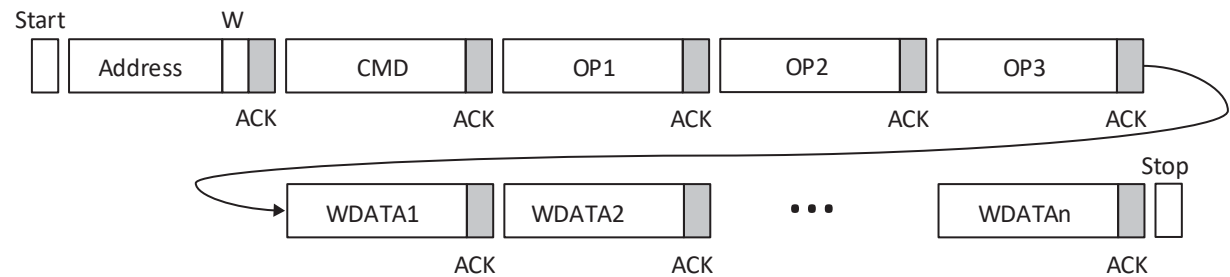


Figure 4.2. I²C Command with Write Data (for example, LSC\_PROG\_INCR\_NV)

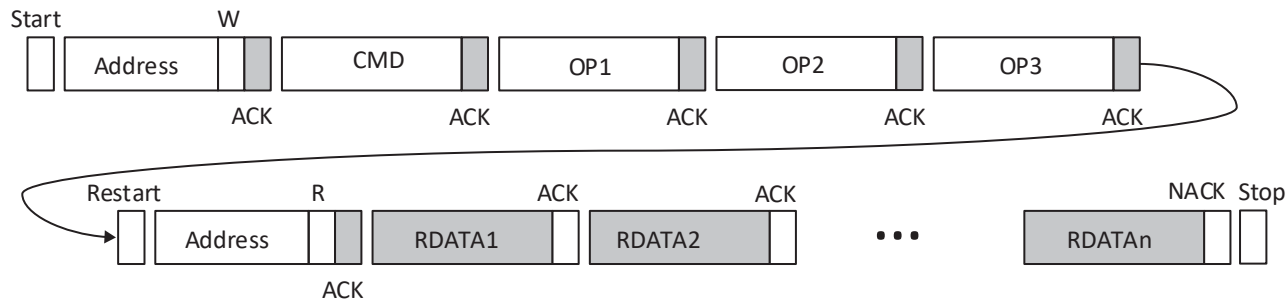
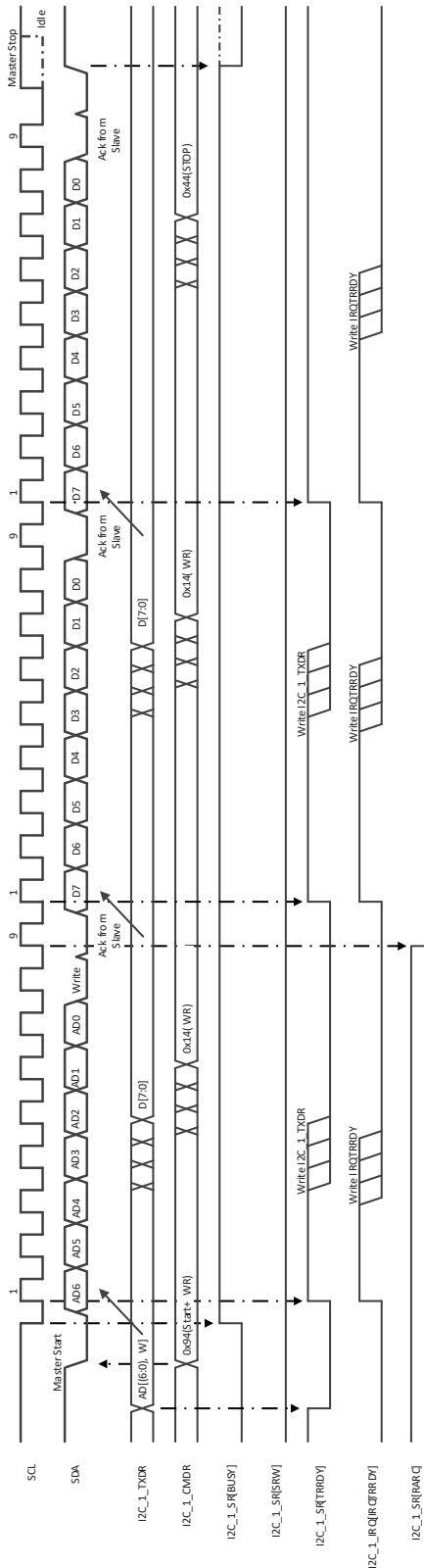


Figure 4.3. I²C Command with Read Data (for example, LSC\_READ\_STATUS)

## 5. I<sup>2</sup>C Functional Waveforms



**Figure 5.1. EFB Master – I<sup>2</sup>C Write**

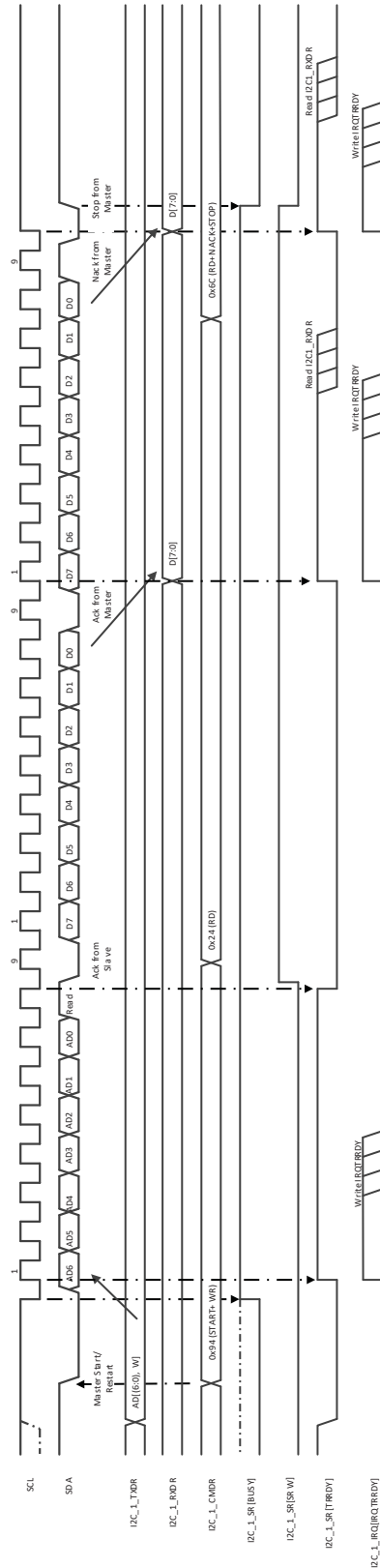
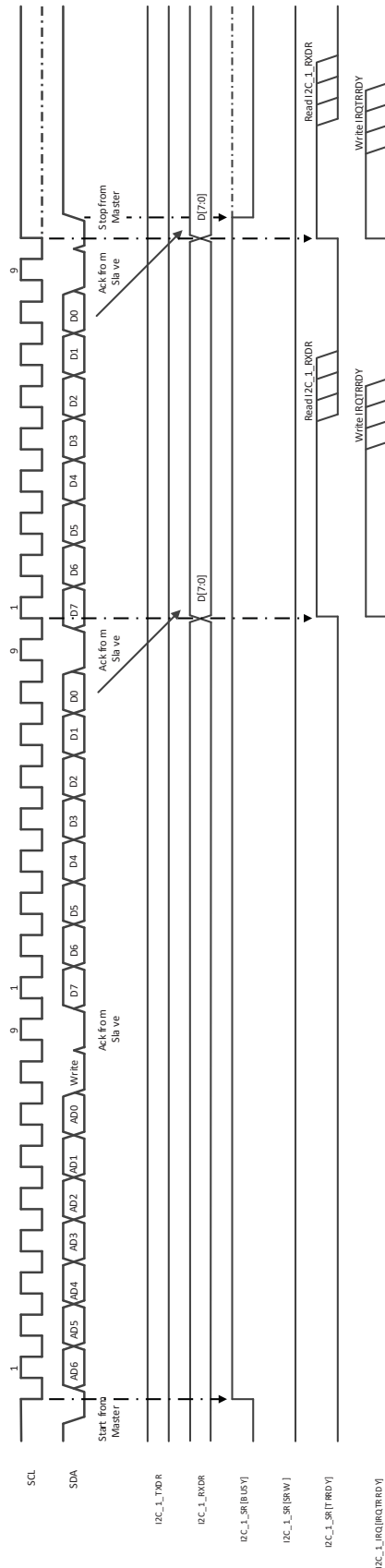
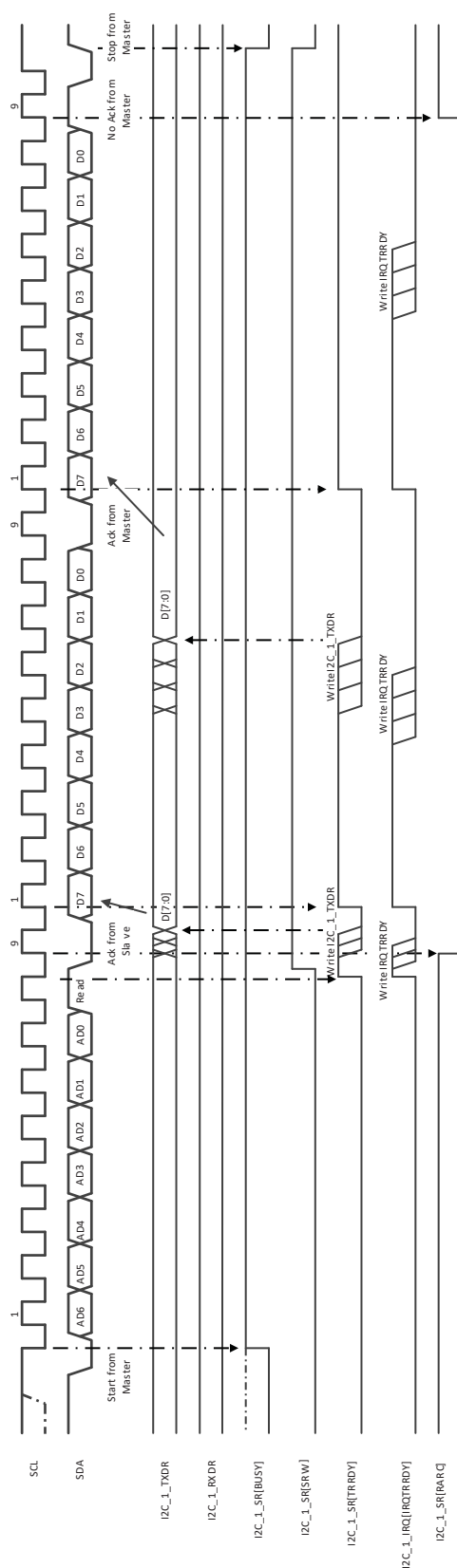


Figure 5.2. EFB Master – I<sup>2</sup>C Read



**Figure 5.3. EFB Slave – I<sup>2</sup>C Write**



**Figure 5.4. EFB Slave – I<sup>2</sup>C Read**



## 6. I<sup>2</sup>C Timing Diagram

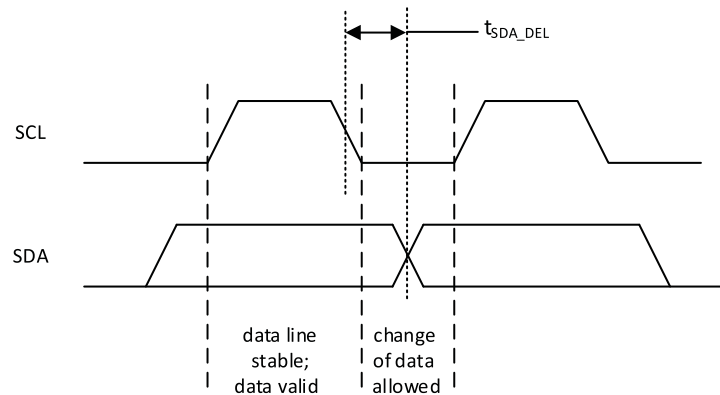


Figure 6.1. I<sup>2</sup>C Bit Transfer Timing

## 7. I<sup>2</sup>C Simulation Model

The I<sup>2</sup>C EFB Register Map translation to the MachXO2 EFB software simulation model is provided in below.

Table 7.1. I<sup>2</sup>C Primary Simulation Mode

I <sup>2</sup> C Primary Register Name	Register Size/Bit Location	Register Function	Address I <sup>2</sup> 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 <sup>2</sup> C Primary Register Name	Register Size/Bit Location	Register Function	Address I <sup>2</sup> 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	0x46	Read	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	0x47	Read	i2crxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/ njport_unit/i2c_1st/
I2C_Receive_Data[7:0]	[7:0]	—	—	—	i2crxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/ njport_unit/i2c_1st/
I2C_1_IRQ	[7:0]	IRQ	0x48	Read/Write	{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/

I <sup>2</sup> C Primary Register Name	Register Size/Bit Location	Register Function	Address I <sup>2</sup> C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_1_IRQEN	[7:0]	IRQ Enable	0x49	Read/Write	{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/
IRQTRDYEN	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 7.2. I<sup>2</sup>C Secondary Simulation Model**

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

I <sup>2</sup> C Secondary Register Name	Register Size/Bit Location	Register Function	Address I <sup>2</sup> C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_2_BR1	[7:0]	Clock Pre-scale	0x4D	Read/Write	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/ njport_unit/i2c_2nd/
I2C_PRESCALER[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/
I2C_2_IRQ	[7:0]	IRQ	0x52	Read/Write	{1'b0, 1'b0, 1'b0, 1'b0, i2csr_2nd_irqs_ts_3, i2csr_2nd_irqs_ts_2, i2csr_2nd_irqs_ts_1, i2csr_2nd_irqs_ts_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQARBL	3	—	—	—	i2csr_2nd_irqs_ts_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDY	2	—	—	—	i2csr_2nd_irqs_ts_2	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTROE	1	—	—	—	i2csr_2nd_irqs_ts_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQHGC	0	—	—	—	i2csr_2nd_irqs_ts_0	../efb_top/efb_pll_sci_inst/u_efb_sci/

I <sup>2</sup> C Secondary Register Name	Register Size/Bit Location	Register Function	Address I <sup>2</sup> C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
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/

## 8. Hardened SPI IP Core

The MachXO2 EFB contains a hard SPI IP core that can be configured as a SPI Master or Slave. When the SPI core is configured as a Master it is able to control other devices with Slave SPI interfaces that are connected to the SPI bus. When the SPI core is configured as a Slave, it is able to interface to an external SPI Master device.

## 9. SPI Registers

The SPI core communicates with the WISHBONE interface through a set of control, command, status and data registers. Table 9.1 shows the register names and their functions. These registers are a subset of the EFB register map.

**Table 9.1. 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	Master 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 9.2. 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 Master Chip Select low assertion (Master 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 Master Chip Select (Master 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 Master Chip Select low assertion and the first edge of SCK (Master Mode only), in SCK periods.

000: ½

001: 1

010: 1.5

...

111: 4

**Table 9.3. SPI Control 1**

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

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

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

0: SPI disabled.

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

**WKUPEN\_USER** Wake-up Enable via User. Enables the SPI core to send a wake-up signal to the onchip Power Controller to wake the part from Standby mode when the User slave 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 slave 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 <a href="#">Figure 12.1</a> to <a href="#">Figure 12.4</a> . 0: Transmit data on the MCLK/CCLK edge defined by SPICR2[CPOL] and SPICR2[CPHA]. 1: Transmit data ½ MCLK/CCLK edge earlier than defined by SPICR2[CPOL] and SPICR2[CPHA].

**Table 9.4. 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	—	—	R/W	R/W	R/W

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

MSTR SPI	Master/Slave Mode. Selects the Master/Slave operation mode of the SPI core. Changing this bit forces the SPI system into idle state. 0: SPI is in Slave mode 1: SPI is in Master mode
MCSH	SPI Master CSSPIN Hold. Holds the Master chip select active when the host is busy, to halt the data transmission without de-asserting chip select. <b>Note:</b> This mode must be used only when the WISHBONE clock has been divided by a value greater than four (4) (greater than six (6) for R1 devices). For more details on the R1 to Standard migration refer to <a href="#">Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices (FPGA-AN-02012)</a> . 0: Master running as normal 1: Master holds chip select low even if there is no data to be transmitted
SDBRE	Slave 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 slave read out data predictably available at high SPI clock rates. When enabled, dummy 0xFF bytes are transmitted in response to a SPI slave 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 master should scan for the initial 0x00 byte when reading the SPI slave to indicate the beginning of actual data. Refer to <a href="#">Figure 11.2</a> . 0: Normal Slave SPI operation 1: Lattice proprietary Slave Dummy Byte Response Enabled <b>Note:</b> This mechanism only applies for 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	<p>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 master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state. Refer to <a href="#">Figure 12.1</a> to <a href="#">Figure 12.4</a>.</p> <p>0: Active-high clocks selected.</p> <p>1: Active-low clocks selected.</p>
CPHA	<p>SPI Clock Phase. Selects the SPI clock format. In master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state. Refer to <a href="#">Figure 12.1</a> to <a href="#">Figure 12.4</a>.</p> <p>0: Data is captured on a leading (first) clock edge, and propagated on the opposite clock edge.</p> <p>1: Data is captured on a trailing (second) clock edge, and propagated on the opposite clock edge*.</p> <p>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. Slave SPI Configuration mode supports default setting only for CPOL, CPHA.</p>
LSBF	<p>LSB-First. LSB appears first on the SPI interface. In master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state. Refer to <a href="#">Figure 12.1</a> to <a href="#">Figure 12.4</a>.</p> <p>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.</p> <p>0: Data is transferred, most significant bit (MSB) first.</p> <p>1: Data is transferred, least significant bit (LSB) first.</p>

**Table 9.5. SPI Clock Prescale**

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

**Note:**

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

**DIVIDER[5:0]** SPI Clock Prescale 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. DIVIDER must be  $\geq 1$ .

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 MachXO2 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 9.6. SPI Master 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 Master Chip Selects. Used in master mode for asserting a specific Master Chip Select (MCSN) line. The register has eight bits, enabling the SPI core to control up to eight external SPI slave devices. Each bit represents one master 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 9.7. 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 it is transmitted last when SPICR2[LSBF]=0 or first when SPICR2[LSBF]=1.  
Note: When operating as a Slave, 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 16.1](#). This timing requires at least one protocol dummy byte be included for all slave SPI read operations.

**Table 9.8. 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 complete

1: SPI Transmitting in progress\*

Note: This bit is non-functional in R1 devices. For more details on the R1 to Standard migration refer to [Designing for Migration from MachXO2-1200-R1 to Standard \(Non-R1\) Devices \(FPGA-AN-02012\)](#).

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 received 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 Slave 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 9.9. 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 9.10. 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

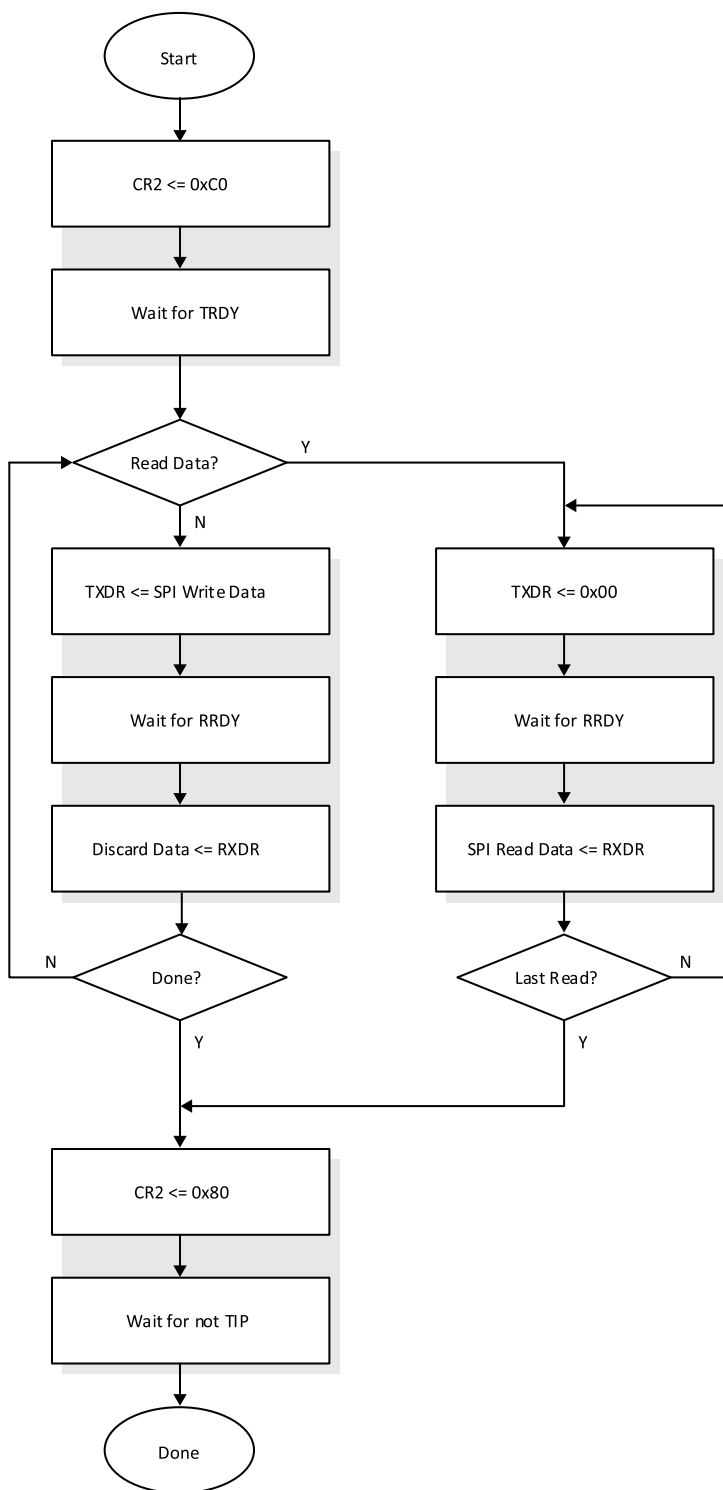
IRQROE	<p>Interrupt Status for Receive Overrun Error.</p> <p>When enabled, indicates ROE is asserted. Write a '1' to this bit to clear the interrupt.</p> <p>1: Receive Overrun Error Interrupt</p> <p>0: No interrupt</p>
IRQMDF	<p>Interrupt Status for Mode Fault.</p> <p>When enabled, indicates MDF is asserted. Write a '1' to this bit to clear the interrupt.</p> <p>1: Mode Fault Interrupt</p> <p>0: No interrupt</p>

**Table 9.11. 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	0	0	0	0
Access	—	—	—	R/W	R/W	—	R/W	R/W

IRQTRDYEN	<p>Interrupt Enable for SPI Transmit Ready.</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>
IRQRRDYEN	<p>Interrupt Enable for SPI Receive Ready</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>
IRQROEEN	<p>Interrupt Enable for Receive Overrun Error</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>
IRQMDFEN	<p>Interrupt Enable for Mode Fault</p> <p>1: Interrupt generation enabled</p> <p>0: Interrupt generation disabled</p>

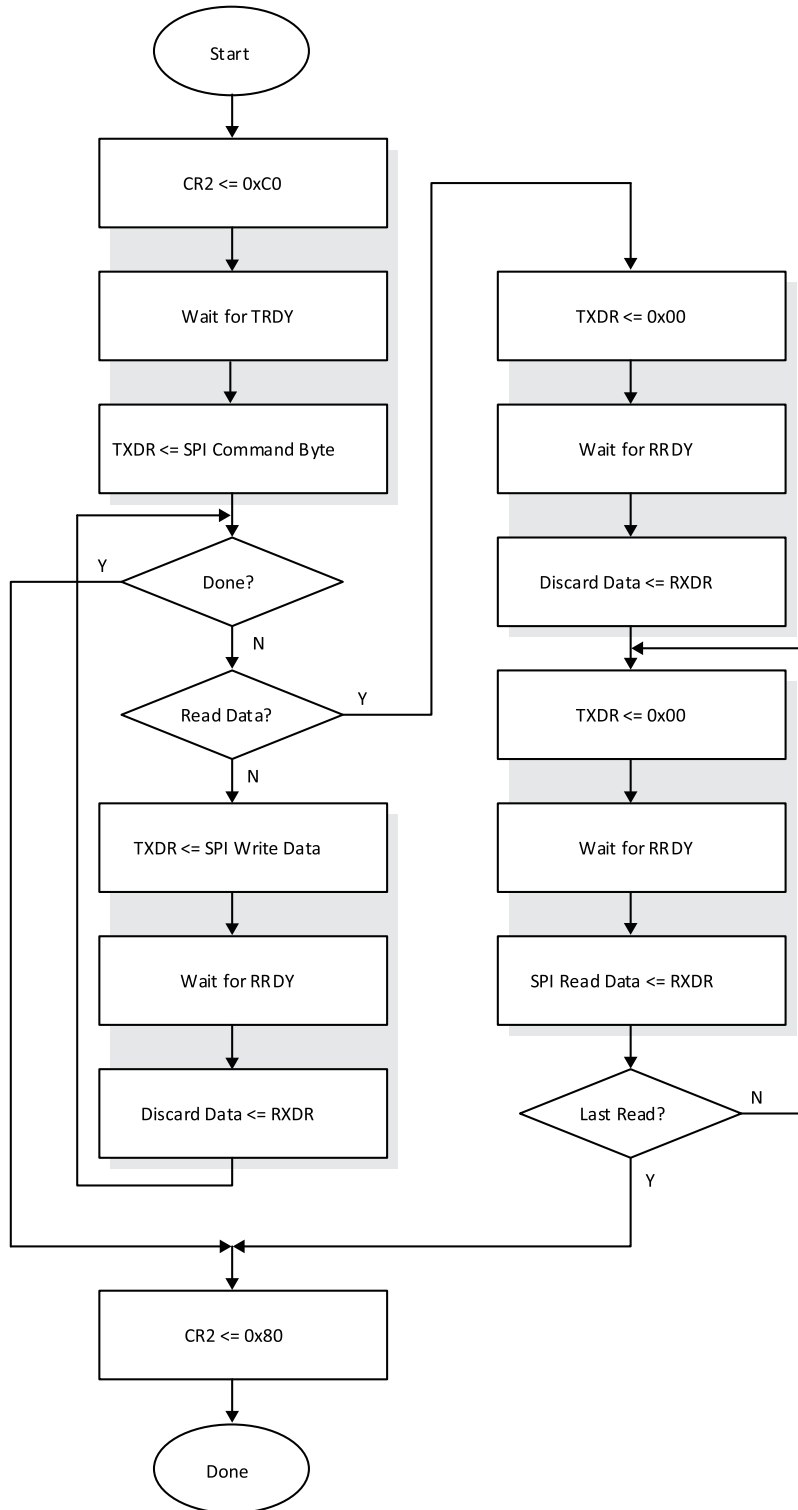
Figure 9.1. shows a flow diagram for controlling Master SPI reads and writes initiated via the WISHBONE interface.



**Figure 9.1. SPI Master 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 11.1](#) for guidance.



**Figure 9.2. SPI Master Read/Write Example (via WISHBONE) – R1 Silicon**

**Note:**

For more details on the R1 to Standard migration refer to [Designing for Migration from MachXO2- 1200-R1 to Standard \(Non-R1\) Devices \(FPGA-AN-02012\)](#).

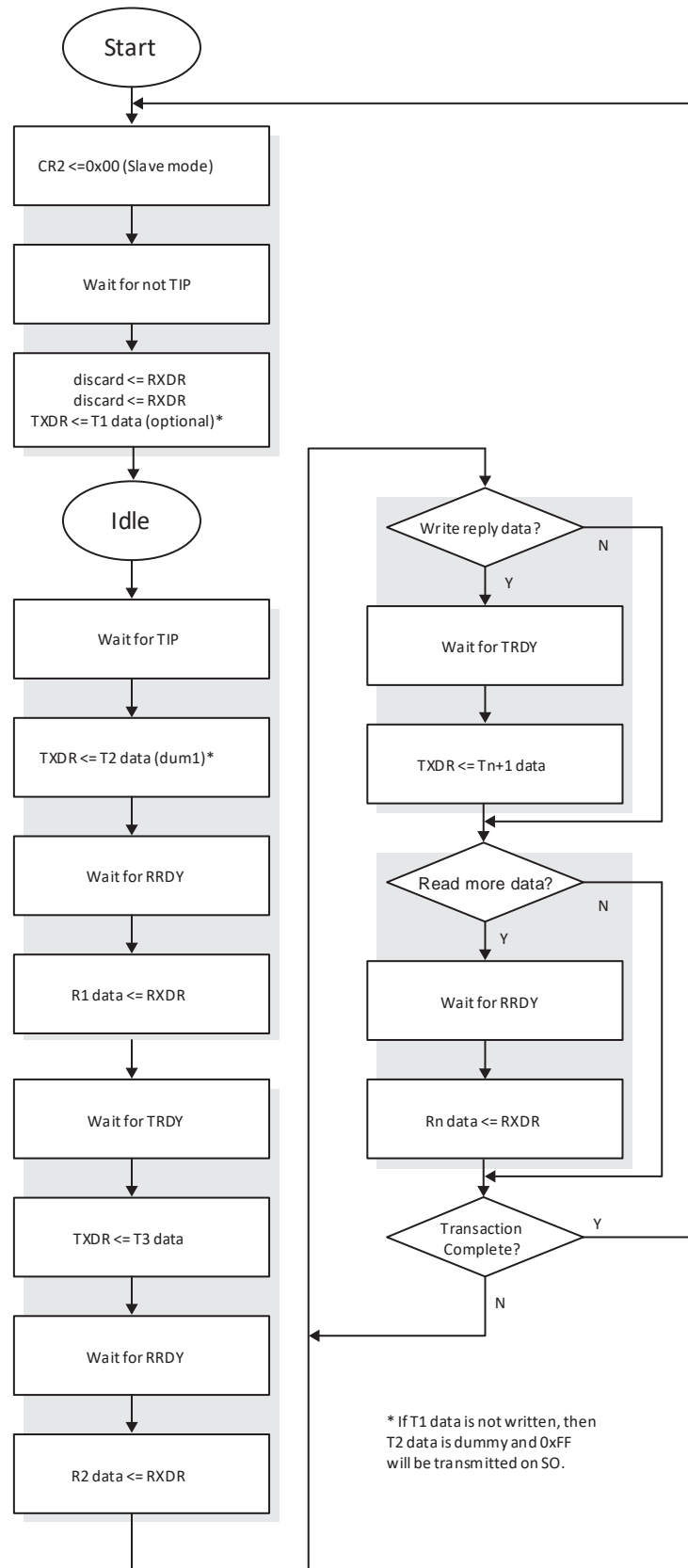
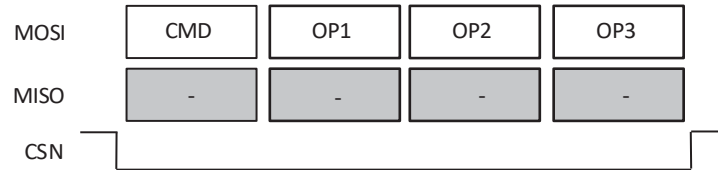


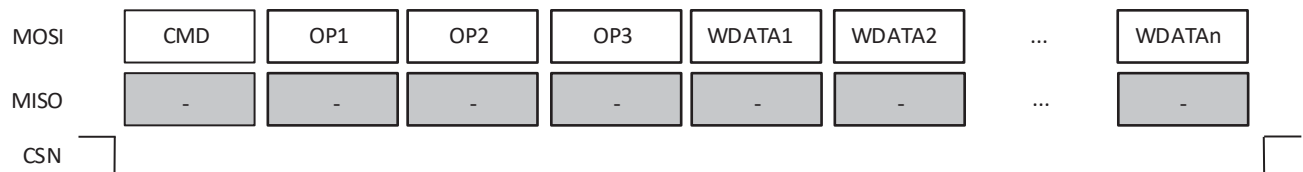
Figure 9.3. SPI Slave Read/Write Example (via WISHBONE) – Production Silicon

## 10. Typical SPI Transactions

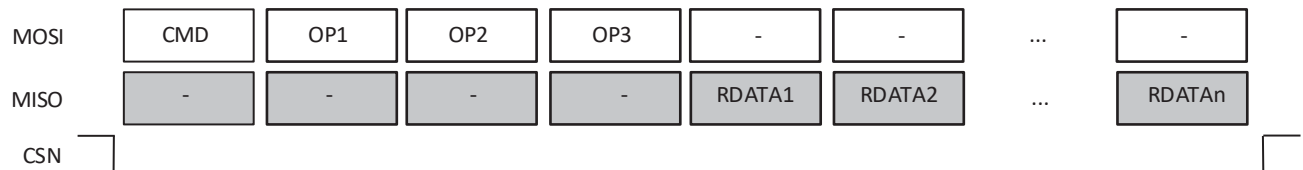
Figure 10.1, Figure 10.2, and Figure 10.3 illustrate typical User SPI bus protocol transactions that are supported by the Master and Slave flows shown in Figure 9.1, Figure 9.2, and Figure 9.3. Additionally, the figures below reference typical sysConfig Configuration commands structures.



**Figure 10.1. Simple SPI Command (for example, ISC\_ERASE)**

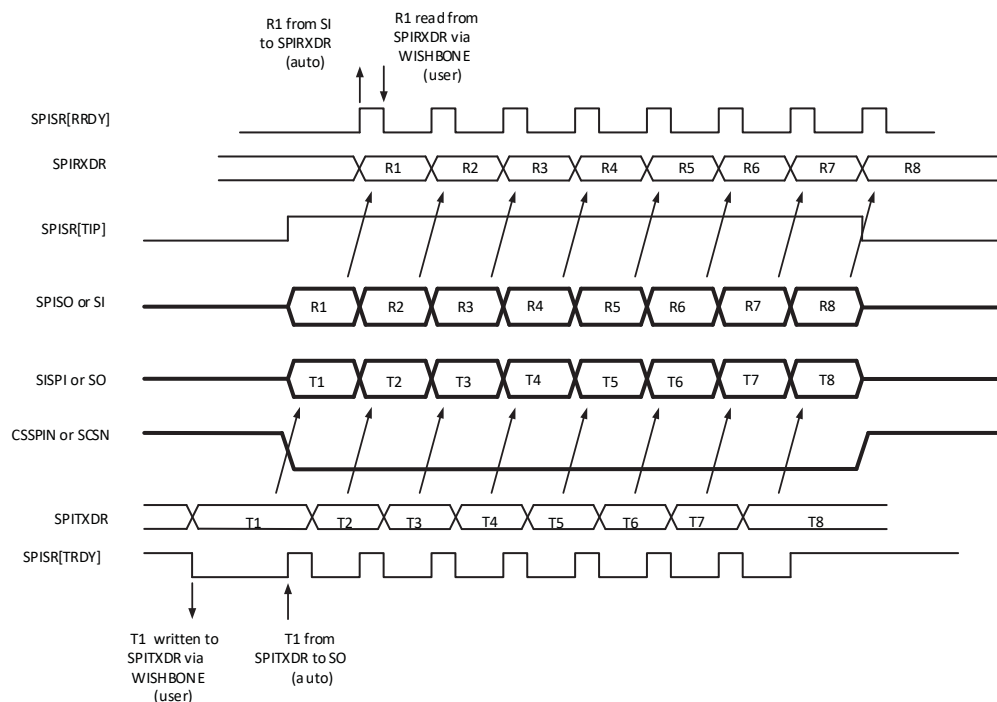


**Figure 10.2. SPI Command w/ Write Data (for example, LSC\_PROG\_INCR\_NV)**

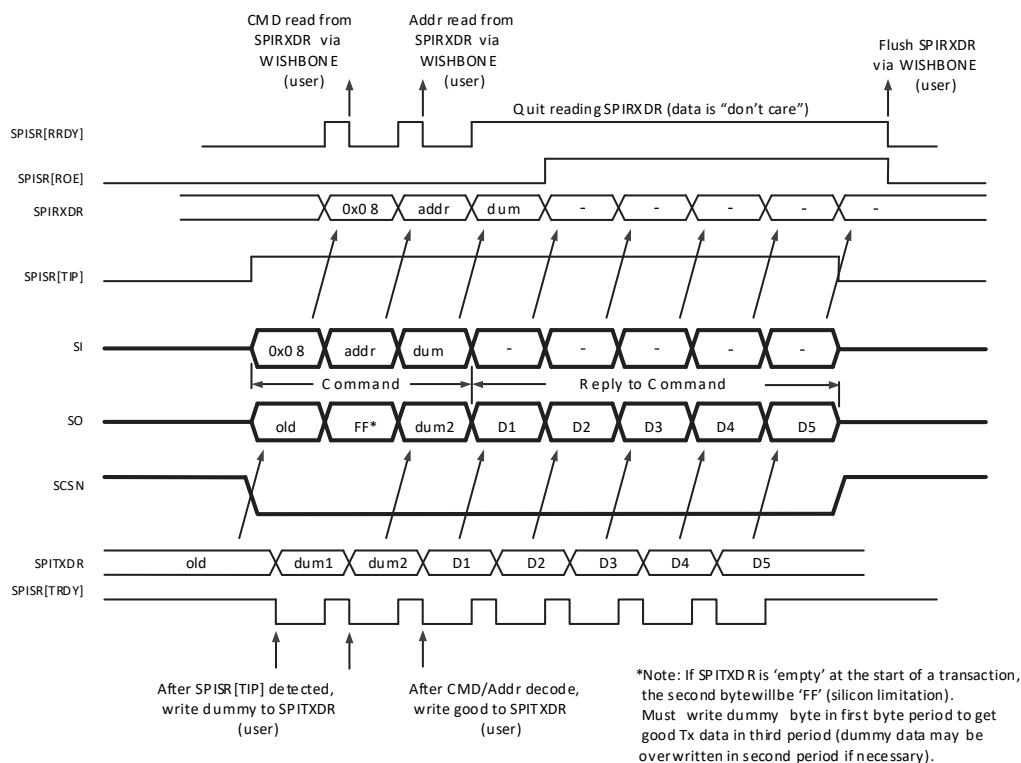


**Figure 10.3. SPI Command w/ Read Data (for example, LSC\_READ\_STATUS)**

## 11. SPI Functional Waveforms



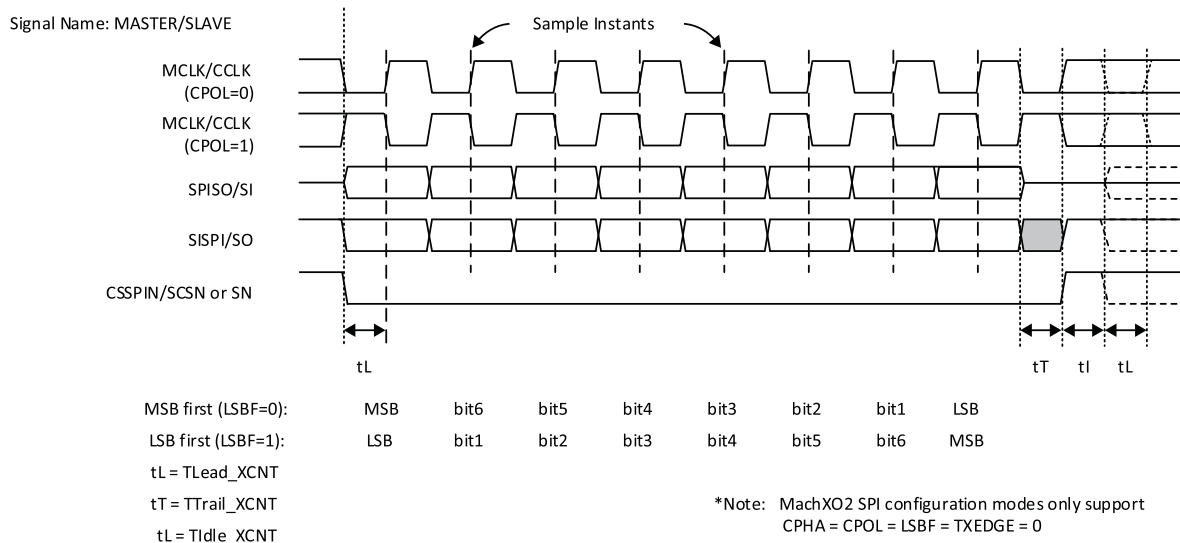
**Figure 11.1. Fully Specified SPI Transaction (MachXO2 as SPI Master or Slave)**



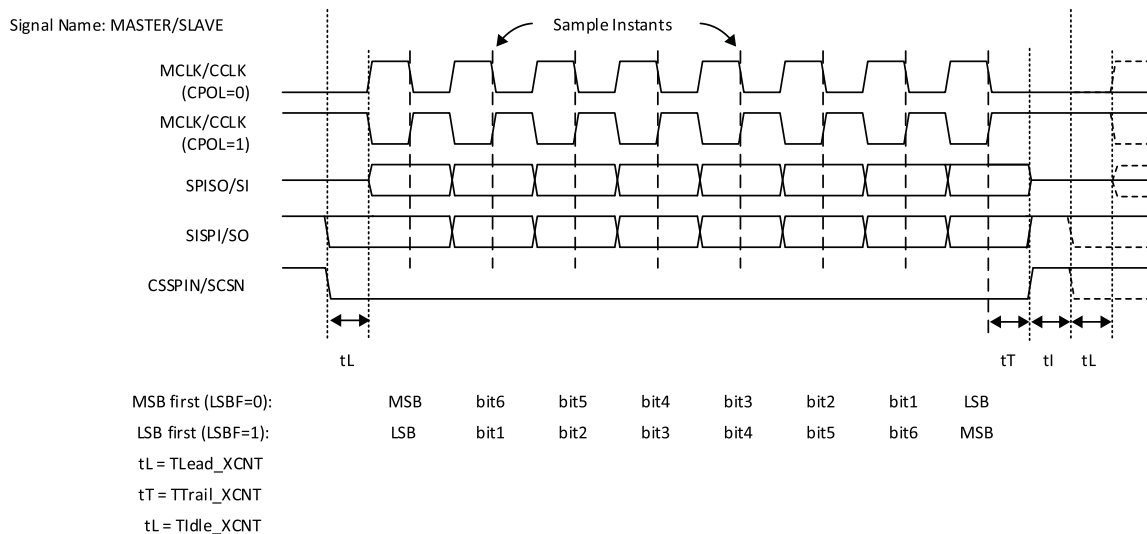
**Figure 11.2. Minimally Specified SPI Transaction Example (MachXO2 as SPI Slave)**



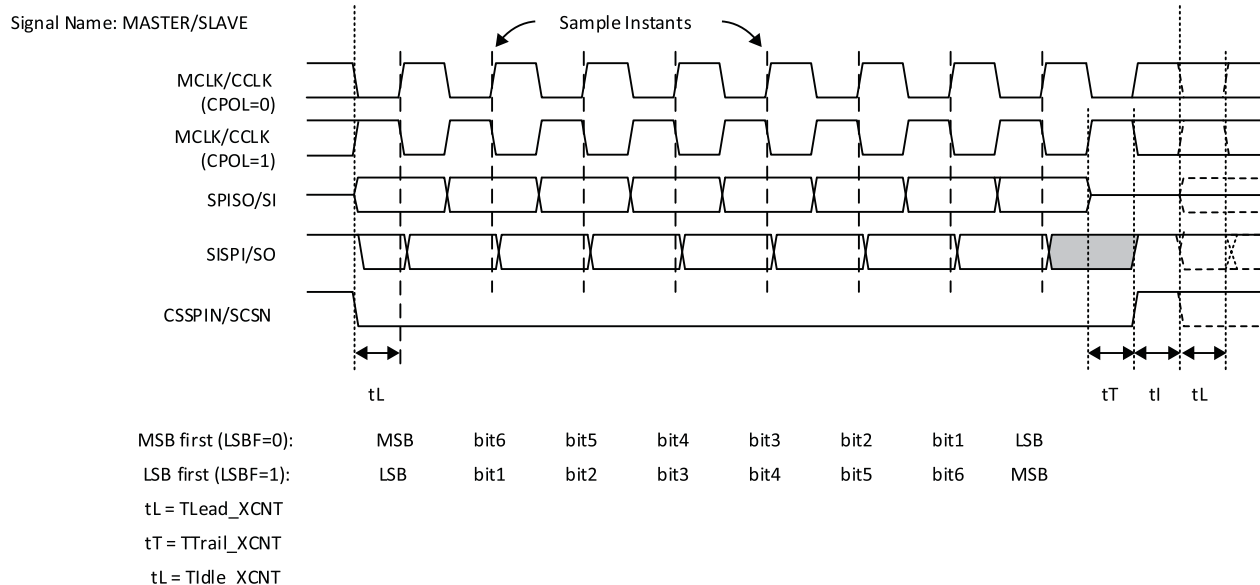
## 12. SPI Timing Diagrams



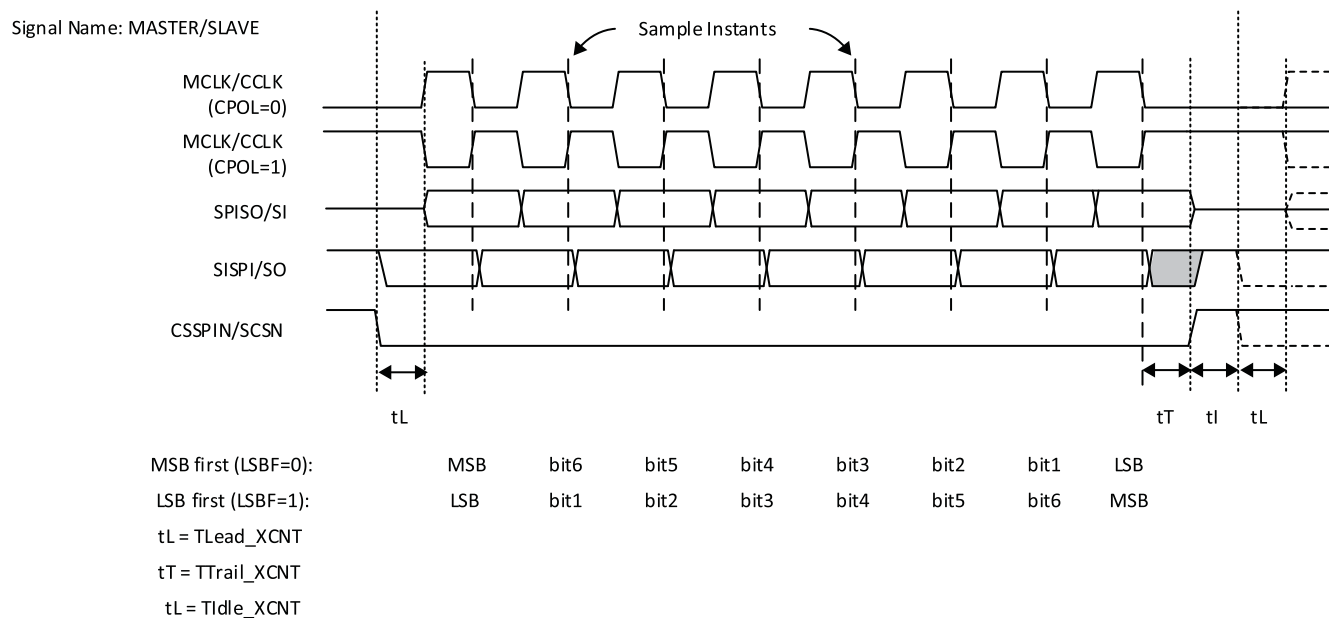
**Figure 12.1. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)**



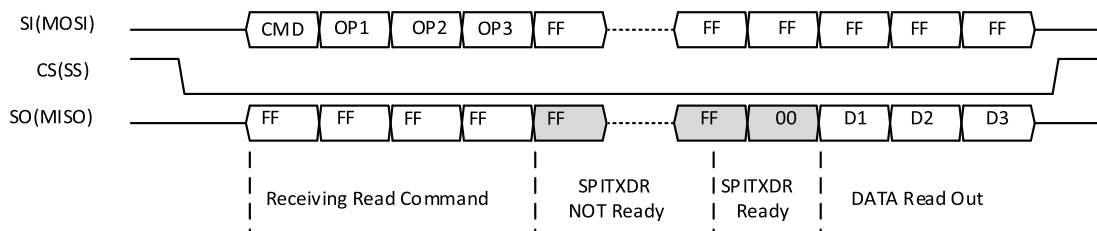
**Figure 12.2. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)**



**Figure 12.3. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)**



**Figure 12.4. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)**



**Figure 12.5. Slave SPI Dummy Byte Response (SPICR2[SDBRE]) Timing**

## 13. SPI Simulation Model

The SPI EFB Register Map translation to the MachXO2 EFB software simulation model is provided below.

**Table 13.1. 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_mcs	../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]	Master Chip Select	0x58	Read/Write	spicsr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
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/

SPI Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
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/
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/

## 14. Hardened Timer/Counter PWM

The MachXO2 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.

### 14.1. 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 14.1](#) 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 14.1. 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 14.2. Timer/Counter Control 0**

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 14.3. Timer/Counter Control 1**

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

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

0: Disabled

1: Enabled

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

**ICEN** Enables the ability to perform a capture operation of the counter value. 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 14.4. Timer/Counter Set Top Counter Value 0**

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

**Note:**

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

**Table 14.5. Timer/Counter Set Top Counter Value 1**

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

**Note:**

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

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

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

**Table 14.6. Timer/Counter Set Compare Counter Value 0**

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

**Note:**

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

**Table 14.7. Timer/Counter Set Compare Counter Value 1**

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

**Note:**

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

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

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

**Table 14.8. Timer/Counter Control 2**

TCCR2								0x64
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)					WBFORCE	WBRESET	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 matched the compare (TCOCR) value or it matched 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 14.9. 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 14.10. 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 14.11. 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 14.12. 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 TCTOPSET0/ 1. The data stored in these registers represents the top value of the counter. The registers update once the counter has completed the current counting cycle. The WISHBONE host has read-only access to these registers. Refer to the Timer/Counter Modes of Operation section for usage details.

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

**Table 14.13. 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 14.14. Timer/Counter Current Compare Counter Value 1**

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

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

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

**Table 14.15. 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 14.16. 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, which combined, 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 time-stamp 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 14.17. 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 14.18. 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 14.19. 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

## 15. Timer Counter Simulation Model

The Timer Counter EFB Register Map translation to the MachXO2 EFB software simulation model is provided below.

**Table 15.1. 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_ovf_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], 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/
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], 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/
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/

Timer/Counter Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
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 [7:0]	0x69	Read	tc_ocr_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
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 [7:0]	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 [15:8]	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/
IRQOVFEN	0	—	—	—	tc_ovf_irqena	../efb_top/efb_pll_sci_inst/u_efb_sci/

## 16. Flash Memory (UFM/Configuration) Access

Designers can access the Flash Memory Configuration Logic interface using the JTAG, SPI, I2C, or WISHBONE interfaces. The MachXO2 Flash Memory consists of two sectors:

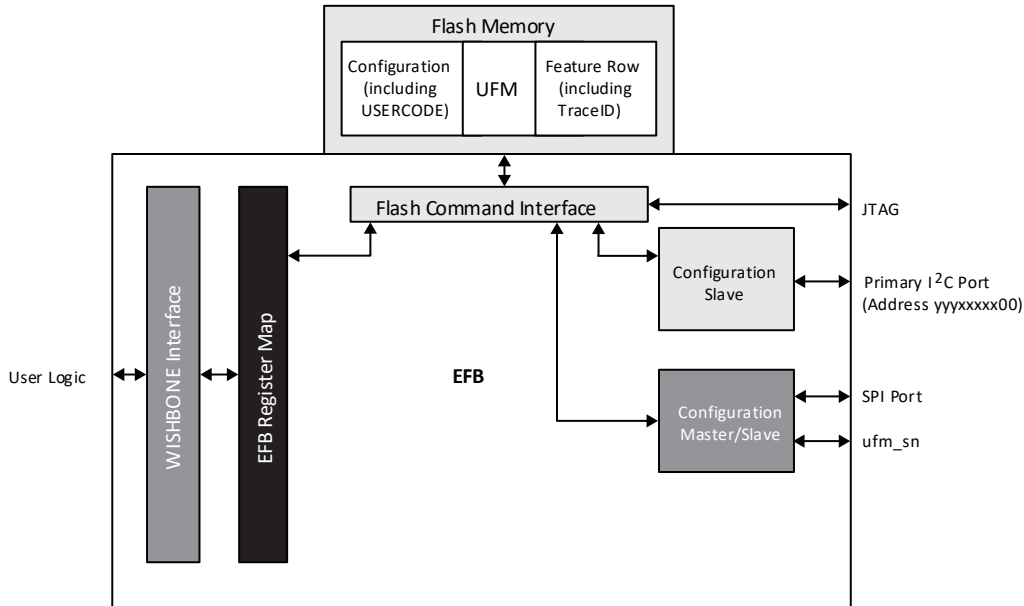
- User Flash Memory (UFM)
  - MachXO2-640 and higher density devices provide one sector of User Flash Memory (UFM).
- Configuration
  - Configuration consists of two sectors Configuration Flash and the Feature Row.

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

### 16.1. Flash Memory (UFM/Configuration) Access Ports

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

Figure 16.1 illustrates the interfaces to the UFM and Configuration Memory sectors.



**Figure 16.1. Interfaces to the UFM/Configuration Sectors**

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

1. JTAG Port
2. Slave SPI Port
3. I2C Primary Port
4. WISHBONE Slave Interface

**Note:**

Enabling Flash Memory (UFM/Configuration) Interface using Enable Configuration Interface command 0x74 Transparent Mode temporarily disables certain features of the device including:

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

Functionality is restored after the Flash Memory (UFM/Configuration) Interface is disabled using Disable Configuration Interface command 0x26 followed by Bypass command 0xFF.

## 16.2. Flash Memory (UFM/Configuration) Access through WISHBONE Slave Interface

The WISHBONE Slave interface of the EFB module enables designers to access the Flash Memory (UFM/Configuration) 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. In addition to the WISHBONE bus signals, an interrupt request output signal is brought to the FPGA fabric. The IP signal is “wbc\_ufm\_irq”, and it functions as an interrupt request to the internal WISHBONE host, based on the data Read/Write FIFO status or arbitration error.

**Note:**

To access the Flash Memory (UFM/Configuration) via WISHBONE in R1 devices, the hard SPI port or the primary I2C port must be enabled. For more details, refer to [Designing for Migration from MachXO2-1200-R1 to Standard \(Non-R1\) Devices \(FPGA-AN-02012\)](#).

The WISHBONE Interface communicates to the Configuration Logic through a set of data, control and status registers. [Table 16.1](#) 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 16.1. WISHBONE to Flash Memory (CFG) 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 Enable	0x75	Read/Write

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

**Table 16.2. Flash Memory (UFM/Configuration) Control**

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

WBCE

WISHBONE Connection Enable. Enables the WISHBONE to establish the read/write connection to the Flash Memory (UFM/Configuration) logic. This bit must be set prior to executing any command through the WISHBONE port. Likewise, this bit must be cleared to terminate the command. Refer to the [Command and Data Transfers to Flash Memory \(UFM/Configuration\) 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 16.3. Flash Memory (UFM/Configuration) 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 that is written to the Flash Memory (UFM/Configuration) logic. Bit 0 is LSB.

**Table 16.4. Flash Memory (UFM/Configuration) Status**

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

CFGSR								0x72
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	Slave SPI Active. Indicates the Slave SPI port has started actively communicating with the Configuration Logic while WBCE is enabled. This port has priority over the I2C and WISHBONE ports and preempts any existing, and prohibit any new, lower priority transaction. This bit is capable of generating an interrupt. 1: Slave SPI port active 0: Slave SPI port not active
I2CACT	I <sup>2</sup> C 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 and preempts 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 16.5. Flash Memory (UFM/Configuration) Receive Data**

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



CFG\_Receive\_Data[7:0] CFG Receive Data. This register holds the byte read from the Flash Memory (UFM/Configuration) logic. Bit 0 in this register is LSB.

**Table 16.6. Flash Memory (UFM/Configuration) 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 Slave SPI Active.  
When enabled, indicates SSPIACT is asserted. Write a '1' to this bit to clear the interrupt.  
1: Slave 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 16.7. Flash Memory (UFM/Configuration) 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 Slave 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 16.8. 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 16.9. EFB Interrupt Source**

EFBIRQ								0x77
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)			UFMCFG_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

UFMCFG_INT	Flash Memory (UFM/Configuration) Interrupt Source. Indicates EFB interrupt source is from the UFM/Configuration 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 SPIIRQ 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

### 16.3. Command and Data Transfers to Flash Memory (UFM/Configuration) Space

The command and data transfers to the Flash Memory (UFM/Configuration) are identical for all the access ports, regardless of their different physical interfaces. The Flash Memory (UFM/Configuration) 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 Flash memory is composed of two sectors, Configuration Memory (sector 0) and UFM (sector 1). The Erase operations are sector based.

### 16.4. Command Summary by Application

**Table 16.10. UFM (Sector 1) Commands**

Command Name	Command MSB LSB	SVF Command Name	Description
Read Status Register	0x3C	LSC_READ_STATUS	Read the 4-byte Configuration Status Register.
Check Busy Flag	0xF0	LSC_CHECK_BUSY	Read the Configuration Busy Flag status.
Bypass	0xFF	ISC_NOOP	Null operation.
Enable Configuration Interface (Transparent Mode)	0x74	ISC_ENABLE_X	Enable Transparent UFM access – All user I/Os (except the hardened user SPI and primary user I <sup>2</sup> 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 UFM access – All user I/Os (except persisted sysCONFIG ports) are tri-stated. User logic ceases to function, UFM remains accessible, and the device enters ‘Offline’ access mode. (The subsequent commands in this table require the interface to be enabled.)
Disable Configuration Interface	0x26	ISC_DISABLE	Disable the configuration (UFM) access.
Set Address	0xB4	LSC_WRITE_ADDRESS	Set the UFM sector 14-bit Address Register.
Reset UFM Address	0x47	LSC_INIT_ADDR_UFM	Reset the address to point to Sector 1, Page 0 of the UFM.
Read UFM	0xCA	LSC_READ_TAG	Read the UFM data. Operand specifies number of pages to read. Address Register is post-incremented.
Erase UFM	0xCB	LSC_ERASE_TAG	Erase the UFM sector only.
Program UFM Page	0xC9	LSC_PROG_TAG	Write one page of data to the UFM. Address Register is post-incremented.

**Table 16.11. Configuration Flash (Sector 0) Commands**

Command Name	Command MSB LSB	SVF Command Name	Description
Read Device ID	0xE0	IDCODE_PUB	Read the 4-byte Device ID (0x01 2b 20 43).
Read USERCODE	0xC0	USERCODE	Read 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 <sup>1</sup>	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 Flash access – All user I/Os (except the hardened user SPI and primary user I <sup>2</sup> 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 Flash access – All user I/Os (except persisted sysCONFIG ports) are tri-stated. User logic ceases to function, UFM remains accessible, and the device enters ‘Offline’ access mode. (The subsequent commands in this table require the interface to be enabled.)
Disable Configuration Interface	0x26	ISC_DISABLE	Exit access mode.
Set Configuration Flash Address	0xB4	LSC_WRITE_ADDRESS	Set the Configuration Flash 14-bit Page Address.
Verify Device ID	0xE2	VERIFY_ID	Verify device ID with 32-bit input, set Fail flag if mismatched.
Reset Configuration Flash Address	0x46	LSC_INIT_ADDRESS	Reset the address to point to Sector 0, Page 0 of the Configuration Flash.
Read Flash	0x73	LSC_READ_INCR_NV	Read the Flash data. Operand specifies number of pages to read. Address Register is post-incremented.
Erase	0x0E	ISC_ERASE	Erase the Config Flash, FEATURE Row, FEABITS, Done bit, Security bits and USER-CODE.
Program Page	0x70	LSC_PROG_INCR_NV	Write 1 page of data to the Flash Memory (Configuration/UFM). Address Register is post-incremented.
Program DONE	0x5E	ISC_PROGRAM_DONE	Program the Done bit.
Program SECURITY	0xCE	ISC_PROGRAM_SECURITY	Program the Security bit (Secures CFG Flash sector).
Program SECURITY PLUS	0xCF	ISC_PROGRAM_SECPLUS	Program the Security Plus bit (Secures CFG and UFM Sectors). Note: SECURITY and SECURITY PLUS commands are mutually exclusive.
Program USERCODE	0xC2	ISC_PROGRAM_USERCODE	Program 32-bit USERCODE.
Read Feature Row	0xE7	LSC_READ_FEATURE	Read Feature Row.
Program Feature Row	0xE4	LSC_PROG_FEATURE	Program Feature Row.
Read FEABITS	0xFB	LSC_READ_FEABITS	Read FEA bits.

Command Name	Command MSB LSB	SVF Command Name	Description
Program FEABits	0xF8	LSC_PROG_FEABITS	Program the FEA bits.

**Note:** The Refresh commands are not supported by the software simulation model.

**Table 16.12. 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 Slave SPI or I<sup>2</sup>C interface shift the most significant bit (MSB) first into the MachXO2 device. This is required only when communicating with the configuration logic inside the MachXO2 device.

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

## 16.5. Command Descriptions by Command Code

**Table 16.13. Erase Flash (0x0E)**

UFM	CFG	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 sector (Config Flash, DONE, security bits, USERCODE)

0: No action

1: Erase

f: Erase Feature sector (Slave 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 18.3](#)) after issuing this command for erasure to complete before issuing a subsequent command other than Read Status or Check Busy.

Erased condition for Flash bits = 0

Examples: 0x0E 04 00 00

Erase CFG sector

0x0E 08 00 00

Erase UFM sector

0x0E 0C 00 00

Erase UFM and CFG sectors

**Table 16.14. Read TraceID Code (0x19)**

UFM	CFG	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 16.15. Disable Configuration Interface (0x26)**

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

Example: 0x26 00 00

Disable Flash Memory (UFM/configuration) interface for change access.

Notes: Must have only two operands.

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

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

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

**Table 16.16. Read Status Register (0x3C)**

UFM	CFG	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 Flash or SRAM Done Flag

When C = 0 SRAM Done bit has been programmed

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

When C=1 Flash Done bit has been programmed

- D = 1 Programmed
- D = 0 Not Programmed

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

B bit 12: Busy Flag (1 = busy)

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

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

EEE bits[25:23]: Configuration Check Status

000: No Error

001: ID ERR

010: CMD ERR

011: CRC ERR

100: Preamble ERR

101: Abort ERR  
110: Overflow ERR  
111: SDM EOF  
(all other bits reserved)

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

Note: Wait at least 1 us 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 16.17. Reset CFG Address (0x46)**

UFM	CFG	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 16.18. Reset UFM Address (0x47)**

UFM	CFG	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 UFM Sector 1, page 0.

**Table 16.19. Program DONE (0x5E)**

UFM	CFG	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 16.20. Program Configuration Flash (0x70)**

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

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

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

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

Use 0x0E to erase CFG 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.

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

```
***Examples:
0x73 10 00 01
0 bytes dummy followed by one page of CFG data (16 bytes total).
0x73 10 00 04
Read 1 page dummy followed by three pages of CFG data (four pages total).
```

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

\*\*\*Examples:

0x73 00 00 01  
0 bytes dummy followed by one page CFG data (16 bytes total).  
0x73 00 00 04  
Read 2 pages dummy, followed by three sets [1 page CFG data, followed by four bytes dummy] (five pages and 12 dummy bytes total).

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

*Operand:	0001 0000 00pp pppp pppp pppp (binary), or
	0000 0000 00pp pppp pppp pppp (binary)
pp..pp:	num_pages            Number of CFG pages to read when num_pages = 1
	Number of CFG pages to read +1 when $1 < \text{num\_pages} \leq 12$



Set to 0x3FFF when num\_pages &gt; 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 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 bytes dummy followed by one page CFG data (16 bytes total)  
0x73 10 00 04  
Read 1 page dummy followed by three pages of CFG data (four pages total)  
0x73 00 00 04  
Read 2 pages dummy, followed by three sets [1 page CFG 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 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 16.24. Enable Configuration Interface (Transparent) (0x74)**

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

**Notes:** The I<sup>2</sup>C interface uses only two operands and all other interfaces use three operands. This command is required to enable modification of the UFM, configuration Flash, or nonvolatile registers (NVR). Terminate this command with command 0x26 followed by command 0xFF.

Exercising this command temporarily disables certain features of the device, notably GSR, user SPI port, primary user I<sup>2</sup>C 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 Flash pumps to fully charge.

**Example:** 0x74 08 00 00  
Enable UFM/configuration interface for change access through a non-I<sup>2</sup>C interface.

**Table 16.25. Refresh (0x79)**

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
—	—	—	—	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 deassertion or I<sup>2</sup>C stop), further bus accesses are prohibited for the duration of tREFRESH. Violating this requirement causes the Refresh process to abort and leave the MachXO2 device in an unprogrammed state.

Occasionally, following a device REFRESH or PROGRAMN pin toggle, the secondary I<sup>2</sup>C 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 I<sup>2</sup>C 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 hardblock signals a STOP on the bus then returns to the idle state. Failure to manually return the Secondary I<sup>2</sup>C port to the idle state may result in an I<sup>2</sup>C bus lock-up condition. Normal I<sup>2</sup>C activity can be commenced without additional delay.

**Table 16.26. STANDBY (0x7D)**

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
—	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 wakeup from standby mode

Notes: Must have only two operands.

The MachXO2 Power Controller needs to be included in the design.

Additionally the following can be used to trigger the Power Controller to wakeup from 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 Slave Configuration (ufm\_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 Estimation and Management for MachXO2 Devices \(FPGA-TN-02161\)](#).

**Table 16.27. Set Address (0xB4)**

UFM	CFG	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: Configuration

1: UFM

aa..aa:address14-bit page address

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

Set Address register to UFM sector, page 10 decimal

**Table 16.28. Read USERCODE (0xC0)**

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

Example: 0xC0 00 00 00  
EN Required = Y Read 4-byte USERCODE from CFG sector  
EN Required = N Read 4-byte USERCODE from SRAM

**Table 16.29. Program USERCODE (0xC2)**

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Hex)
—	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 16.30. Enable Configuration Interface (Offline) (0xC6)**

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

Operand: 08 00 00 – Enable Flash Normal mode. Normal edit mode for Offline configuration.  
Used for all offline commands described in this document, including Erase SRAM.  
00 00 00 – Enable SRAM mode. Optional edit mode. Supports Erase SRAM command only.  
Example: 0xC6 08 00 00 Enable Flash Memory (UFM/configuration) interface for offline change access.  
Notes: Use this command to enable offline modification of the UFM, Configuration 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 tri-states all user I/Os (except persisted sysCONFIG ports). User logic ceases to function. UFM remains accessible.  
Poll the BUSY bit (or wait 5 us) after issuing this command for the Flash pumps to fully charge.

**Table 16.31. Program UFM (0xC9)**

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

Example: 0xC9 00 00 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  
Write one page of data, pointed to by Address Register  
Notes: 16 data bytes must be written following the command and operand bytes to ensure proper data alignment. The Address Register is auto-incremented following the page write.  
Use 0x0E or 0xCB to erase UFM sector prior to executing this command.  
Poll the BUSY bit (or wait 200 us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

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

### Table 16.33. Read UFM (0xCA) (I<sup>2</sup>C/SPI)

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

### Table 16.34. Read UFM (0xCA) (WISHBONE)

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

© 2016-2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at [www.latticesemi.com/legal](http://www.latticesemi.com/legal). All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

- \*\*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 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 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 pages dummy followed by three sets [one page 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 16.35. Erase UFM (0xCB)**

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

- Notes:** Erased condition for UFM bits = '0'  
Poll the BUSY bit (or wait, see [Table 18.3](#).) after issuing this command for erasure to complete before issuing a subsequent command other than Read Status or Check Busy.
- Example:** 0xCB 00 00 00  
Erase UFM sector

**Table 16.36. Program SECURITY (0xCE)**

UFM	CFG	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 16.37. Program SECURITY PLUS (0xCF)**

UFM	CFG	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 16.38. Read Device ID Code (0xE0)**

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Hex)
—	x	—	N	E0	00 00 00	R	4B	See Table 16.35.

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

**Table 16.39. Device ID Table**

Device Name	HE/ZE Devices	HC Devices
MachXO2-256	0x01 2B 00 43	0x01 2B 80 43
MachXO2-640	0x01 2B 10 43	0x01 2B 90 43
MachXO2-1200/MachXO2-640U	0x01 2B 20 43	0x01 2B A0 43
MachXO2-2000/MachXO2-1200U	0x01 2B 30 43	0x01 2B B0 43
MachXO2-4000/MachXO2-2000U	0x01 2B 40 43	0x01 2B C0 43
MachXO2-7000	0x01 2B 50 43	0x01 2B D0 43

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

**Table 16.40. Verify Device ID Code (0xE2)**

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Hex)
—	x	—	Y	E2	00 00 00	W	4B	See Table 16.39.

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 16.41. Program Feature Row (0xE4)**

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

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

uu: 8 bits for the user programmable TraceID

cc cc cc cc: 32 bits of Custom ID code

Note: It is not recommended to reprogram the Feature Row in system as it should be programmed ideally once during manufacturing.

Example: 0xE4 00 00 00 00 00 01 00 00 12 34  
Program Feature Row with User I<sup>2</sup>C address set to 1, default user TraceID string, Custom ID code of 12 34

**Table 16.42. Read Feature Row (0xE7)**

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Hex)
—	—	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 Slave 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 16.43. Check Busy Flag (0xF0)**

UFM	CFG	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 16.44. Program FEABITs (0xF8)**

UFM	CFG	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 00 00

Data Format: bb: Boot Sequence

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

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

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

s: Slave 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

Note: It is not recommended to reprogram the FEABITs in system as it should be programmed ideally once during manufacturing.

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

**Table 16.45. Read FEABITs (0xFB)**

UFM	CFG	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 00 00

Data Format:

- bb: Boot Sequence
  - If b=00 (Default) and m=0 then Single Boot from Configuration Flash
  - If b=00 and m=1 then Dual Boot from Configuration Flash then External if there is a failure
  - If b=01 and m=1 then Single Boot from External Flash
- m: Master SPI Port Persistence  
0=Disabled (Default), 1=Enabled
- i: I2C Port Persistence  
0=Enabled (Default), 1=Disabled
- s: Slave 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

**Table 16.46. Bypass (Null Operation) (0xFF)**

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

Note: Operands are optional

Example: 0xFF FF FF FF Bypass

## 17. Interface to Configuration Flash

The WISHBONE interface of the EFB module allows a WISHBONE host to access the configuration resources of the MachXO2 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 Configuration 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 Configuration Flash is loaded into the Configuration SRAM and the device continues operation with a new configuration.

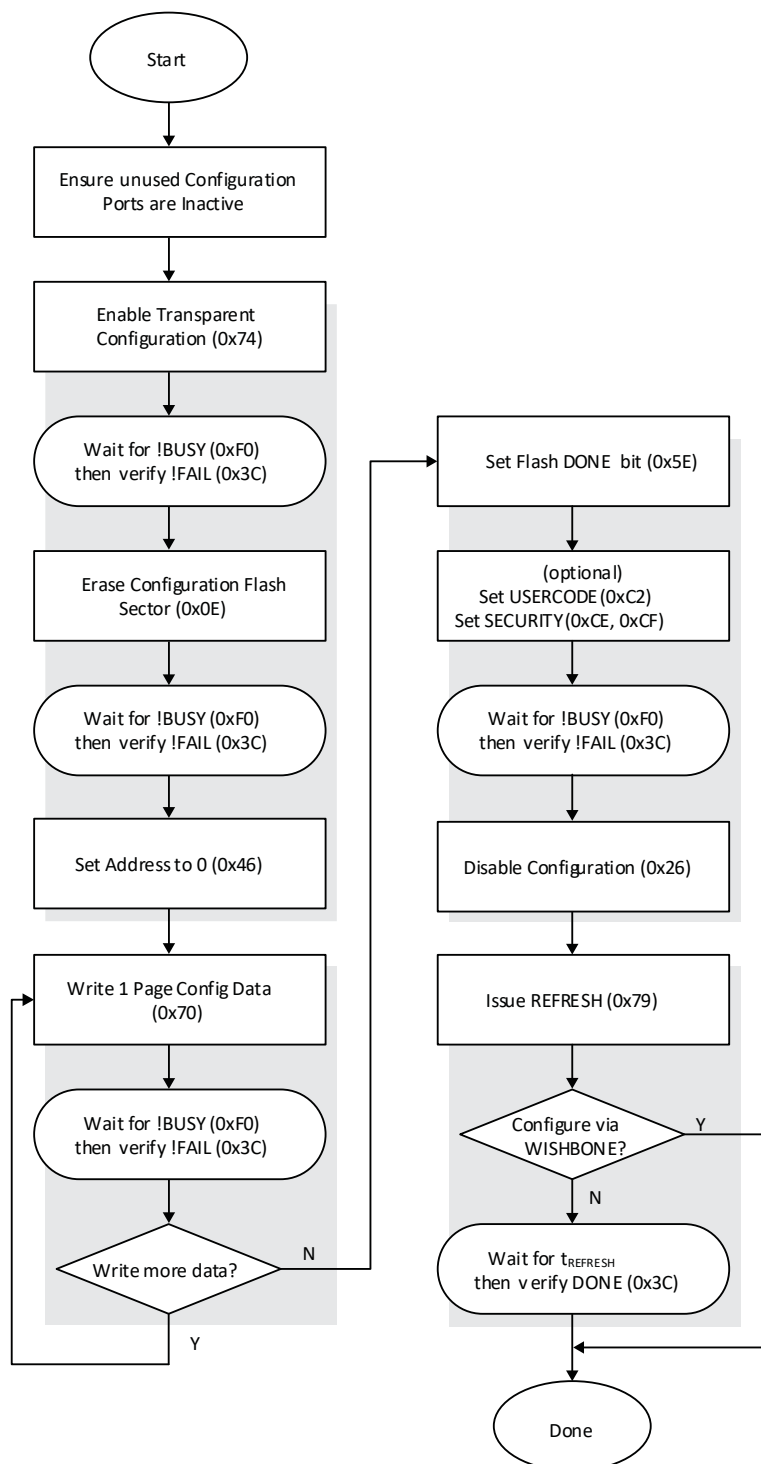
The data transfer and execution of operations is the same as the one documented in the UFM section of this document. This is due to the fact that the UFM is also a Flash Memory resource and the communication between the



WISHBONE host and the configuration logic is performed through the same command, status and data registers. Please see Table 16.1 to Table 18.3 for information on these registers.

Figure 17.1 shows a basic flow diagram for implementing a Configuration Flash Update initiated via any of the sys-CONFIG ports (I2C, SPI, or WISHBONE).

For detailed information on MachXO2 programming and configuration, see [MachXO2 Programming and Configuration Usage Guide \(FPGA-TN-02155\)](#).



**Figure 17.1. Basic Configuration Flash Update Example**

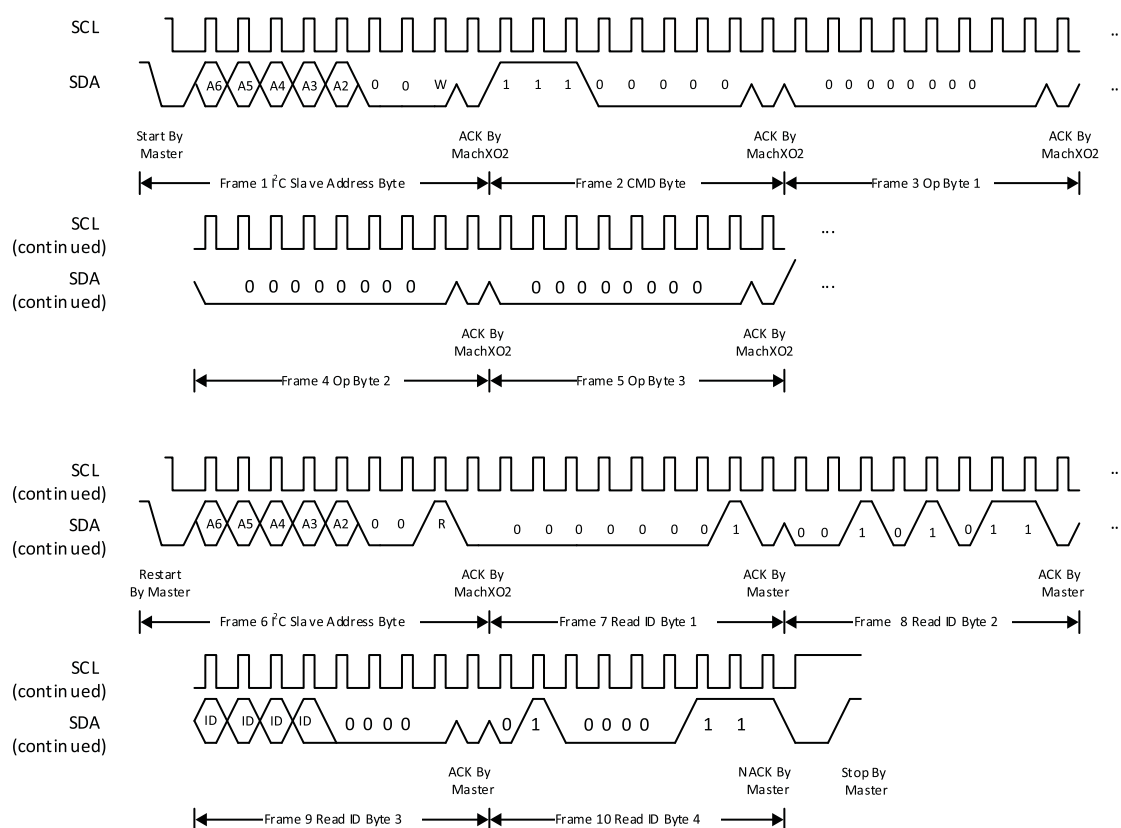
## 18. Command Framing

### 18.1. I<sup>2</sup>C Framing

Each command string sent to the I<sup>2</sup>C EFB port must be correctly framed using the protocol defined for each interface. In the case of I<sup>2</sup>C, the protocol is well known and defined by the industry as shown below.

**Table 18.1. Command Framing Protocol, by Interface**

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



**Figure 18.1. I<sup>2</sup>C Read Device ID Example**

### 18.2. 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 18.2. Command Framing Protocol, by Interface**

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

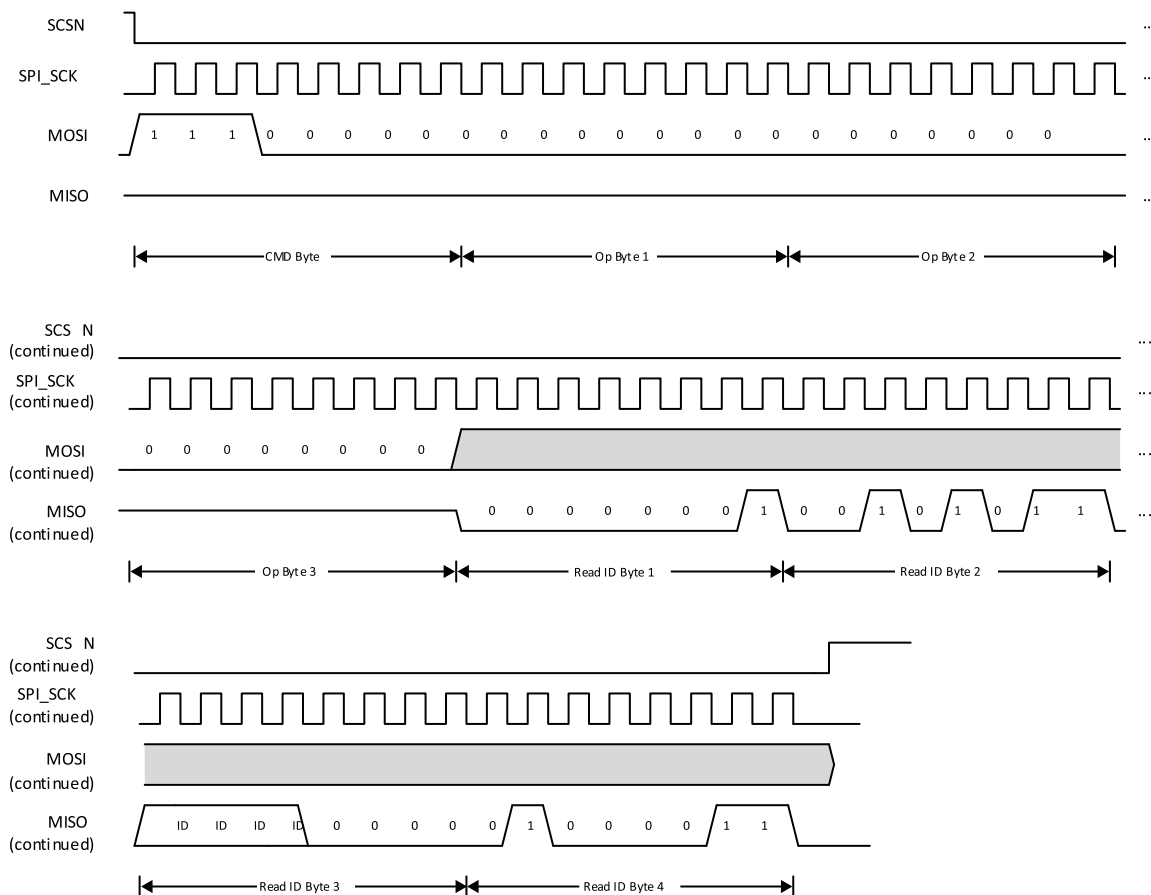


Figure 18.2. SSPI Read Device ID Example

### 18.3. WISHBONE Framing

To access the Flash Memory (UFM/Configuration) 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 18.3. Command Framing Protocol, by Interface

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

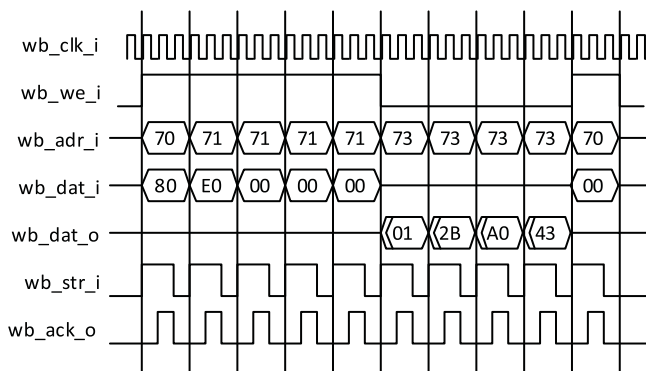


Figure 18.3. WISHBONE Read Device ID Example (-1200 HC Device)

## 19. UFM Write and Read Examples

The UFM and Configuration sectors support page-oriented read and write operations while erase operations are sector-based. Consistent with many Flash memory devices, byte-oriented operations are not supported. Also, as typical with Flash memory devices, attempting to modify a previously written location in Flash requires a read-modify-write operation on the smallest erasable Flash unit. In the case of MachXO2, the smallest erasable unit is the entire UFM sector or the entire Configuration Sector.

For example, to arbitrarily modify a byte value in the UFM, the user must:

1. Read and save all UFM data to an alternate location (for example, EBR);
2. Erase the UFM sector;
3. Modify the selected byte; and
4. Program the UFM page by page.

In some applications it may be appropriate to keep a working copy of the UFM contents in volatile Embedded Block RAM and update the non-volatile UFM at appropriate intervals. The following examples show the sequence of commands for writing and reading from UFM.

**Table 19.1. Write Two 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 UFM Address to 0000
—	—	—	—	—	—
—	—	+	—	—	—
4	W	C9	00 00 01	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Write UFM Page 0 Data
—	—	—	—	—	—
—	—	+	—	—	—
5	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—
—	—	—	—	—	(Repeat until Busy Flag not set, or wait 200 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 UFM Page 1 Data (Note: Address automatically incremented)
—	—	—	—	—	—
—	—	+	—	—	—
7	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—

Instruction Number	R/W1	CMD2	Operand	Data	Comment
—	—	—	—	—	(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 UFM/Configuration 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 19.2. Read One 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 UFM Address to 0001
—	—	—	—	—	—
—	—	+	—	—	—
4	W	CA	10 00 01	—	Read one page UFM (page 1) data
—	R	—	—	10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	—
—	—	—	—	—	—
—	—	+	—	—	—
5	W	26	00 00	—	Disable Configuration Interface
—	—	—	—	—	—
—	—	+	—	—	—
6	W	FF	—	—	Bypass (NOP)
—	—	—	—	—	—

**Notes:**

- When accessing UFM/Configuration 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 19.3. Read Two 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 UFM address to 0000
—	—	—	—	—	—
—	—	+	—	—	—
4	W	CA	10 00 03	—	Read two pages of UFM data, after one page of dummy bytes. <sup>3</sup>
—	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 UFM/Configuration 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.

## 20. Flash Memory Erase and Program Performance

Table 20.1. Flash Memory (UFM/Configuration) Performance in MachXO2 Devices<sup>1</sup>

		MachXO2 -256	MachXO2 -640	MachXO2 -640U	MachXO2 -1200	MachXO2 -1200U	MachXO2 -2000	MachXO2 -2000U	MachXO2 -4000	MachXO2 -7000
CFG Erase (tEraseCFG)	Typ Min.	400	600	800	800	1100	1100	1800	1800	2800
	Typ Max.	700	1100	1400	1400	1900	1900	3100	3100	4800
CFG Program (tProgramCFG)	All	130	270	500	500	740	740	1400	1400	2200
	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
UFM Erase (tEraseUFM)	Typ Min.	—	300	400	400	500	500	600	600	900
	Typ Max.	—	600	700	700	900	900	1000	1000	1600
UFM Program (tProgramUFM)	All	—	40	110	110	140	140	180	180	480
	1 page	—	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
tErase (max)	Note 2	9000	12000	15000	15000	15000	15000	30000	30000	30000

**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.

## 21. Erase/Program/Verify Time Calculation Example

Using the data above, it is possible to roughly calculate the time required to perform an Erase/Program/Verify operation. The calculation assumes nearly 100% bus utilization. Overhead required by bus master processes, if any, is not accounted for in the equation below.

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

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

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

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

Table 21.1. E/P/V Calculation parameters

	BusEff (Single Page Read)	BusEff <sup>3</sup> (Multi Page Read)	tBUSCLK
I <sup>2</sup> C	14	>12	2.5 us min
SPI	12	> 8	0.015 us min
WB	5.25	> 3	0.020 us min

**Notes:**

1. Sector erase times are additive. If a sector (for example, CFG) is not erased, its erase time is 0.
2. Data transfer time is insignificant to tProgram for high-speed data protocols. To account for slow bus speeds (for example, I2C) multiply tVerify by 2.
3. Bus efficiency approaches this value as number of read pages increases.

## Technical Support Assistance

Submit a technical support case through [www.latticesemi.com/techsupport](http://www.latticesemi.com/techsupport).

For frequently asked questions, refer to the Lattice Answer Database at [www.latticesemi.com/en/Support/AnswerDatabase](http://www.latticesemi.com/en/Support/AnswerDatabase).



## Revision History

### Revision 2.8, February 2023

Section	Change Summary
Timer Counter Simulation Model	<p><a href="#">Table 16.20. Program Configuration Flash (0x70)</a>:</p> <ul style="list-style-type: none"> <li>changed the Operands (Hex) from <i>00 00 01</i> to <i>00 00 00</i> in the table;</li> <li>updated the example below the table;</li> <li>removed <i>Operands (0x00 00 00) are equivalent to (0x00 00 01)</i> from the notes.</li> </ul>

### Revision 2.7, October 2021

Section	Change Summary
I2C Registers	<ul style="list-style-type: none"> <li>Added a note to define tTCL as the I2C clock period.</li> <li>Updated the ***optional: note in Figure 3.1. I2C Master Read/Write Example (via WISHBONE) include 'Where tTCL_period represents the I2C clock period'.</li> </ul>
Flash Memory (UFM/Configuration) Access	Updated Table 16.23. Read Configuration Flash (0x73) (WISHBONE) and Table 16.34. Read UFM (0xCA) (WISHBONE) to add option to exclude dummy bytes between frames.
Erase/Program/Verify Time Calculation Example	Changed the equation for tProgram from '0.2 μs * number of Pages to program <sup>2</sup> ' to '0.2 ms * number of Pages to program <sup>2</sup> '.

### Revision 2.6, August 2021

Section	Change Summary
Flash Memory Erase and Program Performance	Updated Min and Max labels in Table 20.1. Flash Memory (UFM/Configuration) Performance in MachXO2 Devices to Typ Min and Typ Max labels respectively.

### Revision 2.5, November 2019

Section	Change Summary
All	<ul style="list-style-type: none"> <li>Changed document number from TN1246 to FPGA-RD-02163.</li> <li>Updated document template.</li> </ul>
Disclaimers	Added this section.

### Revision 2.4, November 2016

Section	Change Summary
I2C Registers	Corrected figure reference in Table 3.7., I2C Status (Primary/Secondary).
Flash Memory (UFM/Configuration) Access	<ul style="list-style-type: none"> <li>Corrected Read UFM description in Table 16.10., UFM (Sector 1) Commands.</li> <li>Corrected Read Flash description in Table 16.11., Configuration Flash (Sector 0) Commands.</li> </ul> <p>Updated document per Product Bulletin PB1381.</p> <ul style="list-style-type: none"> <li>Updated Table 16.22., Read Configuration Flash (0x73) (SPI).</li> <li>Updated Table 16.23., Read Configuration Flash (0x73) (I2C/SPI).</li> <li>Added Table 16.24., Read Configuration Flash (0x73) (WISHBONE).</li> <li>Updated Table 16.33., Read UFM (0xCA) (SPI).</li> <li>Updated Table 16.34., Read UFM (0xCA) (I2C/SPI).</li> <li>Added Table 16.35., Read UFM (0xCA) (WISHBONE).</li> </ul>

### Revision 2.3, March 2016

Section	Change Summary
WISHBONE Read Cycle	Updated this section. Added information on avoiding simulation mismatch in functional simulations.
I2C Registers	Updated this section. Modified Figure 3.1., I2C Master Read/Write Example (via WISHBONE).
Flash Memory Erase and Program Performance	Updated this section. Added tErase (max) values to Table 20.1., Flash Memory (UFM/Configuration) Performance in MachXO2 Devices.

## Revision 2.2, September 2015

Section	Change Summary
I2C Registers	Updated this section. Modified description of RARC and TROE.
SPI Registers section	Updated this. Modified description of TXEDGE and CPOL.
SPI Timing Diagrams	Updated this section. Corrected the following diagrams: <ul style="list-style-type: none"> <li>Figure 12.1., SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)</li> <li>Figure 12.2., SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)</li> <li>Figure 12.3., SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)</li> <li>Figure 12.4., SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)</li> </ul>
Flash Memory (UFM/Configuration) Access Ports	Updated this section. Added disabled feature in Note.
Command Framing	Updated this section. Corrected device name in Figure 18.1., I2C Read Device ID Example.
Technical Support Assistance	Updated this section.

## Revision 2.1, February 2015

Section	Change Summary
WISBONE Bus Interface	Updated this section. Revised In Table 1.2., WISHBONE Slave Interface Signals of the EFB Module. Added details to the wb_clk_i signal name description.
Hardened I2C IP Cores	Updated this section. Added new EFB instantiation requirement for I2C configuration port access per Product Bulletin PB1412.
I2C Registers	Updated this section. <ul style="list-style-type: none"> <li>Changed Figure 3.1., I2C Master Read/Write Example (via WISHBONE).</li> <li>Revised SDA_DEL_SEL[1:0] description.</li> </ul>
SPI Registers	Updated this section. Corrected CPOL description.
SPI Timing Diagrams	Updated this section. Changed the following figures: <ul style="list-style-type: none"> <li>Figure 12.1., SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)</li> <li>Figure 12.2., SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)</li> <li>Figure 12.3., SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)</li> <li>Figure 12.4., SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)</li> </ul>

## Revision 2.0, November 2014

Section	Change Summary
I2C Registers	Updated this section. Added note to SRW description under Table 3.7., I2C Status (Primary/Secondary).
Flash Memory (UFM/Configuration) Access	Updated this section. <ul style="list-style-type: none"> <li>Updated Table 16.11., Configuration Flash (Sector 0) Commands. Updated Erase command description.</li> <li>Updated information on Erase Feature sector under Table 16.14., Erase Flash (0x0E).</li> <li>Updated note under Table 16.42., Program Feature Row (0xE4)</li> <li>Updated note under Table 16.45., Program FEABITs (0xF8).</li> </ul>

## Revision 1.9, June 2014

Section	Change Summary
SPI Timing Diagrams	Updated the following figures: <ul style="list-style-type: none"> <li>Figure 12.1., SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)</li> <li>Figure 12.2., SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)</li> <li>Figure 12.3., SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)</li> <li>Figure 12.4., SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1).</li> </ul>

**Revision 1.8, February 2014**

Section	Change Summary
Flash Memory (UFM/Configuration) Access	Updated Tables 16.22. – 16.24., Read Configuration Flash (0x73) (I2C/WISHBONE/ SPI). Revised Data Size formatting when num_pages>1.

**Revision 1.7, January 2014**

Section	Change Summary
Flash Memory (UFM/Configuration) Access	Updated Table 16.25., Enable Configuration Interface (Transparent) (0x74). Changed Operands (Hex) data and added information to Notes and Example sections.
Flash Memory Erase and Program Performance	Updated Table 20.1., Flash Memory (UFM/Configuration) Performance in MachXO2 Devices.
All	Updated Erase/Program/Verify equation in Erase/Program/Verify Time Calculation Example section.

**Revision 1.6, January 2014**

Section	Change Summary
SPI Timing Diagrams	<ul style="list-style-type: none"> <li>Updated Figure 17-2.</li> <li>Updated Figure 17-4.</li> <li>Updated Figure 17-5.</li> <li>Updated Figure 17-26.</li> </ul>
Revision History	Corrected version 1.5.

**Revision 1.5, December 2013**

Section	Change Summary
SPI Timing Diagrams	Updated Figure 17-2.
Typical I2C Transactions	Added this section.
Command Framing	Added and reorganized this section.
I2C Framing	Moved this section under Command Framing section.
SPI Timing Diagrams	Added Figure 17-18.
Typical SPI Transactions	Added this section.
SPI Framing	Moved this section under Command Framing section.
Flash Memory (UFM/Configuration) Access	Changed Operands (Hex) value in Table 16.30. and added description.
WISHBONE Framing	Moved this section under Command Framing section.
Flash Memory Erase and Program Performance and Erase/Program/Verify Time Calculation Example	Moved these sections after UFM Write and Read Examples section
SPI Registers	Changed CPOL = 1 description in this section.
SPI Timing Diagrams	<ul style="list-style-type: none"> <li>Updated Figure 17-23. Changed second instance of SPISR[TRDY] to SPISR[ROE].</li> <li>Updated Figure 17-24.</li> </ul>

**Revision 1.4, September 2013**

Section	Change Summary
SPI Registers	Updated CPOL definition in this section. Changed SCK idle state from low to high.
SPI Timing Diagrams	Updated SPI Control Timing figures in this section.
All	<ul style="list-style-type: none"> <li>Corrected 0xCA num_pages and Data Size.</li> <li>Removed I2C Clock-Stretching feature per PCN #10A-13.</li> </ul>
	Updated the following figures: <ul style="list-style-type: none"> <li>I2C Master Read/Write Example (via WISHBONE)</li> <li>I2C Slave Read/Write Example (via WISHBONE)</li> <li>EFB Master – I2C Write</li> </ul>

Section	Change Summary
	<ul style="list-style-type: none"> <li>EFB Master – I2C Read</li> </ul>
I2C Registers	Corrected I2C General Call Data Register (Primary/Secondary) table number.
Technical Support Assistance	Updated Technical Support Assistance information

#### Revision 1.3, April 2013

Section	Change Summary
Flash Memory (UFM/Configuration) Access	<ul style="list-style-type: none"> <li>Read Configuration Flash (0x73) (I2C/WISHBONE/SPI) table – Corrected table title.</li> <li>Read Feature Row (0xE7) table – Updated Data Format in the table and description.</li> <li>Updated examples in the Read UFM (0xCA) (WISHBONE/SPI/I2C) table.</li> <li>Added note: SECURITY and SECURITY PLUS commands are mutually exclusive.</li> </ul>
I2C Registers	Updated information in the I2C Master Read/Write Example (via WISHBONE) figure.
UFM Write and Read Examples	Corrected BUSY wait times (1000ns -> 200ns) in Write Two UFM Pages table.
Interface to Configuration Flash	Updated Basic Configuration Flash Update Example, changed "Wait for !BUSY" to "Wait for tREFRESH" in last step.
All	<ul style="list-style-type: none"> <li>Added Wait for tREFRESH caution to Refresh command description.</li> <li>Clarified Secondary I2C non-idle reset issue after REFRESH.</li> <li>Added Erase/Program/Verify time calculation example.</li> <li>Updated (decreased) the maximum WISHBONE clock rate for page reads from 36 MHz to 16.6 MHz.</li> </ul>

#### Revision 1.2, October 2012

Section	Change Summary
All	<ul style="list-style-type: none"> <li>Added restriction: Primary port can be used as Configuration/UFM port or as a user port, but not both.</li> <li>Added restriction: Primary I2C port is unavailable while in ISC_ENABLE_X (transparent) configuration access mode.</li> </ul>

#### Revision 1.1, August 2012

Section	Change Summary
All	Timer/Counter Control 1 table – Corrected names of four LSBs. Updated Timer/Counter Control 0 table and Timer/Counter Control 1 table.
Flash Memory (UFM/Configuration) Access	<ul style="list-style-type: none"> <li>Program Feature Row (0xE4) table – Updated Data Size and Data Format (Hex) columns and text below table for ss, uu and cc cc cc cc. Added example.</li> <li>Read Feature Row (0xE7) table – Updated CMD (Hex) column.</li> <li>Read FEABITs (0xFB) table – Removed example below table.</li> <li>Read USERCODE (0xC0) table – Data Size column updated. "EN Required" value changed from "N" to "Y/N" and example text updated.</li> <li>Device ID Table – Updated Device Name column.</li> <li>Read Status Register (0x3C) table – Updated Data Format column.</li> <li>Verify Device ID Code (0xE2) table – "EN Required" value changed from "N" to "Y" and example text updated</li> </ul>
Interface to Configuration Flash	Updated Basic Configuration Flash Update Example diagram.

#### Revision 1.0, June 2012

Section	Change Summary
All	Initial release.



[www.latticesemi.com](http://www.latticesemi.com)