MachXO2 Family Handbook

HB1010 Version 03.8, May 2013

MachXO2 Family Handbook Table of Contents

May 2013

Section I. MachXO2 Family Data Sheet

DS1035 Version 02.0, January 2013

MachXO2 Family Data Sheet Introduction

January 2013 Data Sheet DS1035

Features

- **Flexible Logic Architecture**
  - Six devices with 256 to 6864 LUT4s and 19 to 335 I/Os
- **Ultra Low Power Devices**
  - Advanced 65 nm low power process
  - As low as 19 µW standby power
  - Programmable low swing differential I/Os
  - Stand-by mode and other power saving options
- **Embedded and Distributed Memory**
  - Up to 240 Kbits sysMEM™ Embedded Block RAM
  - Up to 54 Kbits Distributed RAM
  - Dedicated FIFO control logic
- **On-Chip User Flash Memory**
  - Up to 256 Kbits of User Flash Memory
  - 100,000 write cycles
  - Accessible through WISHBONE, SPI, I^2C and JTAG interfaces
  - Can be used as soft processor PROM or as Flash memory
- **Pre-Engineered Source Synchronous I/O**
  - DDR registers in I/O cells
  - Dedicated gearing logic
  - 7:1 Gearing for Display I/Os
  - Generic DDR, DDRX2, DDRX4
  - Dedicated DDR/DDR2/LPDDR memory with DQS support
- **High Performance, Flexible I/O Buffer**
  - Programmable sysIO™ buffer supports wide range of interfaces:
    - LVCMOS 3.3/2.5/1.8/1.5/1.2
    - LVTTL
    - PCI
    - LVDS, Bus-LVDS, MLVDS, RSDS, LVPECL
    - SSTL 25/18
    - HSTL 18
    - Schmitt trigger inputs, up to 0.5V hysteresis
  - I/Os support hot socketing
  - On-chip differential termination
  - Programmable pull-up or pull-down mode
- **Flexible On-Chip Clocking**
  - Eight primary clocks
  - Up to two edge clocks for high-speed I/O interfaces (top and bottom sides only)
  - Up to two analog PLLs per device with fractional-n frequency synthesis
    - Wide input frequency range (10 MHz to 400 MHz)
- **Non-volatile, Infinitely Reconfigurable**
  - Instant-on – powers up in microseconds
  - Single-chip, secure solution
  - Programmable through JTAG, SPI or I^2C
  - Supports background programming of non-volatile memory
  - Optional dual boot with external SPI memory
- **TransFR™ Reconfiguration**
  - In-field logic update while system operates
- **Enhanced System Level Support**
  - On-chip hardened functions: SPI, I^2C, timer/counter
  - On-chip oscillator with 5.5% accuracy
  - Unique TraceID for system tracking
  - One Time Programmable (OTP) mode
  - Single power supply with extended operating range
  - IEEE Standard 1149.1 boundary scan
  - IEEE 1532 compliant in-system programming
- **Broad Range of Package Options**
  - TQFP, WLCSP, ucBGA, csBGA, caBGA, ftBGA, fpBGA, QFN package options
  - Small footprint package options
    - As small as 2.5x2.5mm
  - Density migration supported
  - Advanced halogen-free packaging

Table 1-1. MachXO2™ Family Selection Guide

		XO2-256	XO2-640	XO2-640U[1]	XO2-1200	XO2-1200U[1]	XO2-2000	XO2-2000U[1]	XO2-4000	XO2-7000
LUTs		256	640	640	1280	1280	2112	2112	4320	6864
Distributed RAM (Kbits)		2	5	5	10	10	16	16	34	54
EBR SRAM (Kbits)		0	18	64	64	74	74	92	92	240
Number of EBR SRAM Blocks (9 Kbits/block)		0	2	7	7	8	8	10	10	26
UFM (Kbits)		0	24	64	64	80	80	96	96	256
Device Options	HC[2]									
	HE[3]									
	ZE[4]									
Number of PLLs		0	0	1	1	1	1	2	2	2
Hardened Functions: I^2C SPI Timer/Counter		 2 1 1	 2 1 1	 2 1 1	 2 1 1	 2 1 1	 2 1 1	 2 1 1	 2 1 1	 2 1 1
Packages		**I/Os**								
25 WLCSP[5] (2.5 x 2.5mm, 0.4mm)					18					
32 QFN[6] (5 x 5mm, 0.5mm)		21								
64 ucBGA (4 x 4mm, 0.4mm)		44								
100 TQFP (14 x 14mm)		55	78		79		79			
132 csBGA (8 x 8mm, 0.5mm)		55	79		104		104		104	
144 TQFP (20 x 20mm)				107	107		111		114	114
184 csBGA[7] (8 x 8mm, 0.5mm)									150	
256 caBGA (14 x 14mm, 0.8mm)							206		206	206
256 ftBGA (17 x 17mm, 1.0mm)						206	206		206	206
332 caBGA (17 x 17mm, 0.8mm)									274	278
484 fpBGA (23 x 23mm, 1.0mm)								278	278	334

1. Ultra high I/O device.
2. High performance with regulator – V_{CC} = 2.5V, 3.3V
3. High performance without regulator – V_{CC} = 1.2V
4. Low power without regulator – V_{CC} = 1.2V
5. WLCSP package only available for ZE devices.
6. QFN package only available for HC and ZE devices.
7. 184 csBGA package only available for HE devices.

Introduction

The MachXO2 family of ultra low power, instant-on, non-volatile PLDs has six devices with densities ranging from 256 to 6864 Look-Up Tables (LUTs). In addition to LUT-based, low-cost programmable logic these devices feature Embedded Block RAM (EBR), Distributed RAM, User Flash Memory (UFM), Phase Locked Loops (PLLs), pre-engineered source synchronous I/O support, advanced configuration support including dual-boot capability and hardened versions of commonly used functions such as SPI controller, I^2C controller and timer/counter. These features allow these devices to be used in low cost, high volume consumer and system applications.

The MachXO2 devices are designed on a 65nm non-volatile low power process. The device architecture has several features such as programmable low swing differential I/Os and the ability to turn off I/O banks, on-chip PLLs

and oscillators dynamically. These features help manage static and dynamic power consumption resulting in low static power for all members of the family.

The MachXO2 devices are available in two versions – ultra low power (ZE) and high performance (HC and HE) devices. The ultra low power devices are offered in three speed grades -1, -2 and -3, with -3 being the fastest. Similarly, the high-performance devices are offered in three speed grades: -4, -5 and -6, with -6 being the fastest. HC devices have an internal linear voltage regulator which supports external V_{CC} supply voltages of 3.3V or 2.5V. ZE and HE devices only accept 1.2V as the external V_{CC} supply voltage. With the exception of power supply voltage all three types of devices (ZE, HC and HE) are functionally compatible and pin compatible with each other.

The MachXO2 PLDs are available in a broad range of advanced halogen-free packages ranging from the space saving 2.5x2.5 mm WLCSP to the 23x23 mm fpBGA. MachXO2 devices support density migration within the same package. Table 1-1 shows the LUT densities, package and I/O options, along with other key parameters.

The pre-engineered source synchronous logic implemented in the MachXO2 device family supports a broad range of interface standards, including LPDDR, DDR, DDR2 and 7:1 gearing for display I/Os.

The MachXO2 devices offer enhanced I/O features such as drive strength control, slew rate control, PCI compatibility, bus-keeper latches, pull-up resistors, pull-down resistors, open drain outputs and hot socketing. Pull-up, pull-down and bus-keeper features are controllable on a “per-pin” basis.

A user-programmable internal oscillator is included in MachXO2 devices. The clock output from this oscillator may be divided by the timer/counter for use as clock input in functions such as LED control, key-board scanner and similar state machines.

The MachXO2 devices also provide flexible, reliable and secure configuration from on-chip Flash memory. These devices can also configure themselves from external SPI Flash or be configured by an external master through the JTAG test access port or through the I^2C port. Additionally, MachXO2 devices support dual-boot capability (using external Flash memory) and remote field upgrade (TransFR) capability.

Lattice provides a variety of design tools that allow complex designs to be efficiently implemented using the MachXO2 family of devices. Popular logic synthesis tools provide synthesis library support for MachXO2. Lattice design tools use the synthesis tool output along with the user-specified preferences and constraints to place and route the design in the MachXO2 device. These tools extract the timing from the routing and back-annotate it into the design for timing verification.

Lattice provides many pre-engineered IP (Intellectual Property) LatticeCORE™ modules, including a number of reference designs licensed free of charge, optimized for the MachXO2 PLD family. By using these configurable soft core IP cores as standardized blocks, users are free to concentrate on the unique aspects of their design, increasing their productivity.

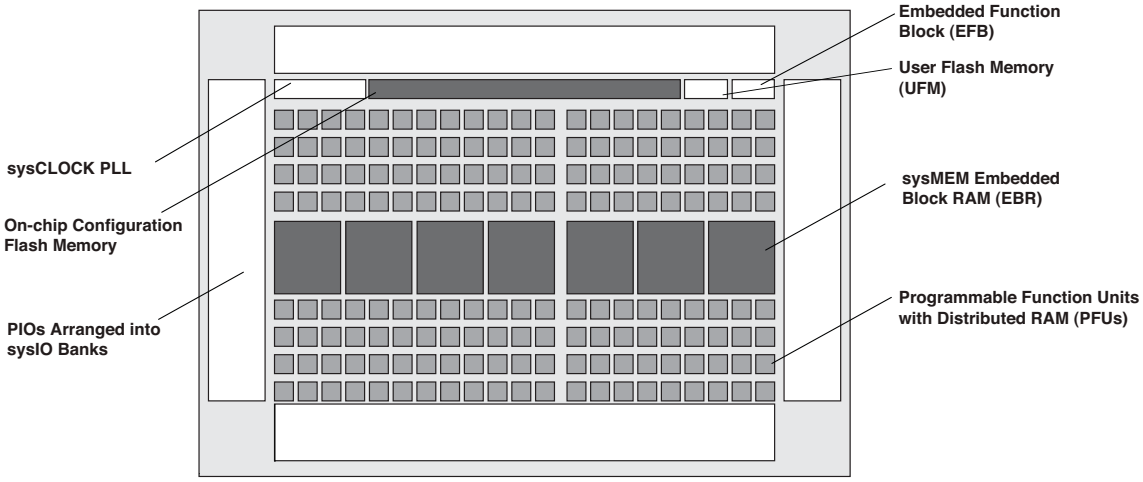
MachXO2 Family Data Sheet Architecture

January 2013 Data Sheet DS1035

Architecture Overview

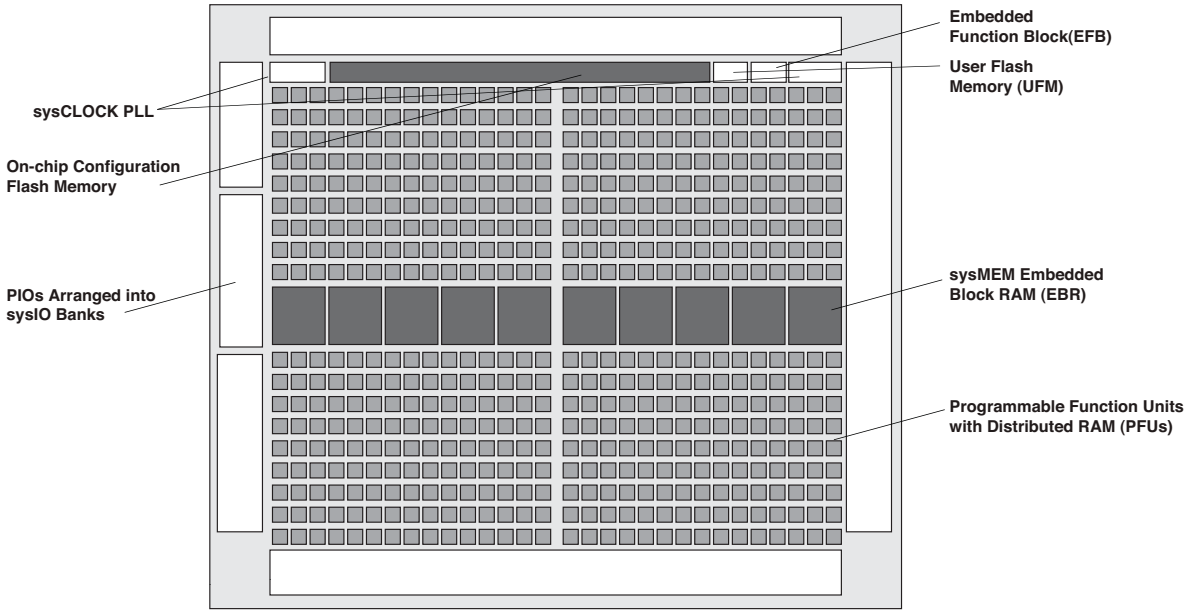
The MachXO2 family architecture contains an array of logic blocks surrounded by Programmable I/O (PIO). The larger logic density devices in this family have sysCLOCK™ PLLs and blocks of sysMEM Embedded Block RAM (EBRs). Figures 2-1 and 2-2 show the block diagrams of the various family members.

Figure 2-1. Top View of the MachXO2-1200 Device



Note: MachXO2-256, and MachXO2-640/U are similar to MachXO2-1200. MachXO2-256 has a lower LUT count and no PLL or EBR blocks. MachXO2-640 has no PLL, a lower LUT count and two EBR blocks. MachXO2-640U has a lower LUT count, one PLL and seven EBR blocks.

Figure 2-2. Top View of the MachXO2-4000 Device



Note: MachXO2-1200U, MachXO2-2000/U and MachXO2-7000 are similar to MachXO2-4000. MachXO2-1200U and MachXO2-2000 have a lower LUT count, one PLL, and eight EBR blocks. MachXO2-2000U has a lower LUT count, two PLLs, and 10 EBR blocks. MachXO2-7000 has a higher LUT count, two PLLs, and 26 EBR blocks.

The logic blocks, Programmable Functional Unit (PFU) and sysMEM EBR blocks, are arranged in a two-dimensional grid with rows and columns. Each row has either the logic blocks or the EBR blocks. The PIO cells are located at the periphery of the device, arranged in banks. The PFU contains the building blocks for logic, arithmetic, RAM, ROM, and register functions. The PIOs utilize a flexible I/O buffer referred to as a sysIO buffer that supports operation with a variety of interface standards. The blocks are connected with many vertical and horizontal routing channel resources. The place and route software tool automatically allocates these routing resources.

In the MachXO2 family, the number of sysIO banks varies by device. There are different types of I/O buffers on the different banks. Refer to the details in later sections of this document. The sysMEM EBRs are large, dedicated fast memory blocks; these blocks are found in MachXO2-640/U and larger devices. These blocks can be configured as RAM, ROM or FIFO. FIFO support includes dedicated FIFO pointer and flag “hard” control logic to minimize LUT usage.

The MachXO2 architecture also provides up to two sysCLOCK Phase Locked Loop (PLL) blocks on MachXO2-640U, MachXO2-1200/U and larger devices. These blocks are located at the ends of the on-chip Flash block. The PLLs have multiply, divide, and phase shifting capabilities that are used to manage the frequency and phase relationships of the clocks.

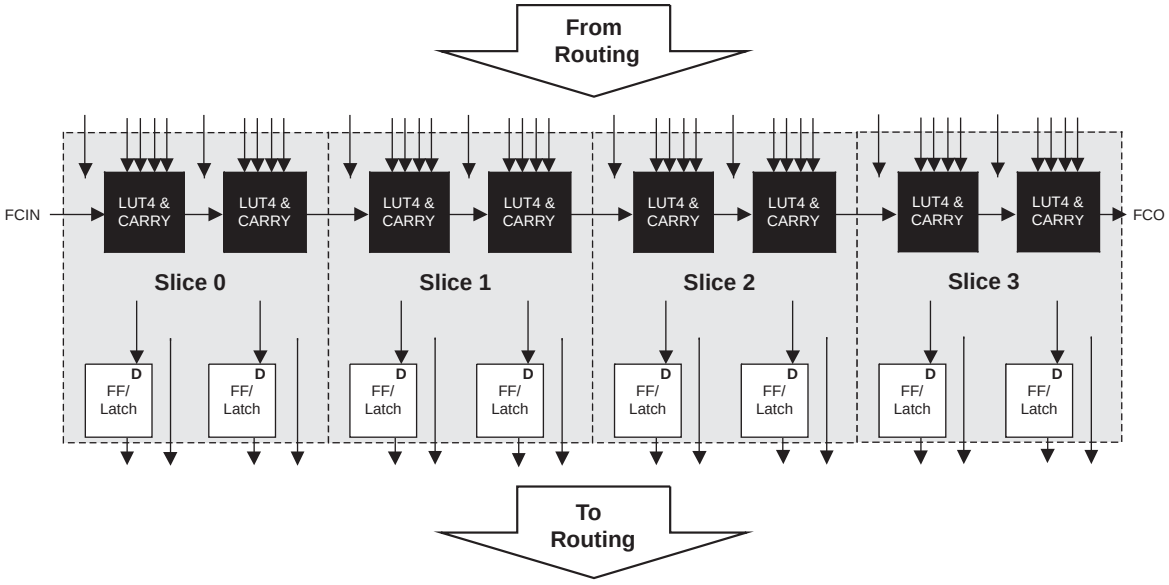
MachXO2 devices provide commonly used hardened functions such as SPI controller, I^2C controller and timer/counter. MachXO2-640/U and higher density devices also provide User Flash Memory (UFM). These hardened functions and the UFM interface to the core logic and routing through a WISHBONE interface. The UFM can also be accessed through the SPI, I^2C and JTAG ports.

Every device in the family has a JTAG port that supports programming and configuration of the device as well as access to the user logic. The MachXO2 devices are available for operation from 3.3V, 2.5V and 1.2V power supplies, providing easy integration into the overall system.

PFU Blocks

The core of the MachXO2 device consists of PFU blocks, which can be programmed to perform logic, arithmetic, distributed RAM and distributed ROM functions. Each PFU block consists of four interconnected slices numbered 0 to 3 as shown in Figure 2-3. Each slice contains two LUTs and two registers. There are 53 inputs and 25 outputs associated with each PFU block.

Figure 2-3. PFU Block Diagram



Slices

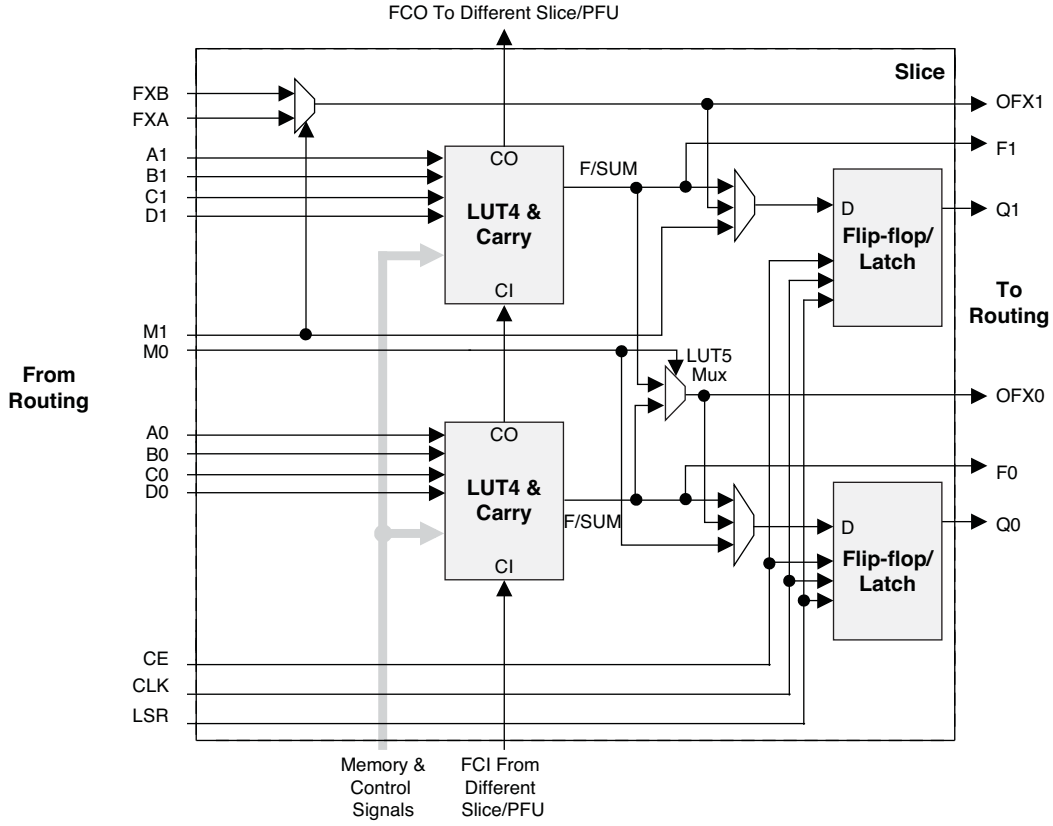
Slices 0-3 contain two LUT4s feeding two registers. Slices 0-2 can be configured as distributed memory. Table 2-1 shows the capability of the slices in PFU blocks along with the operation modes they enable. In addition, each PFU contains logic that allows the LUTs to be combined to perform functions such as LUT5, LUT6, LUT7 and LUT8. The control logic performs set/reset functions (programmable as synchronous/ asynchronous), clock select, chip-select and wider RAM/ROM functions.

Table 2-1. Resources and Modes Available per Slice

Slice	PFU Block	
	Resources	Modes
Slice 0	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM
Slice 1	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM
Slice 2	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM
Slice 3	2 LUT4s and 2 Registers	Logic, Ripple, ROM

Figure 2-4 shows an overview of the internal logic of the slice. The registers in the slice can be configured for positive/negative and edge triggered or level sensitive clocks. All slices have 15 inputs from routing and one from the carry-chain (from the adjacent slice or PFU). There are seven outputs: six for routing and one to carry-chain (to the adjacent PFU). Table 2-2 lists the signals associated with Slices 0-3.

Figure 2-4. Slice Diagram



For Slices 0 and 1, memory control signals are generated from Slice 2 as follows:

- WCK is CLK
- WRE is from LSR
- DI[3:2] for Slice 1 and DI[1:0] for Slice 0 data from Slice 2
- WAD [A:D] is a 4-bit address from slice 2 LUT input

Table 2-2. Slice Signal Descriptions

Function	Type	Signal Names	Description
Input	Data signal	A0, B0, C0, D0	Inputs to LUT4
Input	Data signal	A1, B1, C1, D1	Inputs to LUT4
Input	Multi-purpose	M0/M1	Multi-purpose input
Input	Control signal	CE	Clock enable
Input	Control signal	LSR	Local set/reset
Input	Control signal	CLK	System clock
Input	Inter-PFU signal	FCIN	Fast carry in[1]
Output	Data signals	F0, F1	LUT4 output register bypass signals
Output	Data signals	Q0, Q1	Register outputs
Output	Data signals	OFX0	Output of a LUT5 MUX
Output	Data signals	OFX1	Output of a LUT6, LUT7, LUT8[2] MUX depending on the slice
Output	Inter-PFU signal	FCO	Fast carry out[1]

1. See Figure 2-3 for connection details.
2. Requires two PFUs.

Modes of Operation

Each slice has up to four potential modes of operation: Logic, Ripple, RAM and ROM.

Logic Mode

In this mode, the LUTs in each slice are configured as 4-input combinatorial lookup tables. A LUT4 can have 16 possible input combinations. Any four input logic functions can be generated by programming this lookup table. Since there are two LUT4s per slice, a LUT5 can be constructed within one slice. Larger look-up tables such as LUT6, LUT7 and LUT8 can be constructed by concatenating other slices. Note LUT8 requires more than four slices.

Ripple Mode

Ripple mode supports the efficient implementation of small arithmetic functions. In Ripple mode, the following functions can be implemented by each slice:

- Addition 2-bit
- Subtraction 2-bit
- Add/subtract 2-bit using dynamic control
- Up counter 2-bit
- Down counter 2-bit
- Up/down counter with asynchronous clear
- Up/down counter with preload (sync)
- Ripple mode multiplier building block
- Multiplier support
- Comparator functions of A and B inputs
  - A greater-than-or-equal-to B
  - A not-equal-to B
  - A less-than-or-equal-to B

Ripple mode includes an optional configuration that performs arithmetic using fast carry chain methods. In this configuration (also referred to as CCU2 mode) two additional signals, Carry Generate and Carry Propagate, are generated on a per-slice basis to allow fast arithmetic functions to be constructed by concatenating slices.

RAM Mode

In this mode, a 16x4-bit distributed single port RAM (SPR) can be constructed by using each LUT block in Slice 0 and Slice 1 as a 16x1-bit memory. Slice 2 is used to provide memory address and control signals. A 16x2-bit Pseudo Dual Port RAM (PDPR) memory is created by using one slice as the read-write port and the other companion slice as the read-only port.

MachXO2 devices support distributed memory initialization.

The Lattice design tools support the creation of a variety of different size memories. Where appropriate, the software will construct these using distributed memory primitives that represent the capabilities of the PFU. Table 2-3 shows the number of slices required to implement different distributed RAM primitives. For more information about using RAM in MachXO2 devices, please see TN1201, Memory Usage Guide for MachXO2 Devices.

Table 2-3. Number of Slices Required For Implementing Distributed RAM

	SPR 16x4	PDPR 16x4
Number of slices	3	3

Note: SPR = Single Port RAM, PDPR = Pseudo Dual Port RAM

ROM Mode

ROM mode uses the LUT logic; hence, slices 0-3 can be used in ROM mode. Preloading is accomplished through the programming interface during PFU configuration.

For more information on the RAM and ROM modes, please refer to TN1201, Memory Usage Guide for MachXO2 Devices.

Routing

There are many resources provided in the MachXO2 devices to route signals individually or as buses with related control signals. The routing resources consist of switching circuitry, buffers and metal interconnect (routing) segments.

The inter-PFU connections are made with three different types of routing resources: x1 (spans two PFUs), x2 (spans three PFUs) and x6 (spans seven PFUs). The x1, x2, and x6 connections provide fast and efficient connections in the horizontal and vertical directions.

The design tools take the output of the synthesis tool and places and routes the design. Generally, the place and route tool is completely automatic, although an interactive routing editor is available to optimize the design.

Clock/Control Distribution Network

Each MachXO2 device has eight clock inputs (PCLK [T, C] [Banknum]_[2..0]) – three pins on the left side, two pins each on the bottom and top sides and one pin on the right side. These clock inputs drive the clock nets. These eight inputs can be differential or single-ended and may be used as general purpose I/O if they are not used to drive the clock nets. When using a single ended clock input, only the PCLKT input can drive the clock tree directly.

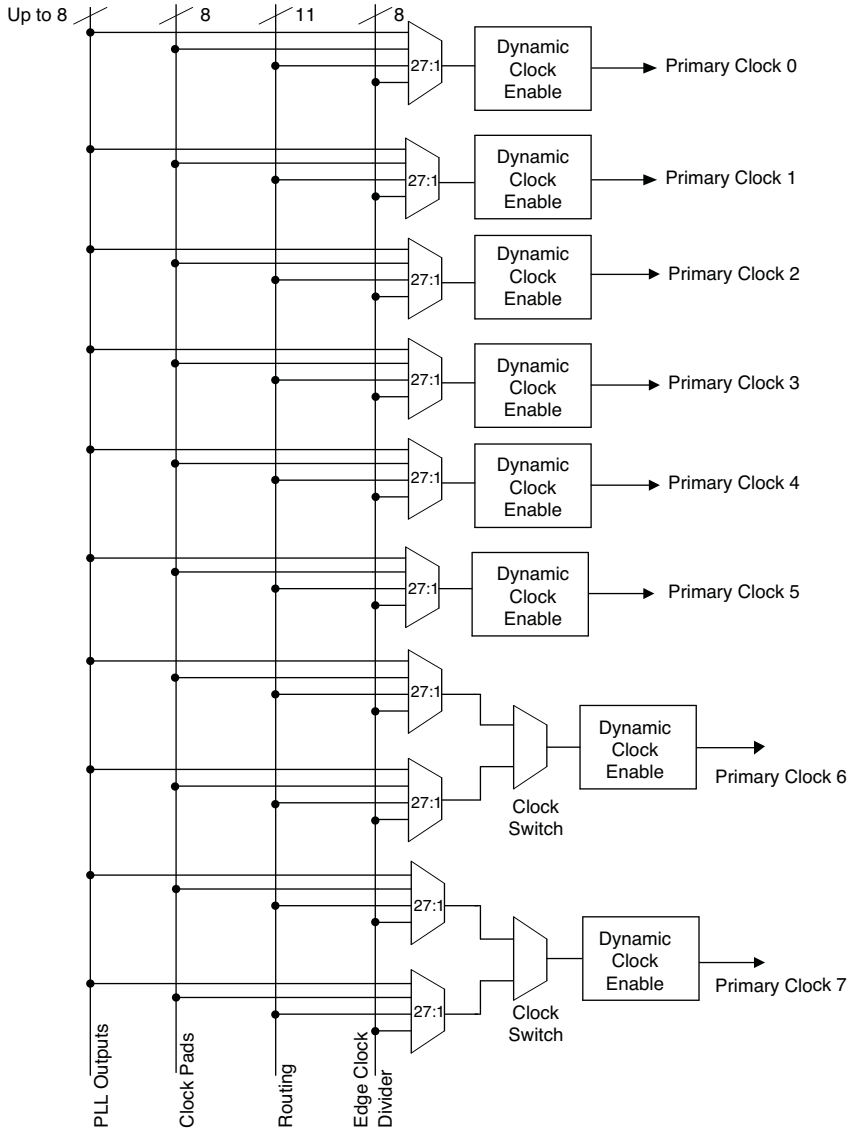
The MachXO2 architecture has three types of clocking resources: edge clocks, primary clocks and secondary high fanout nets. MachXO2-640U, MachXO2-1200/U and higher density devices have two edge clocks each on the top and bottom edges. Lower density devices have no edge clocks. Edge clocks are used to clock I/O registers and have low injection time and skew. Edge clock inputs are from PLL outputs, primary clock pads, edge clock bridge outputs and CIB sources.

The eight primary clock lines in the primary clock network drive throughout the entire device and can provide clocks for all resources within the device including PFUs, EBRs and PICs. In addition to the primary clock signals, MachXO2 devices also have eight secondary high fanout signals which can be used for global control signals, such as clock enables, synchronous or asynchronous clears, presets, output enables, etc. Internal logic can drive the global clock network for internally-generated global clocks and control signals.

The maximum frequency for the primary clock network is shown in the MachXO2 External Switching Characteristics table.

The primary clock signals for the MachXO2-256 and MachXO2-640 are generated from eight 17:1 muxes The available clock sources include eight I/O sources and 9 routing inputs. Primary clock signals for the MachXO2-640U, MachXO2-1200/U and larger devices are generated from eight 27:1 muxes The available clock sources include eight I/O sources, 11 routing inputs, eight clock divider inputs and up to eight sysCLOCK PLL outputs.

Figure 2-5. Primary Clocks for MachXO2 Devices

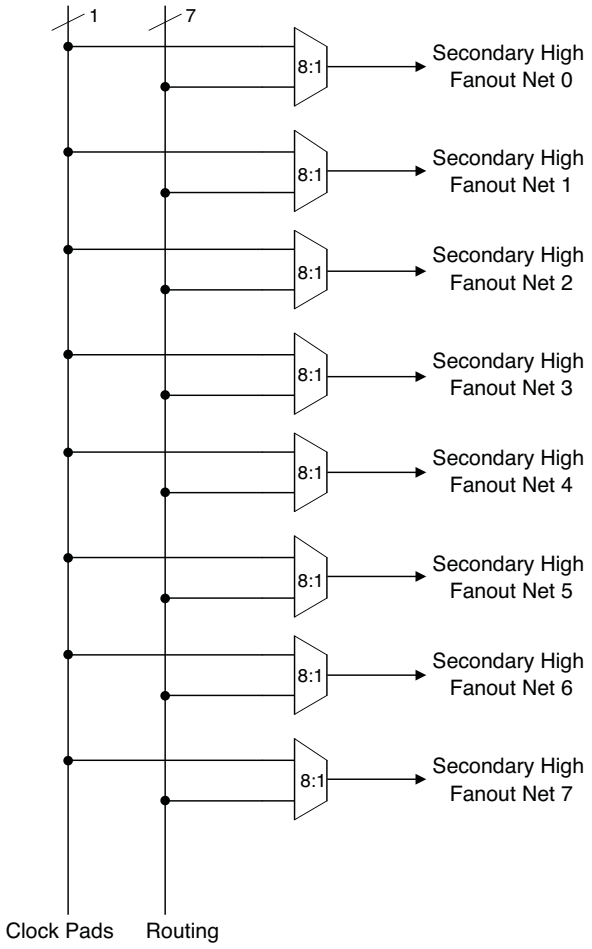


Primary clocks for MachXO2-640U, MachXO2-1200/U and larger devices.

Note: MachXO2-640 and smaller devices do not have inputs from the Edge Clock Divider or PLL and fewer routing inputs. These devices have 17:1 muxes instead of 27:1 muxes.

Eight secondary high fanout nets are generated from eight 8:1 muxes as shown in Figure 2-6. One of the eight inputs to the secondary high fanout net input mux comes from dual function clock pins and the remaining seven come from internal routing. The maximum frequency for the secondary clock network is shown in MachXO2 External Switching Characteristics table.

Figure 2-6. Secondary High Fanout Nets for MachXO2 Devices



sysCLOCK Phase Locked Loops (PLLs)

The sysCLOCK PLLs provide the ability to synthesize clock frequencies. The MachXO2-640U, MachXO2-1200/U and larger devices have one or more sysCLOCK PLL. CLKI is the reference frequency input to the PLL and its source can come from an external I/O pin or from internal routing. CLKFB is the feedback signal to the PLL which can come from internal routing or an external I/O pin. The feedback divider is used to multiply the reference frequency and thus synthesize a higher frequency clock output.

The MachXO2 sysCLOCK PLLs support high resolution (16-bit) fractional-N synthesis. Fractional-N frequency synthesis allows the user to generate an output clock which is a non-integer multiple of the input frequency. For more information about using the PLL with Fractional-N synthesis, please see TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide.

Each output has its own output divider, thus allowing the PLL to generate different frequencies for each output. The output dividers can have a value from 1 to 128. The CLKOS2 and CLKOS3 dividers may also be cascaded together to generate low frequency clocks. The CLKOP, CLKOS, CLKOS2, and CLKOS3 outputs can all be used to drive the MachXO2 clock distribution network directly or general purpose routing resources can be used.

The LOCK signal is asserted when the PLL determines it has achieved lock and de-asserted if a loss of lock is detected. A block diagram of the PLL is shown in Figure 2-7.

The setup and hold times of the device can be improved by programming a phase shift into the CLKOS, CLKOS2, and CLKOS3 output clocks which will advance or delay the output clock with reference to the CLKOP output clock. This phase shift can be either programmed during configuration or can be adjusted dynamically. In dynamic mode, the PLL may lose lock after a phase adjustment on the output used as the feedback source and not relock until the t_{LOCK} parameter has been satisfied.

The MachXO2 also has a feature that allows the user to select between two different reference clock sources dynamically. This feature is implemented using the PLLREFCS primitive. The timing parameters for the PLL are shown in the table.

The MachXO2 PLL contains a WISHBONE port feature that allows the PLL settings, including divider values, to be dynamically changed from the user logic. When using this feature the EFB block must also be instantiated in the design to allow access to the WISHBONE ports. Similar to the dynamic phase adjustment, when PLL settings are updated through the WISHBONE port the PLL may lose lock and not relock until the t_{LOCK} parameter has been satisfied. The timing parameters for the PLL are shown in the table.

For more details on the PLL and the WISHBONE interface, see TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide.

Figure 2-7. PLL Diagram

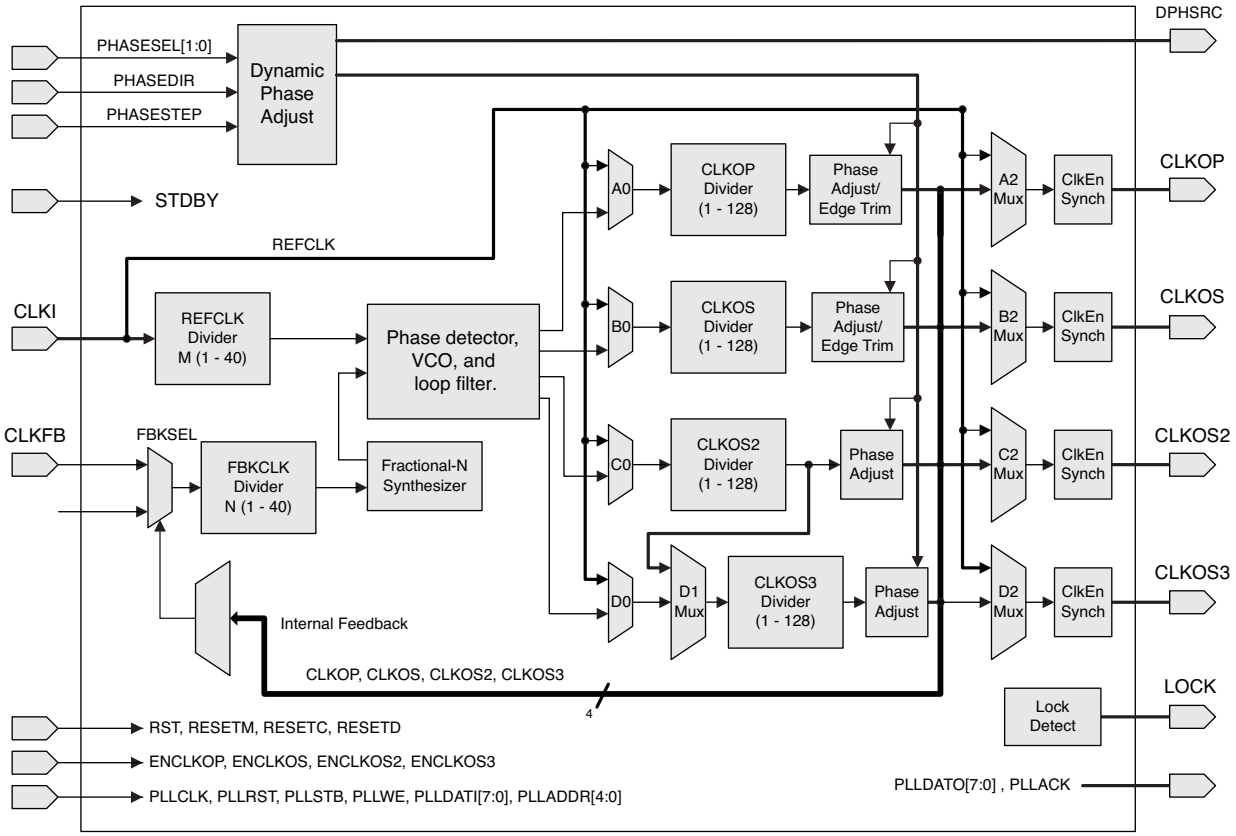


Table 2-4 provides signal descriptions of the PLL block.

Table 2-4. PLL Signal Descriptions

Port Name	I/O	Description
CLKI	I	Input clock to PLL
CLKFB	I	Feedback clock
PHASESEL[1:0]	I	Select which output is affected by Dynamic Phase adjustment ports
PHASEDIR	I	Dynamic Phase adjustment direction
PHASESTEP	I	Dynamic Phase step – toggle shifts VCO phase adjust by one step.
CLKOP	O	Primary PLL output clock (with phase shift adjustment)
CLKOS	O	Secondary PLL output clock (with phase shift adjust)
CLKOS2	O	Secondary PLL output clock2 (with phase shift adjust)
CLKOS3	O	Secondary PLL output clock3 (with phase shift adjust)
LOCK	O	PLL LOCK, asynchronous signal. Active high indicates PLL is locked to input and feedback signals.
DPHSRC	O	Dynamic Phase source – ports or WISHBONE is active
STDBY	I	Standby signal to power down the PLL
RST	I	PLL reset without resetting the M-divider. Active high reset.
RESETM	I	PLL reset - includes resetting the M-divider. Active high reset.
RESETC	I	Reset for CLKOS2 output divider only. Active high reset.
RESETD	I	Reset for CLKOS3 output divider only. Active high reset.
ENCLKOP	I	Enable PLL output CLKOP
ENCLKOS	I	Enable PLL output CLKOS when port is active
ENCLKOS2	I	Enable PLL output CLKOS2 when port is active
ENCLKOS3	I	Enable PLL output CLKOS3 when port is active
PLLCLK	I	PLL data bus clock input signal
PLLRST	I	PLL data bus reset. This resets only the data bus not any register values.
PLLSTB	I	PLL data bus strobe signal
PLLWE	I	PLL data bus write enable signal
PLLADDR [4:0]	I	PLL data bus address
PLLDATI [7:0]	I	PLL data bus data input
PLLDATO [7:0]	O	PLL data bus data output
PLLACK	O	PLL data bus acknowledge signal

sysMEM Embedded Block RAM Memory

The MachXO2-640/U and larger devices contain sysMEM Embedded Block RAMs (EBRs). The EBR consists of a 9-Kbit RAM, with dedicated input and output registers. This memory can be used for a wide variety of purposes including data buffering, PROM for the soft processor and FIFO.

sysMEM Memory Block

The sysMEM block can implement single port, dual port, pseudo dual port, or FIFO memories. Each block can be used in a variety of depths and widths as shown in Table 2-5.

Table 2-5. sysMEM Block Configurations

Memory Mode	Configurations
Single Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9
True Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9
Pseudo Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18
FIFO	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18

Bus Size Matching

All of the multi-port memory modes support different widths on each of the ports. The RAM bits are mapped LSB word 0 to MSB word 0, LSB word 1 to MSB word 1, and so on. Although the word size and number of words for each port varies, this mapping scheme applies to each port.

RAM Initialization and ROM Operation

If desired, the contents of the RAM can be pre-loaded during device configuration. EBR initialization data can be loaded from the UFM. To maximize the number of UFM bits, initialize the EBRs used in your design to an all-zero pattern. Initializing to an all-zero pattern does not use up UFM bits. MachXO2 devices have been designed such that multiple EBRs share the same initialization memory space if they are initialized to the same pattern.

By preloading the RAM block during the chip configuration cycle and disabling the write controls, the sysMEM block can also be utilized as a ROM.

Memory Cascading

Larger and deeper blocks of RAM can be created using EBR sysMEM Blocks. Typically, the Lattice design tools cascade memory transparently, based on specific design inputs.

Single, Dual, Pseudo-Dual Port and FIFO Modes

Figure 2-8 shows the five basic memory configurations and their input/output names. In all the sysMEM RAM modes, the input data and addresses for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the memory array output.

Figure 2-8. sysMEM Memory Primitives

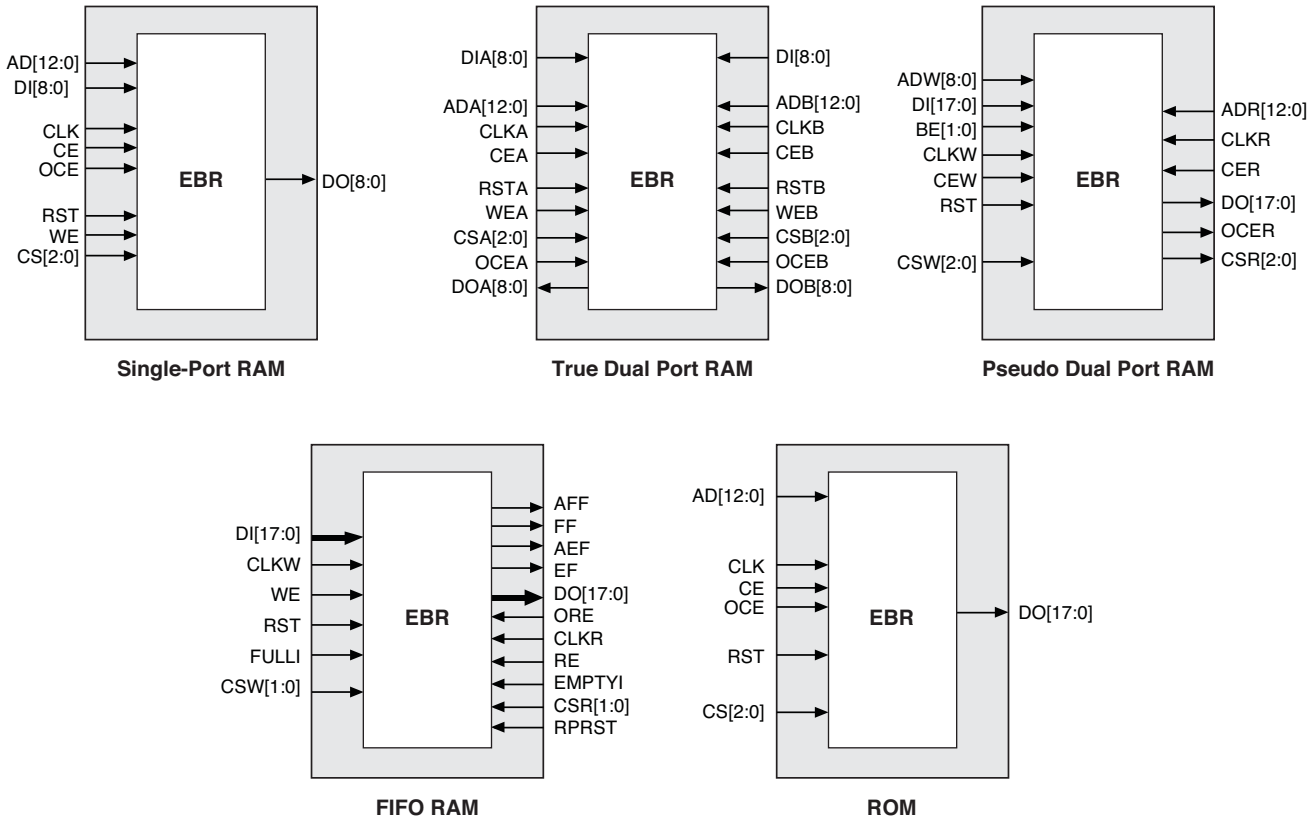


Table 2-6. EBR Signal Descriptions

Port Name	Description	Active State
CLK	Clock	Rising Clock Edge
CE	Clock Enable	Active High
OCE[1]	Output Clock Enable	Active High
RST	Reset	Active High
BE[1]	Byte Enable	Active High
WE	Write Enable	Active High
AD	Address Bus	—
DI	Data In	—
DO	Data Out	—
CS	Chip Select	Active High
AFF	FIFO RAM Almost Full Flag	—
FF	FIFO RAM Full Flag	—
AEF	FIFO RAM Almost Empty Flag	—
EF	FIFO RAM Empty Flag	—
RPRST	FIFO RAM Read Pointer Reset	—

1. Optional signals.
2. For dual port EBR primitives a trailing ‘A’ or ‘B’ in the signal name specifies the EBR port A or port B respectively.
3. For FIFO RAM mode primitive, a trailing ‘R’ or ‘W’ in the signal name specifies the FIFO read port or write port respectively.
4. For FIFO RAM mode primitive FULLI has the same function as CSW(2) and EMPTYI has the same function as CSR(2).
5. In FIFO mode, CLKW is the write port clock, CSW is the write port chip select, CLKR is the read port clock, CSR is the read port chip select, ORE is the output read enable.

The EBR memory supports three forms of write behavior for single or dual port operation:

1. **Normal** – Data on the output appears only during the read cycle. During a write cycle, the data (at the current address) does not appear on the output. This mode is supported for all data widths.

2. **Write Through** – A copy of the input data appears at the output of the same port. This mode is supported for all data widths.

3. **Read-Before-Write** – When new data is being written, the old contents of the address appears at the output.

FIFO Configuration

The FIFO has a write port with data-in, CEW, WE and CLKW signals. There is a separate read port with data-out, RCE, RE and CLKR signals. The FIFO internally generates Almost Full, Full, Almost Empty and Empty Flags. The Full and Almost Full flags are registered with CLKW. The Empty and Almost Empty flags are registered with CLKR. Table 2-7 shows the range of programming values for these flags.

Table 2-7. Programmable FIFO Flag Ranges

Flag Name	Programming Range
Full (FF)	1 to max (up to 2^N-1)
Almost Full (AF)	1 to Full-1
Almost Empty (AE)	1 to Full-1
Empty (EF)	0

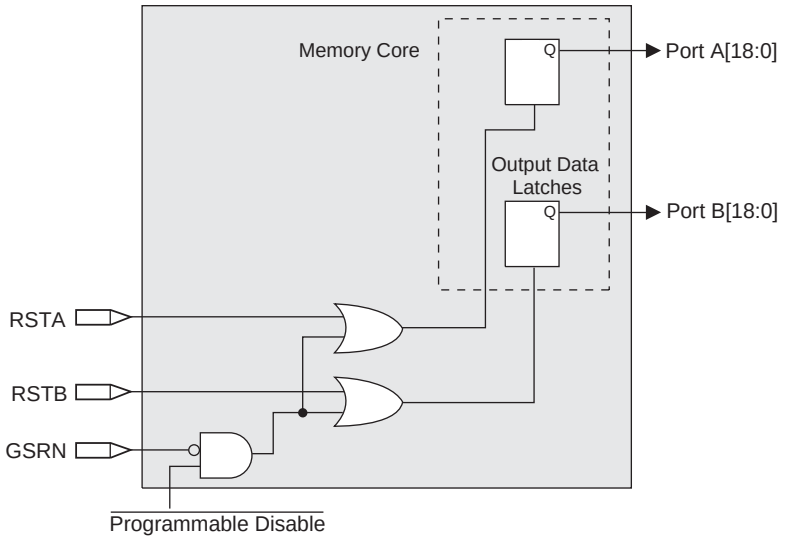
N = Address bit width.

The FIFO state machine supports two types of reset signals: RST and RPRST. The RST signal is a global reset that clears the contents of the FIFO by resetting the read/write pointer and puts the FIFO flags in their initial reset state. The RPRST signal is used to reset the read pointer. The purpose of this reset is to retransmit the data that is in the FIFO. In these applications it is important to keep careful track of when a packet is written into or read from the FIFO.

Memory Core Reset

The memory core contains data output latches for ports A and B. These are simple latches that can be reset synchronously or asynchronously. RSTA and RSTB are local signals, which reset the output latches associated with port A and port B respectively. The Global Reset (GSRN) signal resets both ports. The output data latches and associated resets for both ports are as shown in Figure 2-9.

Figure 2-9. Memory Core Reset

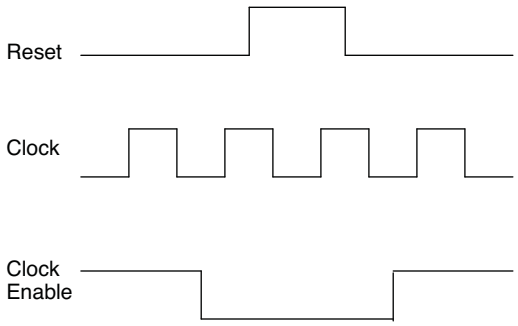


For further information on the sysMEM EBR block, please refer to TN1201, Memory Usage Guide for MachXO2 Devices.

EBR Asynchronous Reset

EBR asynchronous reset or GSR (if used) can only be applied if all clock enables are low for a clock cycle before the reset is applied and released a clock cycle after the reset is released, as shown in Figure 2-10. The GSR input to the EBR is always asynchronous.

Figure 2-10. EBR Asynchronous Reset (Including GSR) Timing Diagram



If all clock enables remain enabled, the EBR asynchronous reset or GSR may only be applied and released after the EBR read and write clock inputs are in a steady state condition for a minimum of $1/f_{MAX}$ (EBR clock). The reset release must adhere to the EBR synchronous reset setup time before the next active read or write clock edge.

If an EBR is pre-loaded during configuration, the GSR input must be disabled or the release of the GSR during device wake up must occur before the release of the device I/Os becoming active.

These instructions apply to all EBR RAM, ROM and FIFO implementations. For the EBR FIFO mode, the GSR signal is always enabled and the WE and RE signals act like the clock enable signals in Figure 2-10. The reset timing rules apply to the RPReset input versus the RE input and the RST input versus the WE and RE inputs. Both RST and RPReset are always asynchronous EBR inputs. For more details refer to TN1201, Memory Usage Guide for MachXO2 Devices.

Note that there are no reset restrictions if the EBR synchronous reset is used and the EBR GSR input is disabled.

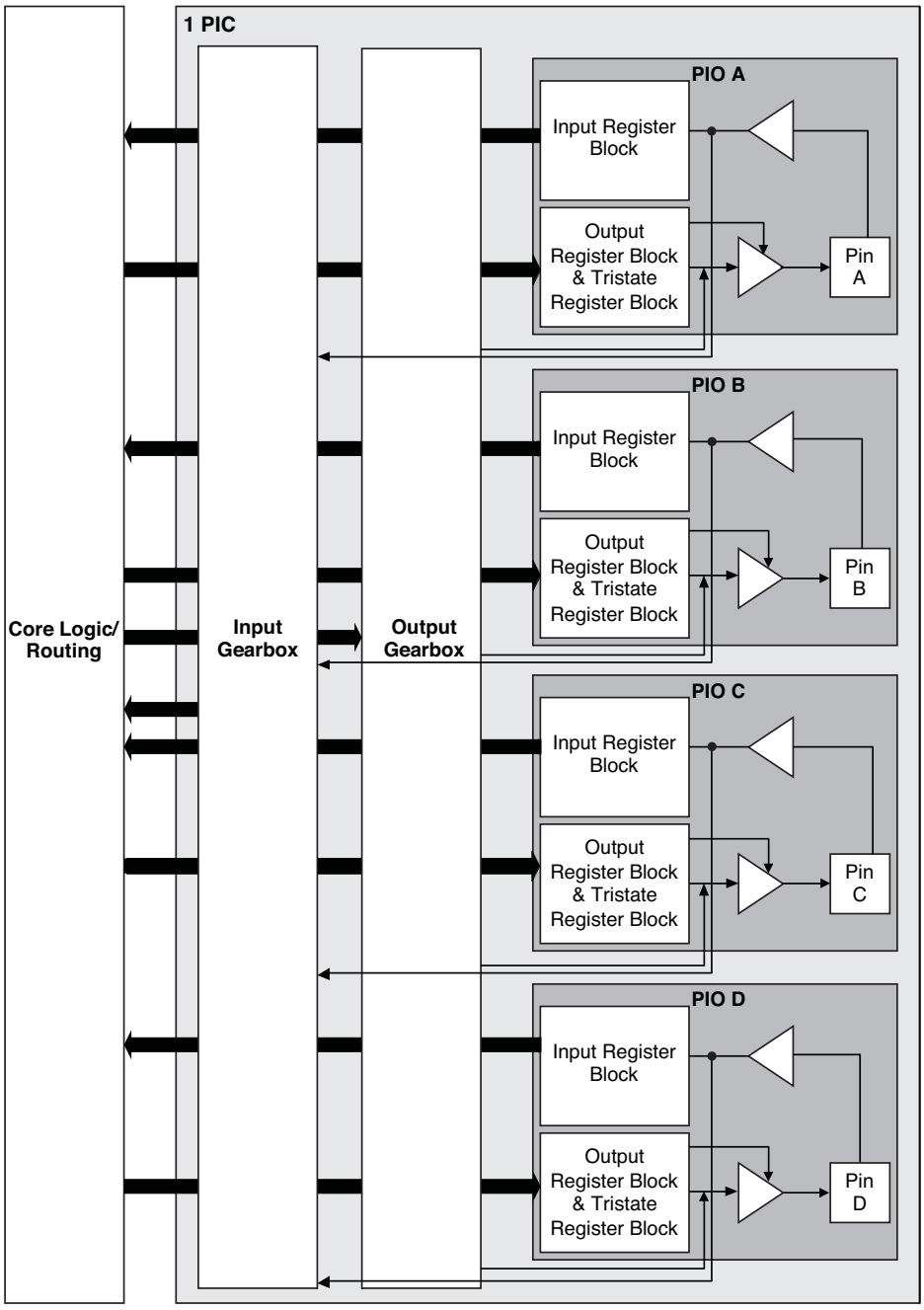
Programmable I/O Cells (PIC)

The programmable logic associated with an I/O is called a PIO. The individual PIO are connected to their respective sysIO buffers and pads. On the MachXO2 devices, the PIO cells are assembled into groups of four PIO cells called a Programmable I/O Cell or PIC. The PICs are placed on all four sides of the device.

On all the MachXO2 devices, two adjacent PIOs can be combined to provide a complementary output driver pair.

The MachXO2-640U, MachXO2-1200/U and higher density devices contain enhanced I/O capability. All PIO pairs on these larger devices can implement differential receivers. Half of the PIO pairs on the top edge of these devices can be configured as true LVDS transmit pairs. The PIO pairs on the bottom edge of these higher density devices have on-chip differential termination and also provide PCI support.

Figure 2-11. Group of Four Programmable I/O Cells



Notes:
1. Input gearbox is available only in PIC on the bottom edge of MachXO2-640U, MachXO2-1200/U and larger devices.
2. Output gearbox is available only in PIC on the top edge of MachXO2-640U, MachXO2-1200/U and larger devices.

PIO

The PIO contains three blocks: an input register block, output register block and tri-state register block. These blocks contain registers for operating in a variety of modes along with the necessary clock and selection logic.

Table 2-8. PIO Signal List

Pin Name	I/O Type	Description
CE	Input	Clock Enable
D	Input	Pin input from sysIO buffer.
INDD	Output	Register bypassed input.
INCK	Output	Clock input
Q0	Output	DDR positive edge input
Q1	Output	Registered input/DDR negative edge input
D0	Input	Output signal from the core (SDR and DDR)
D1	Input	Output signal from the core (DDR)
TD	Input	Tri-state signal from the core
Q	Output	Data output signals to sysIO Buffer
TQ	Output	Tri-state output signals to sysIO Buffer
DQSR90[1]	Input	DQS shift 90-degree read clock
DQSW90[1]	Input	DQS shift 90-degree write clock
DDRCLKPOL[1]	Input	DDR input register polarity control signal from DQS
SCLK	Input	System clock for input and output/tri-state blocks.
RST	Input	Local set reset signal

1. Available in PIO on right edge only.

Input Register Block

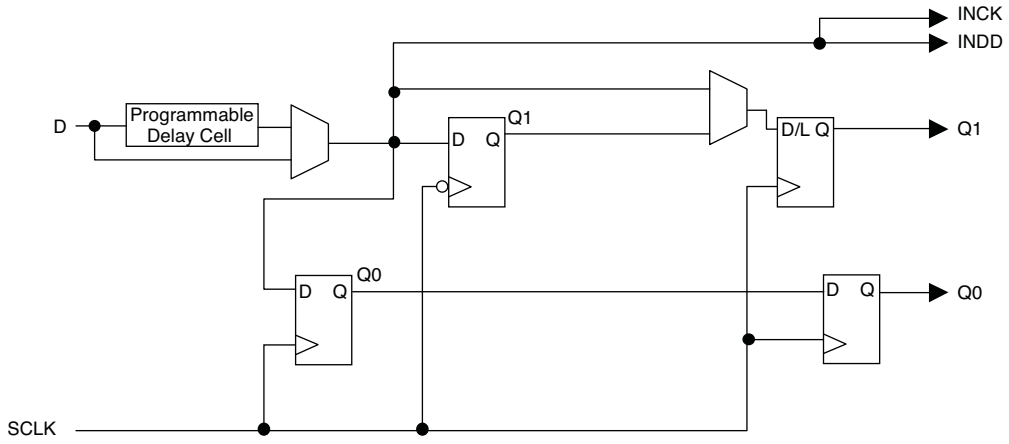
The input register blocks for the PIOs on all edges contain delay elements and registers that can be used to condition high-speed interface signals before they are passed to the device core. In addition to this functionality, the input register blocks for the PIOs on the right edge include built-in logic to interface to DDR memory.

Figure 2-12 shows the input register block for the PIOs located on the left, top and bottom edges. Figure 2-13 shows the input register block for the PIOs on the right edge.

Left, Top, Bottom Edges

Input signals are fed from the sysIO buffer to the input register block (as signal D). If desired, the input signal can bypass the register and delay elements and be used directly as a combinatorial signal (INDD), and a clock (INCK). If an input delay is desired, users can select a fixed delay. I/Os on the bottom edge also have a dynamic delay, DEL[4:0]. The delay, if selected, reduces input register hold time requirements when using a global clock. The input block allows two modes of operation. In single data rate (SDR) the data is registered with the system clock (SCLK) by one of the registers in the single data rate sync register block. In Generic DDR mode, two registers are used to sample the data on the positive and negative edges of the system clock (SCLK) signal, creating two data streams.

Figure 2-12. MachXO2 Input Register Block Diagram (PIO on Left, Top and Bottom Edges)



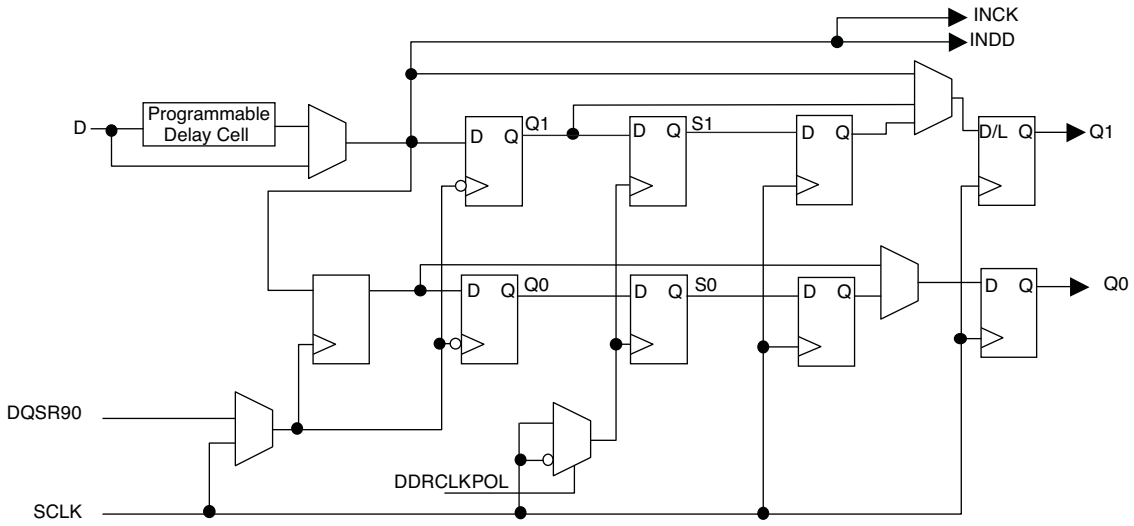
Right Edge

The input register block on the right edge is a superset of the same block on the top, bottom, and left edges. In addition to the modes described above, the input register block on the right edge also supports DDR memory mode.

In DDR memory mode, two registers are used to sample the data on the positive and negative edges of the modified DQS (DQSR90) in the DDR Memory mode creating two data streams. Before entering the core, these two data streams are synchronized to the system clock to generate two data streams.

The signal DDRCLKPOL controls the polarity of the clock used in the synchronization registers. It ensures adequate timing when data is transferred to the system clock domain from the DQS domain. The DQSR90 and DDRCLKPOL signals are generated in the DQS read-write block.

Figure 2-13. MachXO2 Input Register Block Diagram (PIO on Right Edge)



Output Register Block

The output register block registers signals from the core of the device before they are passed to the sysIO buffers.

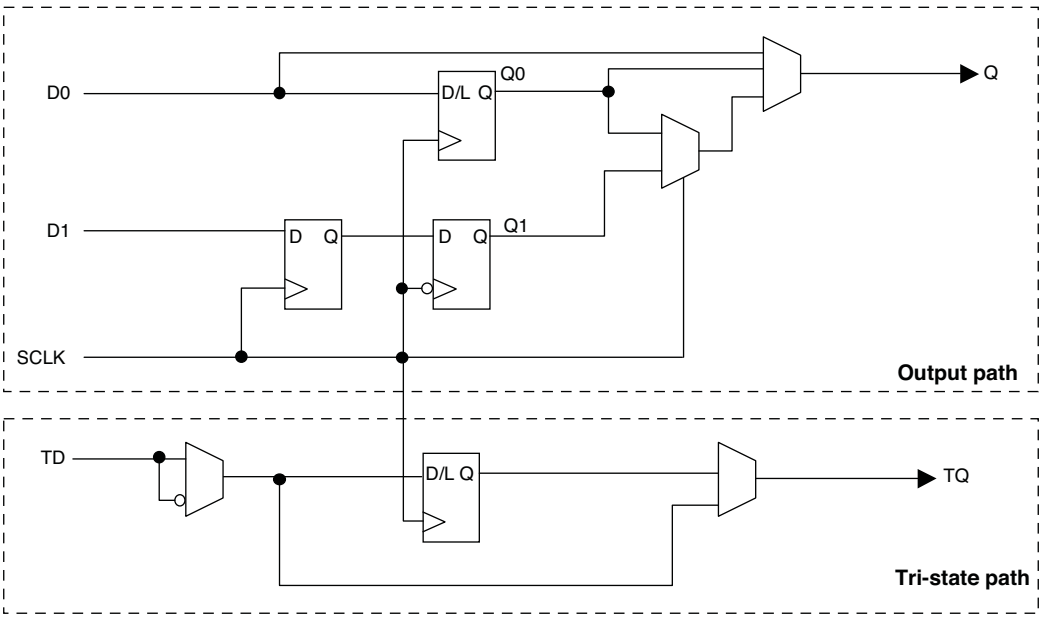
Left, Top, Bottom Edges

In SDR mode, D0 feeds one of the flip-flops that then feeds the output. The flip-flop can be configured as a D-type register or latch.

In DDR generic mode, D0 and D1 inputs are fed into registers on the positive edge of the clock. At the next falling edge the registered D1 input is registered into the register Q1. A multiplexer running off the same clock is used to switch the mux between the outputs of registers Q0 and Q1 that will then feed the output.

Figure 2-14 shows the output register block on the left, top and bottom edges.

Figure 2-14. MachXO2 Output Register Block Diagram (PIO on the Left, Top and Bottom Edges)



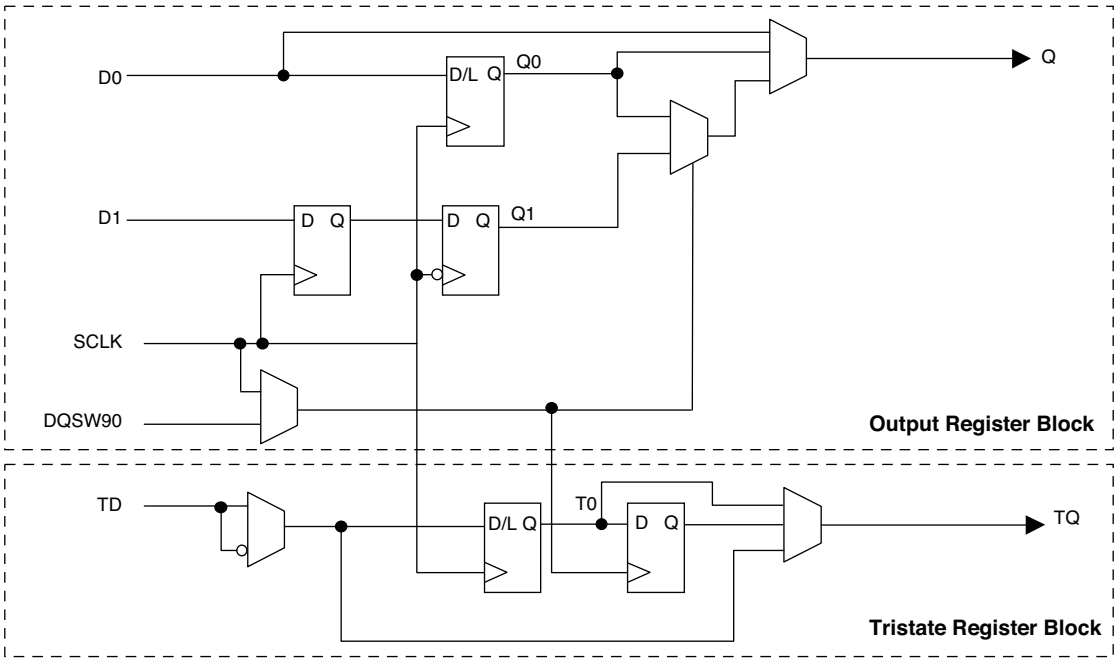
Right Edge

The output register block on the right edge is a superset of the output register on left, top and bottom edges of the device. In addition to supporting SDR and Generic DDR modes, the output register blocks for PIOs on the right edge include additional logic to support DDR-memory interfaces. Operation of this block is similar to that of the output register block on other edges.

In DDR memory mode, D0 and D1 inputs are fed into registers on the positive edge of the clock. At the next falling edge the registered D1 input is registered into the register Q1. A multiplexer running off the DQSW90 signal is used to switch the mux between the outputs of registers Q0 and Q1 that will then feed the output.

Figure 2-15 shows the output register block on the right edge.

Figure 2-15. MachXO2 Output Register Block Diagram (PIO on the Right Edges)



Tri-state Register Block

The tri-state register block registers tri-state control signals from the core of the device before they are passed to the sysIO buffers. The block contains a register for SDR operation. In SDR, TD input feeds one of the flip-flops that then feeds the output.

The tri-state register blocks on the right edge contain an additional register for DDR memory operation. In DDR memory mode, the register TS input is fed into another register that is clocked using the DQSW90 signal. The output of this register is used as a tri-state control.

Input Gearbox

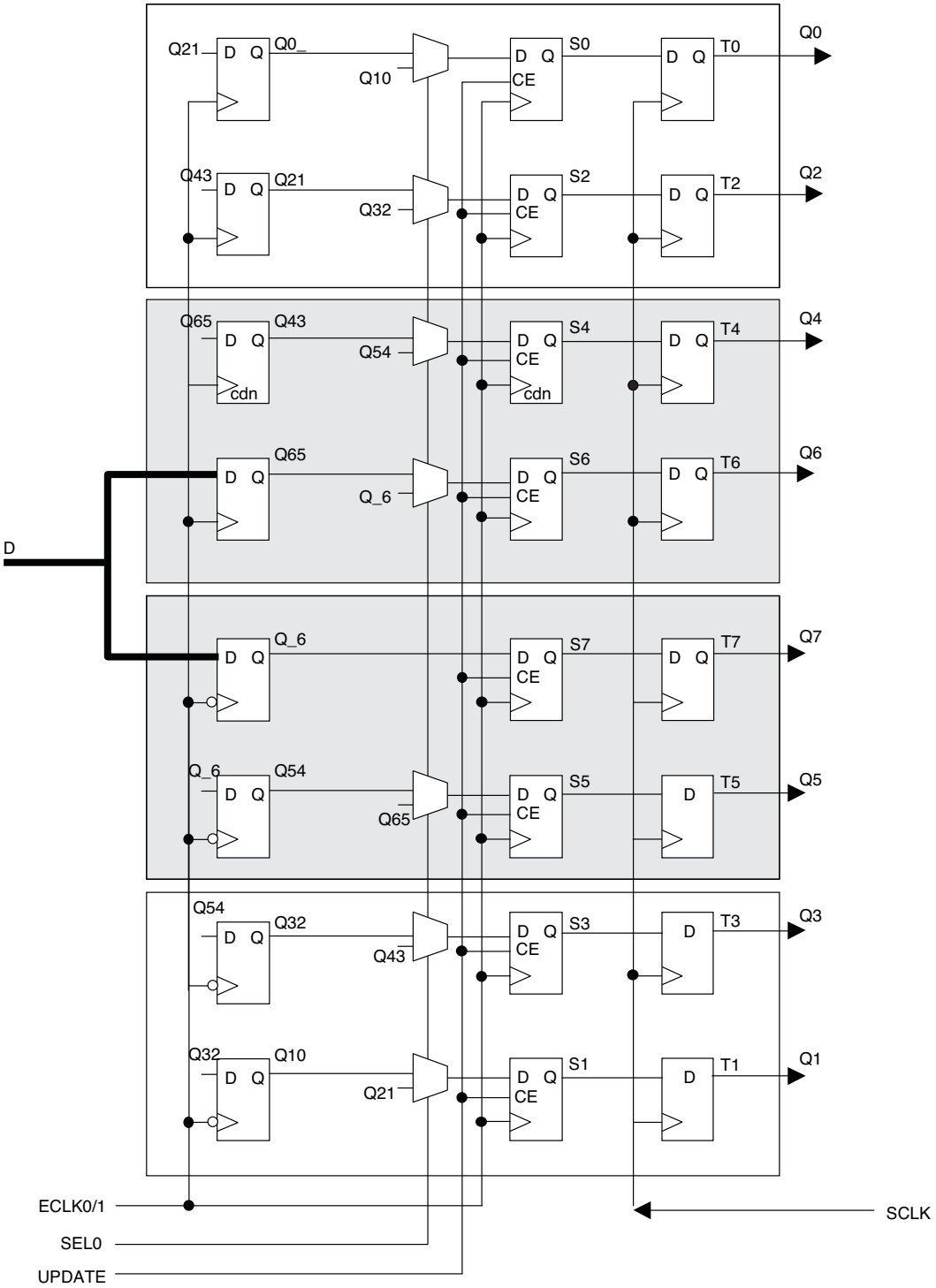
Each PIC on the bottom edge has a built-in 1:8 input gearbox. Each of these input gearboxes may be programmed as a 1:7 de-serializer or as one IDDRX4 (1:8) gearbox or as two IDDRX2 (1:4) gearboxes. Table 2-9 shows the gearbox signals.

Table 2-9. Input Gearbox Signal List

Name	I/O Type	Description
D	Input	High-speed data input after programmable delay in PIO A input register block
ALIGNWD	Input	Data alignment signal from device core
SCLK	Input	Slow-speed system clock
ECLK[1:0]	Input	High-speed edge clock
RST	Input	Reset
Q[7:0]	Output	Low-speed data to device core: Video RX(1:7): Q[6:0] GDDRX4(1:8): Q[7:0] GDDRX2(1:4)(IOL-A): Q4, Q5, Q6, Q7 GDDRX2(1:4)(IOL-C): Q0, Q1, Q2, Q3

These gearboxes have three stage pipeline registers. The first stage registers sample the high-speed input data by the high-speed edge clock on its rising and falling edges. The second stage registers perform data alignment based on the control signals UPDATE and SEL0 from the control block. The third stage pipeline registers pass the data to the device core synchronized to the low-speed system clock. Figure 2-16 shows a block diagram of the input gearbox.

Figure 2-16. Input Gearbox



More information on the input gearbox is available in TN1203, Implementing High-Speed Interfaces with MachXO2 Devices.

Output Gearbox

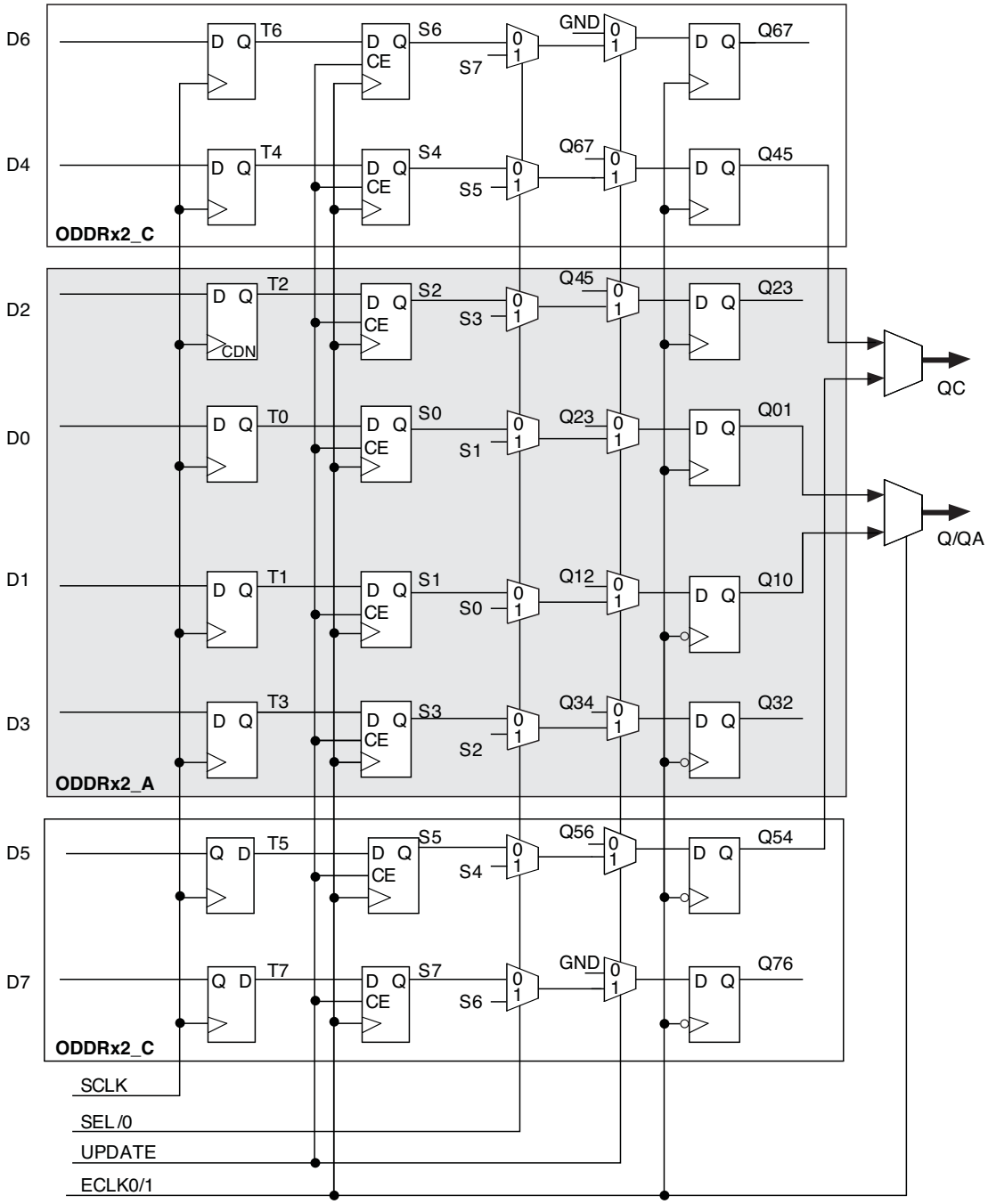
Each PIC on the top edge has a built-in 8:1 output gearbox. Each of these output gearboxes may be programmed as a 7:1 serializer or as one ODDRX4 (8:1) gearbox or as two ODDRX2 (4:1) gearboxes. Table 2-10 shows the gearbox signals.

Table 2-10. Output Gearbox Signal List

Name	I/O Type	Description
Q	Output	High-speed data output
D[7:0]	Input	Low-speed data from device core
Video TX(7:1): D[6:0]		
GDDRX4(8:1): D[7:0]		
GDDRX2(4:1)(IOL-A): D[3:0]		
GDDRX2(4:1)(IOL-C): D[7:4]		
SCLK	Input	Slow-speed system clock
ECLK [1:0]	Input	High-speed edge clock
RST	Input	Reset

The gearboxes have three stage pipeline registers. The first stage registers sample the low-speed input data on the low-speed system clock. The second stage registers transfer data from the low-speed clock registers to the high-speed clock registers. The third stage pipeline registers controlled by high-speed edge clock shift and mux the high-speed data out to the sysIO buffer. Figure 2-17 shows the output gearbox block diagram.

Figure 2-17. Output Gearbox



More information on the output gearbox is available in TN1203, Implementing High-Speed Interfaces with MachXO2 Devices.

DDR Memory Support

Certain PICs on the right edge of MachXO2-640U, MachXO2-1200/U and larger devices, have additional circuitry to allow the implementation of DDR memory interfaces. There are two groups of 14 or 12 PIOs each on the right edge with additional circuitry to implement DDR memory interfaces. This capability allows the implementation of up to 16-bit wide memory interfaces. One PIO from each group contains a control element, the DQS Read/Write

Block, to facilitate the generation of clock and control signals (DQSR90, DQSW90, DDRCLKPOL and DATAVALID). These clock and control signals are distributed to the other PIO in the group through dedicated low skew routing.

DQS Read Write Block

Source synchronous interfaces generally require the input clock to be adjusted in order to correctly capture data at the input register. For most interfaces a PLL is used for this adjustment. However, in DDR memories the clock (referred to as DQS) is not free-running so this approach cannot be used. The DQS Read Write block provides the required clock alignment for DDR memory interfaces. DQSR90 and DQSW90 signals are generated by the DQS Read Write block from the DQS input.

In a typical DDR memory interface design, the phase relationship between the incoming delayed DQS strobe and the internal system clock (during the read cycle) is unknown. The MachXO2 family contains dedicated circuits to transfer data between these domains. To prevent set-up and hold violations, at the domain transfer between DQS (delayed) and the system clock, a clock polarity selector is used. This circuit changes the edge on which the data is registered in the synchronizing registers in the input register block. This requires evaluation at the start of each read cycle for the correct clock polarity. Prior to the read operation in DDR memories, DQS is in tri-state (pulled by termination). The DDR memory device drives DQS low at the start of the preamble state. A dedicated circuit in the DQS Read Write block detects the first DQS rising edge after the preamble state and generates the DDRCLKPOL signal. This signal is used to control the polarity of the clock to the synchronizing registers.

The temperature, voltage and process variations of the DQS delay block are compensated by a set of calibration signals (6-bit bus) from a DLL on the right edge of the device. The DLL loop is compensated for temperature, voltage and process variations by the system clock and feedback loop.

sysIO Buffer

Each I/O is associated with a flexible buffer referred to as a sysIO buffer. These buffers are arranged around the periphery of the device in groups referred to as banks. The sysIO buffers allow users to implement a wide variety of standards that are found in today's systems including LVCMOS, TTL, PCI, SSTL, HSTL, LVDS, BLVDS, MLVDS and LVPECL.

Each bank is capable of supporting multiple I/O standards. In the MachXO2 devices, single-ended output buffers, ratioed input buffers (LVTTL, LVCMOS and PCI), differential (LVDS) and referenced input buffers (SSTL and HSTL) are powered using I/O supply voltage (V_{CCIO}). Each sysIO bank has its own V_{CCIO}. In addition, each bank has a voltage reference, V_{REF}, which allows the use of referenced input buffers independent of the bank V_{CCIO}.

MachXO2-256 and MachXO2-640 devices contain single-ended ratioed input buffers and single-ended output buffers with complementary outputs on all the I/O banks. Note that the single-ended input buffers on these devices do not contain PCI clamps. In addition to the single-ended I/O buffers these two devices also have differential and referenced input buffers on all I/Os. The I/Os are arranged in pairs, the two pads in the pair are described as "T" and "C", where the true pad is associated with the positive side of the differential input buffer and the comp (complementary) pad is associated with the negative side of the differential input buffer.

MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000 and MachXO2-7000 devices contain three types of sysIO buffer pairs.

1. **Left and Right sysIO Buffer Pairs**
   The sysIO buffer pairs in the left and right banks of the device consist of two single-ended output drivers and two single-ended input buffers (for ratioed inputs such as LVCMOS and LVTTL). The I/O pairs on the left and right of the devices also have differential and referenced input buffers.

2. **Bottom sysIO Buffer Pairs**
   The sysIO buffer pairs in the bottom bank of the device consist of two single-ended output drivers and two single-ended input buffers (for ratioed inputs such as LVCMOS and LVTTL). The I/O pairs on the bottom also have differential and referenced input buffers. Only the I/Os on the bottom banks have programmable PCI clamps

and differential input termination. The PCI clamp is enabled after V_{CC} and V_{CCIO} are at valid operating levels and the device has been configured.

3. **Top sysIO Buffer Pairs**
   The sysIO buffer pairs in the top bank of the device consist of two single-ended output drivers and two single-ended input buffers (for ratioed inputs such as LVCMOS and LVTTL). The I/O pairs on the top also have differential and referenced I/O buffers. Half of the sysIO buffer pairs on the top edge have true differential outputs. The sysIO buffer pair comprising of the A and B PIOs in every PIC on the top edge have a differential output driver. The referenced input buffer can also be configured as a differential input buffer.

Typical I/O Behavior During Power-up

The internal power-on-reset (POR) signal is deactivated when V_{CC} and V_{CCIO0} have reached V_{PORUP} level defined in the Power-On-Reset Voltage table in the DC and Switching Characteristics section of this data sheet. After the POR signal is deactivated, the FPGA core logic becomes active. It is the user's responsibility to ensure that all V_{CCIO} banks are active with valid input logic levels to properly control the output logic states of all the I/O banks that are critical to the application. The default configuration of the I/O pins in a blank device is tri-state with a weak pull-down to GND (some pins such as PROGRAMN and the JTAG pins have weak pull-up to V_{CCIO} as the default functionality). The I/O pins will maintain the blank configuration until V_{CC} and V_{CCIO} (for I/O banks containing configuration I/Os) have reached V_{PORUP} levels at which time the I/Os will take on the user-configured settings only after a proper download/configuration.

There are various ways a user can ensure that there are no spurious signals on critical outputs as the device powers up. These are discussed in more detail in TN1202, MachXO2 sysIO Usage Guide.

Supported Standards

The MachXO2 sysIO buffer supports both single-ended and differential standards. Single-ended standards can be further subdivided into LVCMOS, LVTTL, and PCI. The buffer supports the LVTTL, PCI, LVCMOS 1.2, 1.5, 1.8, 2.5, and 3.3V standards. In the LVCMOS and LVTTL modes, the buffer has individually configurable options for drive strength, bus maintenance (weak pull-up, weak pull-down, bus-keeper latch or none) and open drain. BLVDS, MLVDS and LVPECL output emulation is supported on all devices. The MachXO2-640U, MachXO2-1200/U and higher devices support on-chip LVDS output buffers on approximately 50% of the I/Os on the top bank. Differential receivers for LVDS, BLVDS, MLVDS and LVPECL are supported on all banks of MachXO2 devices. PCI support is provided in the bottom bank of theMachXO2-640U, MachXO2-1200/U and higher density devices. Table 2-11 summarizes the I/O characteristics of the MachXO2 PLDs.

Tables 2-11 and 2-12 show the I/O standards (together with their supply and reference voltages) supported by the MachXO2 devices. For further information on utilizing the sysIO buffer to support a variety of standards please see TN1202, MachXO2 sysIO Usage Guide.

Table 2-11. I/O Support Device by Device

	MachXO2-256, MachXO2-640	**MachXO2-640U, MachXO2-1200**	**MachXO2-1200U MachXO2-2000/U, MachXO2-4000, MachXO2-7000**
Number of I/O Banks	4	4	6
Type of Input Buffers	Single-ended (all I/O banks) Differential Receivers (all I/O banks)	Single-ended (all I/O banks) Differential Receivers (all I/O banks) Differential input termination (bottom side)	Single-ended (all I/O banks) Differential Receivers (all I/O banks) Differential input termination (bottom side)

	MachXO2-256, MachXO2-640	MachXO2-640U, MachXO2-1200	MachXO2-1200U MachXO2-2000/U, MachXO2-4000, MachXO2-7000
Types of Output Buffers	Single-ended buffers with complementary outputs (all I/O banks)	Single-ended buffers with complementary outputs (all I/O banks) Differential buffers with true LVDS outputs (50% on top side)	Single-ended buffers with complementary outputs (all I/O banks) Differential buffers with true LVDS outputs (50% on top side)
Differential Output Emulation Capability	All I/O banks	All I/O banks	All I/O banks
PCI Clamp Support	No	Clamp on bottom side only	Clamp on bottom side only

Table 2-12. Supported Input Standards

	VCCIO (Typ.)				
Input Standard	**3.3V**	**2.5V**	**1.8V**	**1.5**	**1.2V**
Single-Ended Interfaces					
LVTTL	✓	✓[2]	✓[2]	✓[2]	
LVCMOS33	✓	✓[2]	✓[2]	✓[2]	
LVCMOS25	✓[2]	✓	✓[2]	✓[2]	
LVCMOS18	✓[2]	✓[2]	✓	✓[2]	
LVCMOS15	✓[2]	✓[2]	✓[2]	✓	✓[2]
LVCMOS12	✓[2]	✓[2]	✓[2]	✓[2]	✓
PCI[1]	✓				
SSTL18 (Class I, Class II)			✓		
SSTL25 (Class I, Class II)		✓			
HSTL18 (Class I, Class II)			✓		
Differential Interfaces					
LVDS	✓	✓			
BLVDS, MVDS, LVPECL, RSDS	✓	✓			
Differential SSTL18 Class I, II			✓		
Differential SSTL25 Class I, II		✓			
Differential HSTL18 Class I, II			✓		

1. Bottom banks of MachXO2-640U, MachXO2-1200/U and higher density devices only.
2. Reduced functionality. Refer to TN1202, MachXO2 sysIO Usage Guide for more detail.

Table 2-13. Supported Output Standards

Output Standard	V_{CCIO} (Typ.)
Single-Ended Interfaces	
LVTTL	3.3
LVCMOS33	3.3
LVCMOS25	2.5
LVCMOS18	1.8
LVCMOS15	1.5
LVCMOS12	1.2
LVCMOS33, Open Drain	—
LVCMOS25, Open Drain	—
LVCMOS18, Open Drain	—
LVCMOS15, Open Drain	—
LVCMOS12, Open Drain	—
PCI33	3.3
SSTL25 (Class I)	2.5
SSTL18 (Class I)	1.8
HSTL18(Class I)	1.8
Differential Interfaces	
LVDS[1, 2]	2.5, 3.3
BLVDS, MLVDS, RSDS [2]	2.5
LVPECL[2]	3.3
Differential SSTL18	1.8
Differential SSTL25	2.5
Differential HSTL18	1.8

1. MachXO2-640U, MachXO2-1200/U and larger devices have dedicated LVDS buffers.
2. These interfaces can be emulated with external resistors in all devices.

sysIO Buffer Banks

The numbers of banks vary between the devices of this family. MachXO2-1200U, MachXO2-2000/U and higher density devices have six I/O banks (one bank on the top, right and bottom side and three banks on the left side). The MachXO2-1200 and lower density devices have four banks (one bank per side). Figures 2-18 and 2-19 show the sysIO banks and their associated supplies for all devices.

Figure 2-18. MachXO2-1200U, MachXO2-2000/U, MachXO2-4000 and MachXO2-7000 Banks

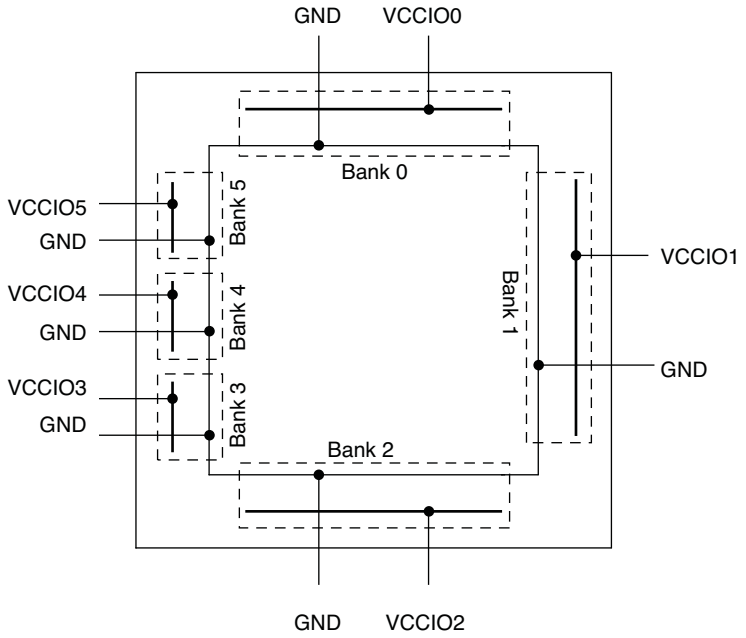
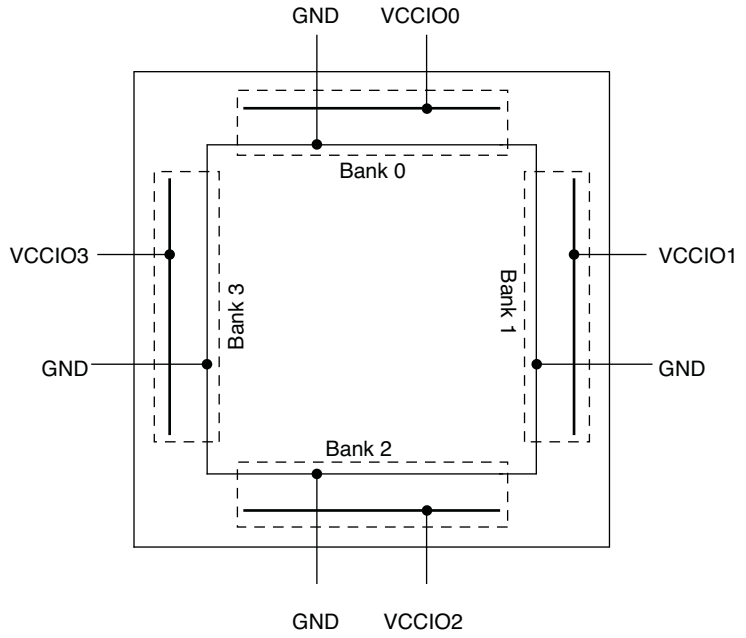


Figure 2-19. MachXO2-256, MachXO2-640/U and MachXO2-1200 Banks



Hot Socketing

The MachXO2 devices have been carefully designed to ensure predictable behavior during power-up and power-down. Leakage into I/O pins is controlled to within specified limits. This allows for easy integration with the rest of the system. These capabilities make the MachXO2 ideal for many multiple power supply and hot-swap applications.

On-chip Oscillator

Every MachXO2 device has an internal CMOS oscillator. The oscillator output can be routed as a clock to the clock tree or as a reference clock to the sysCLOCK PLL using general routing resources. The oscillator frequency can be divided by internal logic. There is a dedicated programming bit and a user input to enable/disable the oscillator. The oscillator frequency ranges from 2.08 MHz to 133 MHz. The software default value of the Master Clock (MCLK) is nominally 2.08 MHz. When a different MCLK is selected during the design process, the following sequence takes place:

1. Device powers up with a nominal MCLK frequency of 2.08 MHz.
2. During configuration, users select a different master clock frequency.
3. The MCLK frequency changes to the selected frequency once the clock configuration bits are received.
4. If the user does not select a master clock frequency, then the configuration bitstream defaults to the MCLK frequency of 2.08 MHz.

Table 2-14 lists all the available MCLK frequencies.

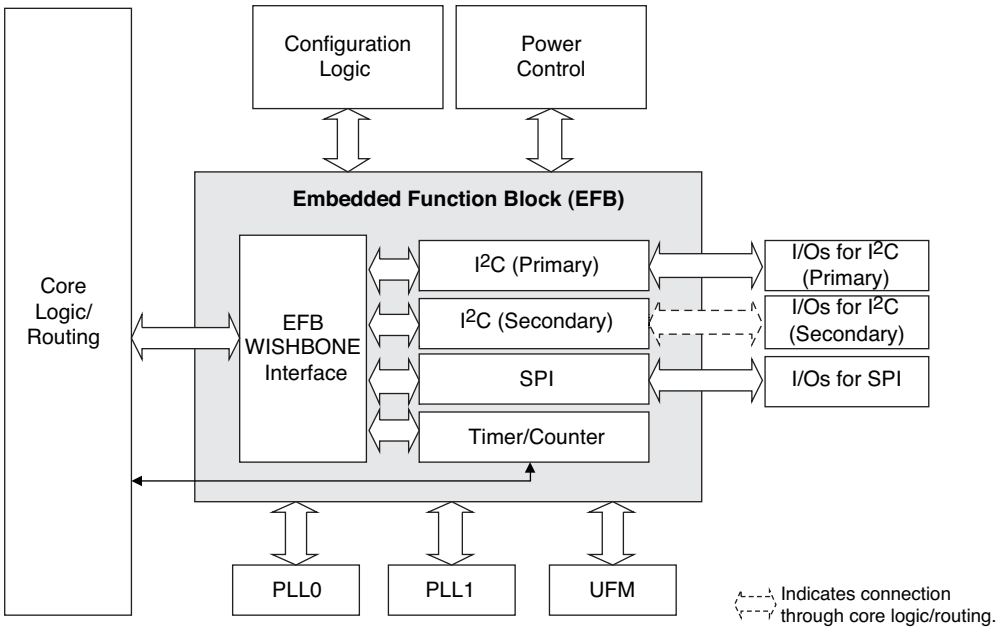
Table 2-14. Available MCLK Frequencies

MCLK (MHz, Nominal)	MCLK (MHz, Nominal)	MCLK (MHz, Nominal)
2.08 (default)	9.17	33.25
2.46	10.23	38
3.17	13.3	44.33
4.29	14.78	53.2
5.54	20.46	66.5
7	26.6	88.67
8.31	29.56	133

Embedded Hardened IP Functions and User Flash Memory

All MachXO2 devices provide embedded hardened functions such as SPI, I^2C and Timer/Counter. MachXO2-640/U and higher density devices also provide User Flash Memory (UFM). These embedded blocks interface through the WISHBONE interface with routing as shown in Figure 2-20.

Figure 2-20. Embedded Function Block Interface



Hardened I²C IP Core

Every MachXO2 device contains two I²C IP cores. These are the primary and secondary I²C IP cores. Either of the two cores can be configured either as an I²C master or as an I²C slave. The only difference between the two IP cores is that the primary core has pre-assigned I/O pins whereas users can assign I/O pins for the secondary core.

When the IP core is configured as a master it will be able to control other devices on the I²C bus through the interface. When the core is configured as the slave, the device will be able to provide I/O expansion to an I²C Master. The I²C cores support the following functionality:

- Master and Slave operation
- 7-bit and 10-bit addressing
- Multi-master arbitration support
- Clock stretching
- Up to 400 KHz data transfer speed
- General call support
- Interface to custom logic through 8-bit WISHBONE interface

Figure 2-21. I²C Core Block Diagram

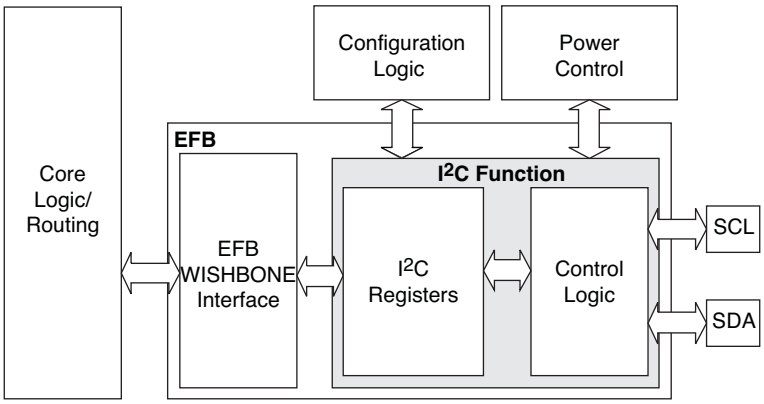


Table 2-15 describes the signals interfacing with the I²C cores.

Table 2-15. I²C Core Signal Description

Signal Name	I/O	Description
i2c_scl	Bi-directional	Bi-directional clock line of the I²C core. The signal is an output if the I²C core is in master mode. The signal is an input if the I²C core is in slave mode. MUST be routed directly to the pre-assigned I/O of the chip. Refer to the Pinout Information section of this document for detailed pad and pin locations of I²C ports in each MachXO2 device.
i2c_sda	Bi-directional	Bi-directional data line of the I²C core. The signal is an output when data is transmitted from the I²C core. The signal is an input when data is received into the I²C core. MUST be routed directly to the pre-assigned I/O of the chip. Refer to the Pinout Information section of this document for detailed pad and pin locations of I²C ports in each MachXO2 device.
i2c_irqo	Output	Interrupt request output signal of the I²C core. The intended usage of this signal is for it to be connected to the WISHBONE master controller (i.e. a microcontroller or state machine) and request an interrupt when a specific condition is met. These conditions are described with the I²C register definitions.
cfg_wake	Output	Wake-up signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, I²C Tab.
cfg_stdby	Output	Stand-by signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, I²C Tab.

Hardened SPI IP Core

Every MachXO2 device has a hard SPI IP core that can be configured as a SPI master or slave. When the IP core is configured as a master it will be able to control other SPI enabled chips connected to the SPI bus. When the core is configured as the slave, the device will be able to interface to an external SPI master. The SPI IP core on MachXO2 devices supports the following functions:

- Configurable Master and Slave modes
- Full-Duplex data transfer
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase
- LSB First or MSB First Data Transfer
- Interface to custom logic through 8-bit WISHBONE interface

There are some limitations on the use of the hardened user SPI. These are defined in the following technical notes:

- TN1087, Minimizing System Interruption During Configuration Using TransFR Technology (Appendix B)
- TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices

Figure 2-22. SPI Core Block Diagram

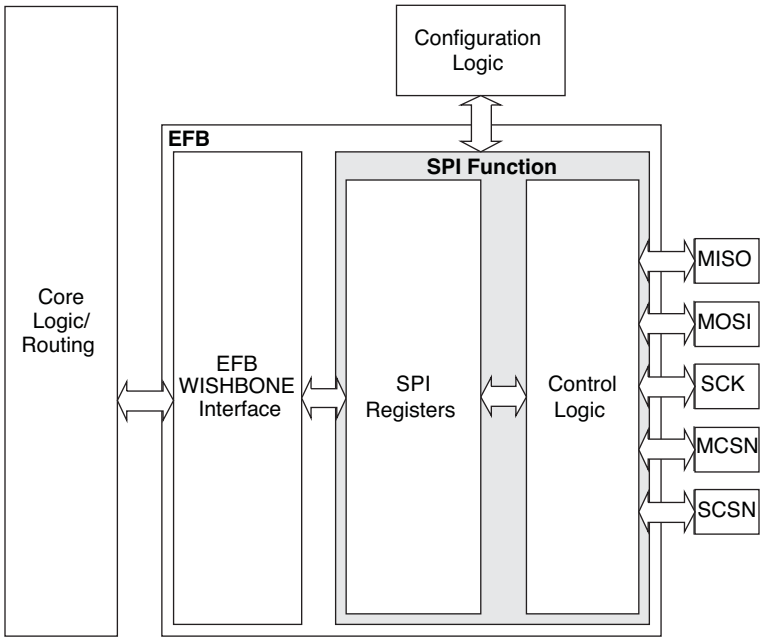


Table 2-16 describes the signals interfacing with the SPI cores.

Table 2-16. SPI Core Signal Description

Signal Name	I/O	Master/Slave	Description
spi_csn[0]	O	Master	SPI master chip-select output
spi_csn[1..7]	O	Master	Additional SPI chip-select outputs (total up to eight slaves)
spi_scsn	I	Slave	SPI slave chip-select input
spi_irq	O	Master/Slave	Interrupt request
spi_clk	I/O	Master/Slave	SPI clock. Output in master mode. Input in slave mode.
spi_miso	I/O	Master/Slave	SPI data. Input in master mode. Output in slave mode.
spi_mosi	I/O	Master/Slave	SPI data. Output in master mode. Input in slave mode.
ufm_sn	I	Slave	Configuration Slave Chip Select (active low), dedicated for selecting the User Flash Memory (UFM).
cfg_stdby	O	Master/Slave	Stand-by signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, SPI Tab.
cfg_wake	O	Master/Slave	Wake-up signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, SPI Tab.

Hardened Timer/Counter

MachXO2 devices provide 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. The Timer/Counter supports the following functions:

- Supports the following modes of operation:
  - Watchdog timer
  - Clear timer on compare match
  - Fast PWM
  - Phase and Frequency Correct PWM
- Programmable clock input source
- Programmable input clock prescaler
- One static interrupt output to routing
- One wake-up interrupt to on-chip standby mode controller.
- Three independent interrupt sources: overflow, output compare match, and input capture
- Auto reload
- Time-stamping support on the input capture unit
- Waveform generation on the output
- Glitch-free PWM waveform generation with variable PWM period
- Internal WISHBONE bus access to the control and status registers
- Stand-alone mode with preloaded control registers and direct reset input

Figure 2-23. Timer/Counter Block Diagram

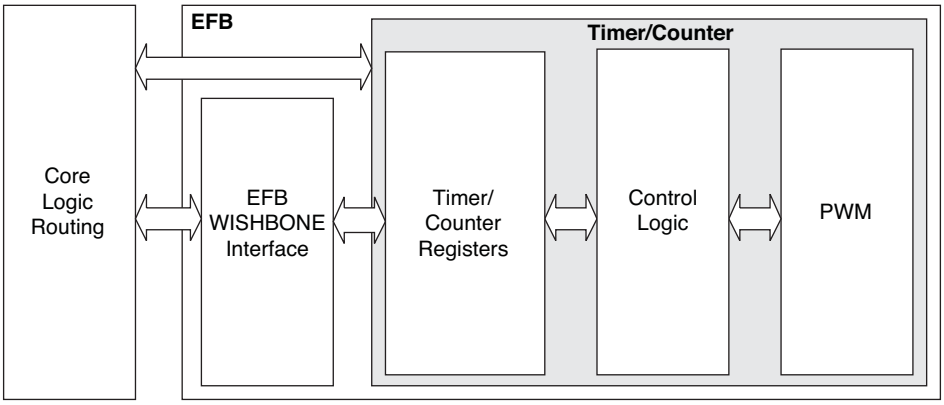


Table 2-17. Timer/Counter Signal Description

Port	I/O	Description
tc_clki	I	Timer/Counter input clock signal
tc_rstn	I	Register tc_rstn_ena is preloaded by configuration to always keep this pin enabled
tc_ic	I	Input capture trigger event, applicable for non-pwm modes with WISHBONE interface. If enabled, a rising edge of this signal will be detected and synchronized to capture tc_cnt value into tc_icr for time-stamping.
tc_int	O	Without WISHBONE – Can be used as overflow flag With WISHBONE – Controlled by three IRQ registers
tc_oc	O	Timer counter output signal

For more details on these embedded functions, please refer to TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

User Flash Memory (UFM)

MachXO2-640/U and higher density devices provide a User Flash Memory block, which can be used for a variety of applications including storing a portion of the configuration image, initializing EBRs, to store PROM data or, as a general purpose user Flash memory. The UFM block connects to the device core through the embedded function block WISHBONE interface. Users can also access the UFM block through the JTAG, I^2C and SPI interfaces of the device. The UFM block offers the following features:

- Non-volatile storage up to 256Kbits
- 100K write cycles
- Write access is performed page-wise; each page has 128 bits (16 bytes)
- Auto-increment addressing
- WISHBONE interface

For more information on the UFM, please refer to TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

Standby Mode and Power Saving Options

MachXO2 devices are available in three options for maximum flexibility: ZE, HC and HE devices. The ZE devices have ultra low static and dynamic power consumption. These devices use a 1.2V core voltage that further reduces power consumption. The HC and HE devices are designed to provide high performance. The HC devices have a built-in voltage regulator to allow for 2.5V V_{CC} and 3.3V V_{CC} while the HE devices operate at 1.2V V_{CC}.

MachXO2 devices have been designed with features that allow users to meet the static and dynamic power requirements of their applications by controlling various device subsystems such as the bandgap, power-on-reset circuitry, I/O bank controllers, power guard, on-chip oscillator, PLLs, etc. In order to maximize power savings, MachXO2 devices support an ultra low power Stand-by mode. While most of these features are available in all three device types, these features are mainly intended for use with MachXO2 ZE devices to manage power consumption.

In the stand-by mode the MachXO2 devices are powered on and configured. Internal logic, I/Os and memories are switched on and remain operational, as the user logic waits for an external input. The device enters this mode when the standby input of the standby controller is toggled or when an appropriate I^2C or JTAG instruction is issued by an external master. Various subsystems in the device such as the band gap, power-on-reset circuitry etc can be configured such that they are automatically turned “off” or go into a low power consumption state to save power when the device enters this state.

Table 2-18. MachXO2 Power Saving Features Description

Device Subsystem	Feature Description
Bandgap	The bandgap can be turned off in standby mode. When the Bandgap is turned off, analog circuitry such as the POR, PLLs, on-chip oscillator, and referenced and differential I/O buffers are also turned off. Bandgap can only be turned off for 1.2V devices.
Power-On-Reset (POR)	The POR can be turned off in standby mode. This monitors VCC levels. In the event of unsafe V_{CC} drops, this circuit reconfigures the device. When the POR circuitry is turned off, limited power detector circuitry is still active. This option is only recommended for applications in which the power supply rails are reliable.
On-Chip Oscillator	The on-chip oscillator has two power saving features. It may be switched off if it is not needed in your design. It can also be turned off in Standby mode.
PLL	Similar to the on-chip oscillator, the PLL also has two power saving features. It can be statically switched off if it is not needed in a design. It can also be turned off in Standby mode. The PLL will wait until all output clocks from the PLL are driven low before powering off.
I/O Bank Controller	Referenced and differential I/O buffers (used to implement standards such as HSTL, SSTL and LVDS) consume more than ratioed single-ended I/Os such as LVCMOS and LVTTL. The I/O bank controller allows the user to turn these I/Os off dynamically on a per bank selection.
Dynamic Clock Enable for Primary Clock Nets	Each primary clock net can be dynamically disabled to save power.
Power Guard	Power Guard is a feature implemented in input buffers. This feature allows users to switch off the input buffer when it is not needed. This feature can be used in both clock and data paths. Its biggest impact is that in the standby mode it can be used to switch off clock inputs that are distributed using general routing resources.

For more details on the standby mode refer to TN1198, Power Estimation and Management for MachXO2 Devices.

Power On Reset

MachXO2 devices have power-on reset circuitry to monitor V_{CCINT} and V_{CCIO} voltage levels during power-up and operation. At power-up, the POR circuitry monitors V_{CCINT} and V_{CCIO0} (controls configuration) voltage levels. It then triggers download from the on-chip configuration Flash memory after reaching the V_{PORUP} level specified in the Power-On-Reset Voltage table in the DC and Switching Characteristics section of this data sheet. For devices without voltage regulators (ZE and HE devices), V_{CCINT} is the same as the V_{CC} supply voltage. For devices with voltage regulators (HC devices), V_{CCINT} is regulated from the V_{CC} supply voltage. From this voltage reference, the time taken for configuration and entry into user mode is specified as Flash Download Time ($t_{REFRESH}$) in the DC and Switching Characteristics section of this data sheet. Before and during configuration, the I/Os are held in tristate. I/Os are released to user functionality once the device has finished configuration. Note that for HC devices, a separate POR circuit monitors external V_{CC} voltage in addition to the POR circuit that monitors the internal post-regulated power supply voltage level.

Once the device enters into user mode, the POR circuitry can optionally continue to monitor V_{CCINT} levels. If V_{CCINT} drops below $V_{PORDNBG}$ level (with the bandgap circuitry switched on) or below $V_{PORDNSRAM}$ level (with the bandgap circuitry switched off to conserve power) device functionality cannot be guaranteed. In such a situation the POR issues a reset and begins monitoring the V_{CCINT} and V_{CCIO} voltage levels. $V_{PORDNBG}$ and $V_{PORDNSRAM}$ are both specified in the Power-On-Reset Voltage table in the DC and Switching Characteristics section of this data sheet.

Note that once a ZE or HE device enters user mode, users can switch off the bandgap to conserve power. When the bandgap circuitry is switched off, the POR circuitry also shuts down. The device is designed such that a minimal, low power POR circuit is still operational (this corresponds to the $V_{PORDNSRAM}$ reset point described in the paragraph above). However this circuit is not as accurate as the one that operates when the bandgap is switched on. The low power POR circuit emulates an SRAM cell and is biased to trip before the vast majority of SRAM cells flip. If users are concerned about the V_{CC} supply dropping below V_{CC} (min) they should not shut down the bandgap or POR circuit.

Configuration and Testing

This section describes the configuration and testing features of the MachXO2 family.

IEEE 1149.1-Compliant Boundary Scan Testability

All MachXO2 devices have boundary scan cells that are accessed through an IEEE 1149.1 compliant test access port (TAP). This allows functional testing of the circuit board, on which the device is mounted, through a serial scan path that can access all critical logic nodes. Internal registers are linked internally, allowing test data to be shifted in and loaded directly onto test nodes, or test data to be captured and shifted out for verification. The test access port consists of dedicated I/Os: TDI, TDO, TCK and TMS. The test access port shares its power supply with V_{CCIO} Bank 0 and can operate with LVCMOS3.3, 2.5, 1.8, 1.5, and 1.2 standards.

For more details on boundary scan test, see AN8066, Boundary Scan Testability with Lattice sysIO Capability and TN1087, Minimizing System Interruption During Configuration Using TransFR Technology.

Device Configuration

All MachXO2 devices contain two ports that can be used for device configuration. The Test Access Port (TAP), which supports bit-wide configuration and the sysCONFIG port which supports serial configuration through I^2C or SPI. The TAP supports both the IEEE Standard 1149.1 Boundary Scan specification and the IEEE Standard 1532 In-System Configuration specification. There are various ways to configure a MachXO2 device:

1. Internal Flash Download
2. JTAG
3. Standard Serial Peripheral Interface (Master SPI mode) – interface to boot PROM memory
4. System microprocessor to drive a serial slave SPI port (SSPI mode)
5. Standard I^2C Interface to system microprocessor

Upon power-up, the configuration SRAM is ready to be configured using the selected sysCONFIG port. Once a configuration port is selected, it will remain active throughout that configuration cycle. The IEEE 1149.1 port can be activated any time after power-up by sending the appropriate command through the TAP port. Optionally the device can run a CRC check upon entering the user mode. This will ensure that the device was configured correctly.

The sysCONFIG port has 10 dual-function pins which can be used as general purpose I/Os if they are not required for configuration. See TN1204, MachXO2 Programming and Configuration Usage Guide for more information about using the dual-use pins as general purpose I/Os.

Lattice design software uses proprietary compression technology to compress bit-streams for use in MachXO2 devices. Use of this technology allows Lattice to provide a lower cost solution. In the unlikely event that this technology is unable to compress bitstreams to fit into the amount of on-chip Flash memory, there are a variety of techniques that can be utilized to allow the bitstream to fit in the on-chip Flash memory. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

The Test Access Port (TAP) has five dual purpose pins (TDI, TDO, TMS and TCK). These pins are dual function pins - TDI, TDO, TMS and TCK can be used as general purpose I/O if desired. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

TransFR (Transparent Field Reconfiguration)

TransFR is a unique Lattice technology that allows users to update their logic in the field without interrupting system operation using a simple push-button solution. For more details refer to TN1087, Minimizing System Interruption During Configuration Using TransFR Technology for details.

Security and One-Time Programmable Mode (OTP)

For applications where security is important, the lack of an external bitstream provides a solution that is inherently more secure than SRAM-based FPGAs. This is further enhanced by device locking. MachXO2 devices contain security bits that, when set, prevent the readback of the SRAM configuration and non-volatile Flash memory spaces. The device can be in one of two modes:

1. Unlocked – Readback of the SRAM configuration and non-volatile Flash memory spaces is allowed.

2. Permanently Locked – The device is permanently locked.

Once set, the only way to clear the security bits is to erase the device. To further complement the security of the device, a One Time Programmable (OTP) mode is available. Once the device is set in this mode it is not possible to erase or re-program the Flash and SRAM OTP portions of the device. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

Dual Boot

MachXO2 devices can optionally boot from two patterns, a primary bitstream and a golden bitstream. If the primary bitstream is found to be corrupt while being downloaded into the SRAM, the device shall then automatically re-boot from the golden bitstream. Note that the primary bitstream must reside in the on-chip Flash. The golden image MUST reside in an external SPI Flash. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

Soft Error Detection

The SED feature is a CRC check of the SRAM cells after the device is configured. This check ensures that the SRAM cells were configured successfully. This feature is enabled by a configuration bit option. The Soft Error Detection can also be initiated in user mode via an input to the fabric. The clock for the Soft Error Detection circuit is generated using a dedicated divider. The undivided clock from the on-chip oscillator is the input to this divider. For low power applications users can switch off the Soft Error Detection circuit. For more details, refer to TN1206, MachXO2 Soft Error Detection Usage Guide.

TraceID

Each MachXO2 device contains a unique (per device), TraceID that can be used for tracking purposes or for IP security applications. The TraceID is 64 bits long. Eight out of 64 bits are user-programmable, the remaining 56 bits are factory-programmed. The TraceID is accessible through the EFB WISHBONE interface and can also be accessed through the SPI, I^2C, or JTAG interfaces.

Density Shifting

The MachXO2 family has been designed to enable density migration within the same package. Furthermore, the architecture ensures a high success rate when performing design migration from lower density devices to higher density devices. In many cases, it is also possible to shift a lower utilization design targeted for a high-density device to a lower density device. However, the exact details of the final resource utilization will impact the likely success in each case. For more details refer to the MachXO2 migration files.

MachXO2 Family Data Sheet DC and Switching Characteristics

January 2013 Data Sheet DS1035

Absolute Maximum Ratings[1, 2, 3, 4]

	MachXO2 ZE/HE (1.2V)	MachXO2 HC (2.5V/3.3V)
Supply Voltage V_{CC}	-0.5 to 1.32V	-0.5 to 3.75V
Output Supply Voltage V_{CCIO}	-0.5 to 3.75V	-0.5 to 3.75V
I/O Tri-state Voltage Applied[5]	-0.5 to 3.75V	-0.5 to 3.75V
Dedicated Input Voltage Applied	-0.5 to 3.75V	-0.5 to 3.75V
Storage Temperature (Ambient)	-55°C to 125°C	-55°C to 125°C
Junction Temperature (T_J)	-40°C to 125°C	-40°C to 125°C

1. Stress above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.
2. Compliance with the Lattice Thermal Management document is required.
3. All voltages referenced to GND.
4. Overshoot and undershoot of -2V to (V_{IHMAX} + 2) volts is permitted for a duration of <20ns.
5. The dual function I^2C pins SCL and SDA are limited to -0.25V to 3.75V or to -0.3V with a duration of <20ns.

Recommended Operating Conditions[1]

Symbol	Parameter	Min.	Max.	Units
V_{CC}[1]	Core Supply Voltage for 1.2V Devices	1.14	1.26	V
	Core Supply Voltage for 2.5V/3.3V Devices	2.375	3.465	V
V_{CCIO}[1, 2, 3]	I/O Driver Supply Voltage	1.14	3.465	V
t_{JCOM}	Junction Temperature Commercial Operation	0	85	°C
t_{JIND}	Junction Temperature Industrial Operation	-40	100	°C

1. Like power supplies must be tied together. For example, if V_{CCIO} and V_{CC} are both the same voltage, they must also be the same supply.
2. See recommended voltages by I/O standard in subsequent table.
3. V_{CCIO} pins of unused I/O banks should be connected to the V_{CC} power supply on boards.

Power Supply Ramp Rates[1]

Symbol	Parameter	Min.	Typ.	Max.	Units
t_{RAMP}	Power supply ramp rates for all power supplies.	0.01	—	100	V/ms

1. Assumes monotonic ramp rates.

Power-On-Reset Voltage Levels[1, 2, 3, 4]

Symbol	Parameter	Min.	Typ.	Max.	Units
V_{PORUP}	Power-On-Reset ramp up trip point (band gap based circuit monitoring V_{CCINT} and V_{CCIO})	0.9	—	1.06	V
$V_{PORUPEXT}$	Power-On-Reset ramp up trip point (band gap based circuit monitoring external V_{CC} power supply)	1.5	—	2.1	V
$V_{PORDNBG}$	Power-On-Reset ramp down trip point (band gap based circuit monitoring V_{CCINT})	—	—	0.93	V
$V_{PORDNSRAM}$	Power-On-Reset ramp down trip point (SRAM based circuit monitoring V_{CCINT})	—	0.6	—	V

1. These POR trip points are only provided for guidance. Device operation is only characterized for power supply voltages specified under recommended operating conditions.
2. For devices without voltage regulators V_{CCINT} is the same as the V_{CC} supply voltage. For devices with voltage regulators, V_{CCINT} is regulated from the V_{CC} supply voltage.
3. Note that V_{PORUP} (min.) and $V_{PORDNBG}$ (max.) are in different process corners. For any given process corner $V_{PORDNBG}$ (max.) is always 12.0mV below V_{PORUP} (min.).
4. $V_{PORUPEXT}$ is for HC devices only. In these devices a separate POR circuit monitors the external V_{CC} power supply.

Programming/Erase Specifications

Symbol	Parameter	Min.	Max.[1]	Units
$N_{PROGCYC}$	Flash Programming cycles per $t_{RETENTION}$	—	10,000	Cycles
	Flash functional programming cycles	—	100,000	
$t_{RETENTION}$	Data retention at 100°C junction temperature	10	—	Years
	Data retention at 85°C junction temperature	20	—	

1. Maximum Flash memory reads are limited to 7.5E13 cycles over the lifetime of the product.

Hot Socketing Specifications[1, 2, 3]

Symbol	Parameter	Condition	Max.	Units
I_{DK}	Input or I/O leakage Current	$0 < V_{IN} < V_{IH}$ (MAX)	+/-1000	µA

1. Insensitive to sequence of V_{CC} and V_{CCIO}. However, assumes monotonic rise/fall rates for V_{CC} and V_{CCIO}.
2. $0 < V_{CC} < V_{CC}$ (MAX), $0 < V_{CCIO} < V_{CCIO}$ (MAX).
3. I_{DK} is additive to I_{PU}, I_{PD} or I_{BH}.

ESD Performance

Please refer to the MachXO2 Product Family Qualification Summary for complete qualification data, including ESD performance.

DC Electrical Characteristics

Over Recommended Operating Conditions

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{IL}, I_{IH}[1, 4]	Input or I/O Leakage	Clamp OFF and $V_{CCIO} < V_{IN} < V_{IH}$ (MAX)	—	—	+175	µA
		Clamp OFF and $V_{IN} = V_{CCIO}$	-10	—	10	µA
		Clamp OFF and V_{CCIO} - 0.97V < V_{IN} < V_{CCIO}	-175	—	—	µA
		Clamp OFF and 0V < V_{IN} < V_{CCIO} - 0.97V	—	—	10	µA
		Clamp OFF and V_{IN} = GND	—	—	10	µA
		Clamp ON and 0V < V_{IN} < V_{CCIO}	—	—	10	µA
I_{PU}	I/O Active Pull-up Current	0 < V_{IN} < 0.7 V_{CCIO}	-30	—	-309	µA
I_{PD}	I/O Active Pull-down Current	V_{IL} (MAX) < V_{IN} < V_{CCIO}	30	—	305	µA
I_{BHLS}	Bus Hold Low sustaining current	$V_{IN} = V_{IL}$ (MAX)	30	—	—	µA
I_{BHHS}	Bus Hold High sustaining current	$V_{IN} = 0.7V_{CCIO}$	-30	—	—	µA
I_{BHLO}	Bus Hold Low Overdrive current	0 ≤ V_{IN} ≤ V_{CCIO}	—	—	305	µA
I_{BHHO}	Bus Hold High Overdrive current	0 ≤ V_{IN} ≤ V_{CCIO}	—	—	-309	µA
V_{BHT}[3]	Bus Hold Trip Points		V_{IL} (MAX)	—	V_{IH} (MIN)	V
C1	I/O Capacitance[2]	V_{CCIO} = 3.3V, 2.5V, 1.8V, 1.5V, 1.2V, V_{CC} = Typ., V_{IO} = 0 to V_{IH} (MAX)	3	5	9	pf
C2	Dedicated Input Capacitance[2]	V_{CCIO} = 3.3V, 2.5V, 1.8V, 1.5V, 1.2V, V_{CC} = Typ., V_{IO} = 0 to V_{IH} (MAX)	3	5.5	7	pf
V_{HYST}	Hysteresis for Schmitt Trigger Inputs[5]	V_{CCIO} = 3.3V, Hysteresis = Large	—	450	—	mV
		V_{CCIO} = 2.5V, Hysteresis = Large	—	250	—	mV
		V_{CCIO} = 1.8V, Hysteresis = Large	—	125	—	mV
		V_{CCIO} = 1.5V, Hysteresis = Large	—	100	—	mV
		V_{CCIO} = 3.3V, Hysteresis = Small	—	250	—	mV
		V_{CCIO} = 2.5V, Hysteresis = Small	—	150	—	mV
		V_{CCIO} = 1.8V, Hysteresis = Small	—	60	—	mV
		V_{CCIO} = 1.5V, Hysteresis = Small	—	40	—	mV

1. Input or I/O leakage current is measured with the pin configured as an input or as an I/O with the output driver tri-stated. It is not measured with the output driver active. Bus maintenance circuits are disabled.
2. T_A 25°C, f = 1.0MHz.
3. Please refer to V_{IL} and V_{IH} in the sysIO Single-Ended DC Electrical Characteristics table of this document.
4. When V_{IH} is higher than V_{CCIO}, a transient current typically of 30ns in duration or less with a peak current of 6mA can occur on the high-to-low transition. For true LVDS output pins in MachXO2-640U, MachXO2-1200/U and larger devices, V_{IH} must be less than or equal to V_{CCIO}.
5. With bus keeper circuit turned on. For more details, refer to TN1202, MachXO2 sysIO Usage Guide.

Static Supply Current – ZE Devices[1, 2, 3, 6]

Symbol	Parameter	Device	Typ.[4]	Units
I_{CC}	Core Power Supply	LCMXO2-256ZE	18	µA
		LCMXO2-640ZE	28	µA
		LCMXO2-1200ZE	56	µA
		LCMXO2-2000ZE	80	µA
		LCMXO2-4000ZE	124	µA
		LCMXO2-7000ZE	189	µA
I_{CCIO}	Bank Power Supply[5] V_{CCIO} = 2.5V	All devices	0	mA

1. For further information on supply current, please refer to TN1198, Power Estimation and Management for MachXO2 Devices.
2. Assumes blank pattern with the following characteristics: all outputs are tri-stated, all inputs are configured as LVCMOS and held at V_{CCIO} or GND, on-chip oscillator is off, on-chip PLL is off. To estimate the impact of turning each of these items on, please refer to the following table or for more detail with your specific design use the Power Calculator tool.
3. Frequency = 0 MHz.
4. T_J = 25°C, power supplies at nominal voltage.
5. Does not include pull-up/pull-down.
6. To determine the MachXO2 peak start-up current data, use the Power Calculator tool.

Static Power Consumption Contribution of Different Components – ZE Devices

The table below can be used for approximating static power consumption. For a more accurate power analysis for your design please use the Power Calculator tool.

Symbol	Parameter	Typ.	Units
I_{DCBG}	Bandgap DC power contribution	101	µA
I_{DCPOR}	POR DC power contribution	38	µA
$I_{DCIOBANKCONTROLLER}$	DC power contribution per I/O bank controller	143	µA

Static Supply Current – HC/HE Devices[1, 2, 3, 6]

Symbol	Parameter	Device	Typ.[4]	Units
I_{CC}	Core Power Supply	LCMXO2-256HC	1.15	mA
		LCMXO2-640HC	1.84	mA
		LCMXO2-640UHC	3.48	mA
		LCMXO2-1200HC	3.49	mA
		LCMXO2-1200UHC	4.80	mA
		LCMXO2-2000HC	4.80	mA
		LCMXO2-2000UHC	8.44	mA
		LCMXO2-4000HC	8.45	mA
		LCMXO2-7000HC	12.87	mA
		LCMXO2-2000HE	1.39	mA
		LCMXO2-4000HE	2.55	mA
		LCMXO2-7000HE	4.06	mA
I_{CCIO}	Bank Power Supply[5] V_{CCIO} = 2.5V	All devices	0	mA

1. For further information on supply current, please refer to TN1198, Power Estimation and Management for MachXO2 Devices.
2. Assumes blank pattern with the following characteristics: all outputs are tri-stated, all inputs are configured as LVCMOS and held at V_{CCIO} or GND, on-chip oscillator is off, on-chip PLL is off.
3. Frequency = 0 MHz.
4. T_J = 25°C, power supplies at nominal voltage.
5. Does not include pull-up/pull-down.
6. To determine the MachXO2 peak start-up current data, use the Power Calculator tool.

Programming and Erase Flash Supply Current – ZE Devices[1, 2, 3, 4]

Symbol	Parameter	Device	Typ.[5]	Units
I_{CC}	Core Power Supply	LCMXO2-256ZE	13	mA
		LCMXO2-640ZE	14	mA
		LCMXO2-1200ZE	15	mA
		LCMXO2-2000ZE	17	mA
		LCMXO2-4000ZE	18	mA
		LCMXO2-7000ZE	20	mA
I_{CCIO}	Bank Power Supply[6]	All devices	0	mA

1. For further information on supply current, please refer to TN1198, Power Estimation and Management for MachXO2 Devices.
2. Assumes all inputs are held at V_{CCIO} or GND and all outputs are tri-stated.
3. Typical user pattern.
4. JTAG programming is at 25 MHz.
5. TJ = 25°C, power supplies at nominal voltage.
6. Per bank. V_{CCIO} = 2.5V. Does not include pull-up/pull-down.

Programming and Erase Flash Supply Current – HC/HE Devices[1, 2, 3, 4]

Symbol	Parameter	Device	Typ.[5]	Units
I_{CC}	Core Power Supply	LCMXO2-256HC	14.6	mA
		LCMXO2-640HC	16.1	mA
		LCMXO2-640UHC	18.8	mA
		LCMXO2-1200HC	18.8	mA
		LCMXO2-1200UHC	22.1	mA
		LCMXO2-2000HC	22.1	mA
		LCMXO2-2000UHC	26.8	mA
		LCMXO2-4000HC	26.8	mA
		LCMXO2-7000HC	33.2	mA
		LCMXO2-2000HE	18.3	mA
		LCMXO2-2000UHE	20.4	mA
		LCMXO2-4000HE	20.4	mA
		LCMXO2-7000HE	23.9	mA
I_{CCIO}	Bank Power Supply[6]	All devices	0	mA

1. For further information on supply current, please refer to TN1198, Power Estimation and Management for MachXO2 Devices.
2. Assumes all inputs are held at V_{CCIO} or GND and all outputs are tri-stated.
3. Typical user pattern.
4. JTAG programming is at 25 MHz.
5. T_J = 25°C, power supplies at nominal voltage.
6. Per bank. V_{CCIO} = 2.5V. Does not include pull-up/pull-down.

sysIO Recommended Operating Conditions

Standard	V_{CCIO} (V)			V_{REF} (V)		
	Min.	Typ.	Max.	Min.	Typ.	Max.
LVCMOS 3.3	3.135	3.3	3.465	—	—	—
LVCMOS 2.5	2.375	2.5	2.625	—	—	—
LVCMOS 1.8	1.71	1.8	1.89	—	—	—
LVCMOS 1.5	1.425	1.5	1.575	—	—	—
LVCMOS 1.2	1.14	1.2	1.26	—	—	—
LVTTL	3.135	3.3	3.465	—	—	—
PCI[3]	3.135	3.3	3.465	—	—	—
SSTL25	2.375	2.5	2.625	1.15	1.25	1.35
SSTL18	1.71	1.8	1.89	0.833	0.9	0.969
HSTL18	1.71	1.8	1.89	0.816	0.9	1.08
LVDS25[1, 2]	2.375	2.5	2.625	—	—	—
LVDS33[1, 2]	3.135	3.3	3.465	—	—	—
LVPECL[1]	3.135	3.3	3.465	—	—	—
BLVDS[1]	2.375	2.5	2.625	—	—	—
RSDS[1]	2.375	2.5	2.625	—	—	—
SSTL18D	1.71	1.8	1.89	—	—	—
SSTL25D	2.375	2.5	2.625	—	—	—
HSTL18D	1.71	1.8	1.89	—	—	—

1. Inputs on-chip. Outputs are implemented with the addition of external resistors.
2. MachXO2-640U, MachXO2-1200/U and larger devices have dedicated LVDS buffers
3. Input on the bottom bank of the MachXO2-640U, MachXO2-1200/U and larger devices only.

sysIO Single-Ended DC Electrical Characteristics[1, 2]

Input/Output Standard	V_{IL}		V_{IH}		V_{OL} Max. (V)	V_{OH} Min. (V)	I_{OL} Max.[4] (mA)	I_{OH} Max.[4] (mA)
	Min. (V)[3]	Max. (V)	Min. (V)	Max. (V)				
LVCMOS 3.3 LVTTL	-0.3	0.8	2.0	3.6	0.4	V_{CCIO} - 0.4	4	-4
							8	-8
							12	-12
							16	-16
							24	-24
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 2.5	-0.3	0.7	1.7	3.6	0.4	V_{CCIO} - 0.4	4	-4
							8	-8
							12	-12
							16	-16
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 1.8	-0.3	$0.35V_{CCIO}$	$0.65V_{CCIO}$	3.6	0.4	V_{CCIO} - 0.4	4	-4
							8	-8
							12	-12
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 1.5	-0.3	$0.35V_{CCIO}$	$0.65V_{CCIO}$	3.6	0.4	V_{CCIO} - 0.4	4	-4
							8	-8
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 1.2	-0.3	$0.35V_{CCIO}$	$0.65V_{CCIO}$	3.6	0.4	V_{CCIO} - 0.4	4	-2
							8	-6
					0.2	V_{CCIO} - 0.2	0.1	-0.1
PCI	-0.3	$0.3V_{CCIO}$	$0.5V_{CCIO}$	3.6	$0.1V_{CCIO}$	$0.9V_{CCIO}$	1.5	-0.5
SSTL25 Class I	-0.3	V_{REF} - 0.18	V_{REF} + 0.18	3.6	0.54	V_{CCIO} - 0.62	8	8
SSTL25 Class II	-0.3	V_{REF} - 0.18	V_{REF} +0.18	3.6	NA	NA	NA	NA
SSTL18 Class I	-0.3	V_{REF} - 0.125	V_{REF} +0.125	3.6	0.40	V_{CCIO} - 0.40	8	8
SSTL18 Class II	-0.3	V_{REF} - 0.125	V_{REF} +0.125	3.6	NA	NA	NA	NA
HSTL18 Class I	-0.3	V_{REF} - 0.1	V_{REF} +0.1	3.6	0.40	V_{CCIO} - 0.40	8	8
HSTL18 Class II	-0.3	V_{REF} - 0.1	V_{REF} +0.1	3.6	NA	NA	NA	NA

1. MachXO2 devices allow LVCMOS inputs to be placed in I/O banks where V_{CCIO} is different from what is specified in the applicable JEDEC specification. This is referred to as a ratioed input buffer. In a majority of cases this operation follows or exceeds the applicable JEDEC specification. The cases where MachXO2 devices do not meet the relevant JEDEC specification are documented in the table below.
2. MachXO2 devices allow for LVCMOS referenced I/Os which follow applicable JEDEC specifications. For more details about mixed mode operation please refer to please refer to TN1202, MachXO2 sysIO Usage Guide.
3. The dual function I^2C pins SCL and SDA are limited to a V_{IL} min of -0.25V or to -0.3V with a duration of <10ns.
4. The average DC current drawn by I/Os between GND connections, or between the last GND in an I/O bank and the end of an I/O bank, as shown in the logic signal connections table shall not exceed n * 8mA. Where n is the number of I/Os between bank GND connections or between the last GND in a bank and the end of a bank.

Input Standard	V_{CCIO} (V)	V_{IL} Max. (V)
LVCMOS 33	1.5	0.685
LVCMOS 25	1.5	1.687
LVCMOS 18	1.5	1.164

sysIO Differential Electrical Characteristics

The LVDS differential output buffers are available on the top side of MachXO2-640U, MachXO2-1200/U and higher density devices in the MachXO2 PLD family.

LVDS

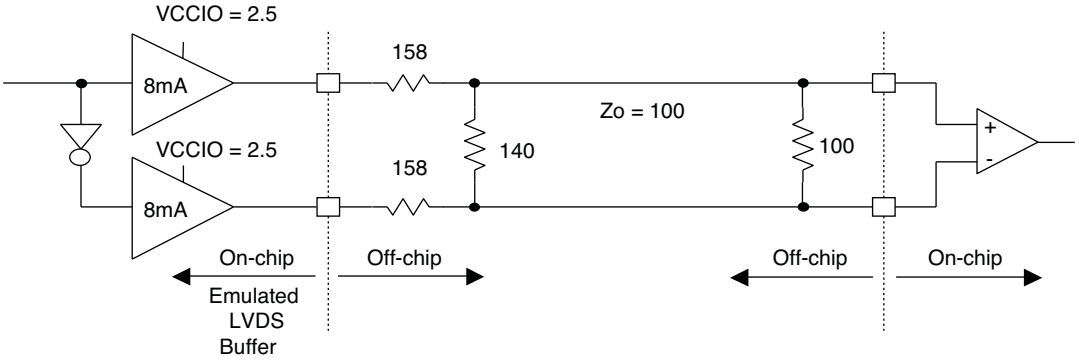
Over Recommended Operating Conditions

Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Units
V_{INP}, V_{INM}	Input Voltage	V_{CCIO} = 3.3	0	—	2.605	V
		V_{CCIO} = 2.5	0	—	2.05	V
V_{THD}	Differential Input Threshold		±100	—		mV
V_{CM}	Input Common Mode Voltage	V_{CCIO} = 3.3V	0.05	—	2.6	V
		V_{CCIO} = 2.5V	0.05	—	2.0	V
I_{IN}	Input current	Power on	—	—	±10	µA
V_{OH}	Output high voltage for V_{OP} or V_{OM}	R_T = 100 Ohm	—	1.375	—	V
V_{OL}	Output low voltage for V_{OP} or V_{OM}	R_T = 100 Ohm	0.90	1.025	—	V
V_{OD}	Output voltage differential	(V_{OP} - V_{OM}), R_T = 100 Ohm	250	350	450	mV
ΔV_{OD}	Change in V_{OD} between high and low		—	—	50	mV
V_{OS}	Output voltage offset	(V_{OP} - V_{OM})/2, R_T = 100 Ohm	1.125	1.20	1.395	V
ΔV_{OS}	Change in V_{OS} between H and L		—	—	50	mV
I_{OSD}	Output short circuit current	V_{OD} = 0V driver outputs shorted	—	—	24	mA

LVDS Emulation

MachXO2 devices can support LVDS outputs via emulation (LVDS25E). The output is emulated using complementary LVCMOS outputs in conjunction with resistors across the driver outputs on all devices. The scheme shown in Figure 3-1 is one possible solution for LVDS standard implementation. Resistor values in Figure 3-1 are industry standard values for 1% resistors.

Figure 3-1. LVDS Using External Resistors (LVDS25E)



Note: All resistors are ±1%.

Table 3-1. LVDS25E DC Conditions

Over Recommended Operating Conditions

Parameter	Description	Typ.	Units
Z_{OUT}	Output impedance	20	Ohms
R_S	Driver series resistor	158	Ohms
R_P	Driver parallel resistor	140	Ohms
R_T	Receiver termination	100	Ohms
V_{OH}	Output high voltage	1.43	V
V_{OL}	Output low voltage	1.07	V
V_{OD}	Output differential voltage	0.35	V
V_{CM}	Output common mode voltage	1.25	V
Z_{BACK}	Back impedance	100.5	Ohms
I_{DC}	DC output current	6.03	mA

BLVDS

The MachXO2 family supports the BLVDS standard through emulation. The output is emulated using complementary LVCMOS outputs in conjunction with resistors across the driver outputs. The input standard is supported by the LVDS differential input buffer. BLVDS is intended for use when multi-drop and bi-directional multi-point differential signaling is required. The scheme shown in Figure 3-2 is one possible solution for bi-directional multi-point differential signals.

Figure 3-2. BLVDS Multi-point Output Example

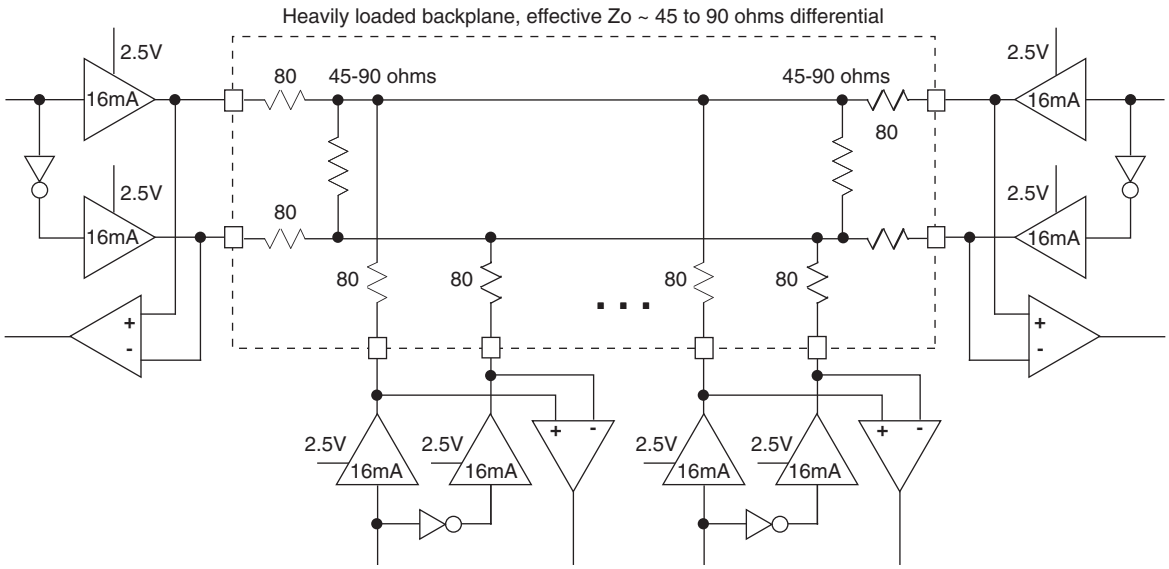


Table 3-2. BLVDS DC Conditions[1]

Over Recommended Operating Conditions

Symbol	Description	Nominal		Units
		Zo = 45	Zo = 90	
Z_{OUT}	Output impedance	10	10	Ohms
R_S	Driver series resistance	80	80	Ohms
R_{TLEFT}	Left end termination	45	90	Ohms
R_{TRIGHT}	Right end termination	45	90	Ohms
V_{OH}	Output high voltage	1.376	1.480	V
V_{OL}	Output low voltage	1.124	1.020	V
V_{OD}	Output differential voltage	0.253	0.459	V
V_{CM}	Output common mode voltage	1.250	1.250	V
I_{DC}	DC output current	11.236	10.204	mA

1. For input buffer, see LVDS table.

LVPECL

The MachXO2 family supports the differential LVPECL standard through emulation. This output standard is emulated using complementary LVCMOS outputs in conjunction with resistors across the driver outputs on all the devices. The LVPECL input standard is supported by the LVDS differential input buffer. The scheme shown in Differential LVPECL is one possible solution for point-to-point signals.

Figure 3-3. Differential LVPECL

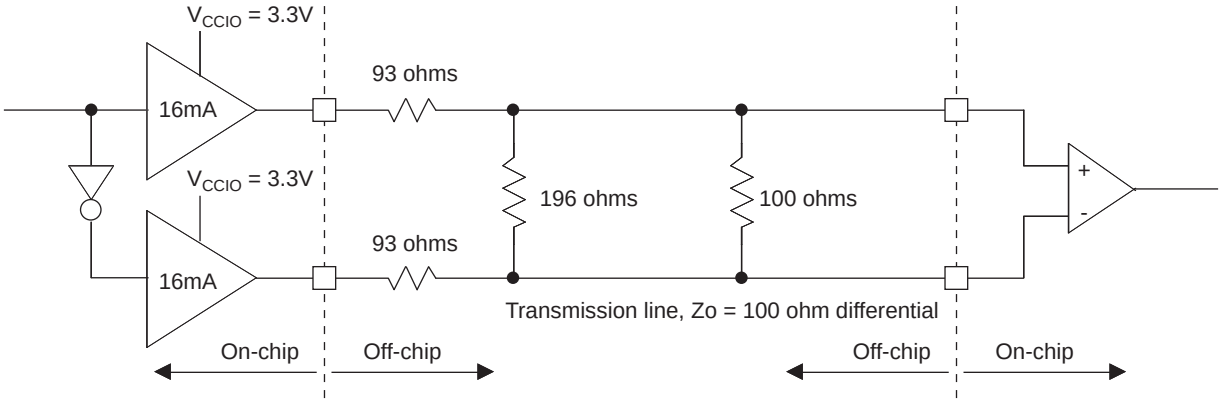


Table 3-3. LVPECL DC Conditions[1]

Over Recommended Operating Conditions

Symbol	Description	Nominal	Units
Z_{OUT}	Output impedance	10	Ohms
R_S	Driver series resistor	93	Ohms
R_P	Driver parallel resistor	196	Ohms
R_T	Receiver termination	100	Ohms
V_{OH}	Output high voltage	2.05	V
V_{OL}	Output low voltage	1.25	V
V_{OD}	Output differential voltage	0.80	V
V_{CM}	Output common mode voltage	1.65	V
Z_{BACK}	Back impedance	100.5	Ohms
I_{DC}	DC output current	12.11	mA

1. For input buffer, see LVDS table.

For further information on LVPECL, BLVDS and other differential interfaces please see details of additional technical documentation at the end of the data sheet.

RSDS

The MachXO2 family supports the differential RSDS standard. The output standard is emulated using complementary LVCMOS outputs in conjunction with resistors across the driver outputs on all the devices. The RSDS input standard is supported by the LVDS differential input buffer. The scheme shown in Figure 3-4 is one possible solution for RSDS standard implementation. Use LVDS25E mode with suggested resistors for RSDS operation. Resistor values in Figure 3-4 are industry standard values for 1% resistors.

Figure 3-4. RSDS (Reduced Swing Differential Standard)

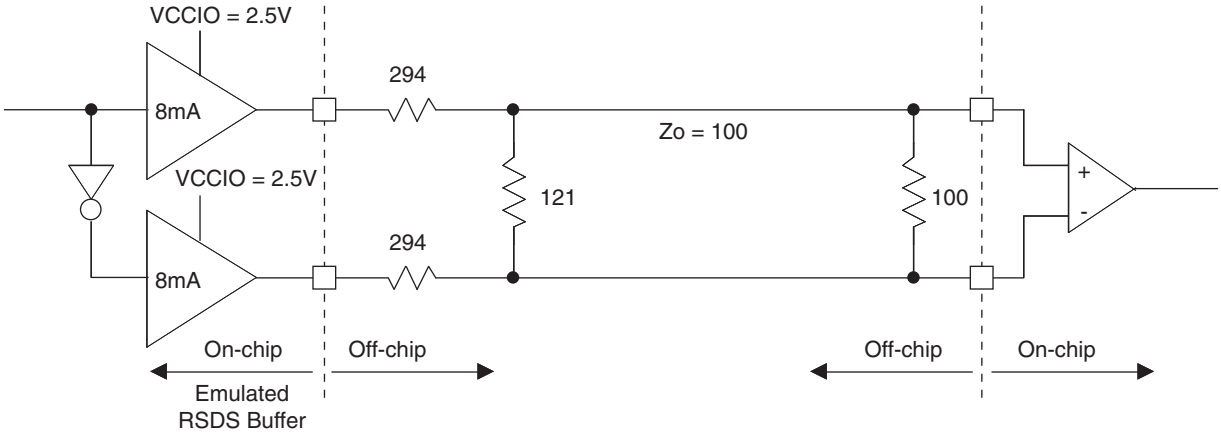


Table 3-4. RSDS DC Conditions

Parameter	Description	Typical	Units
Z_{OUT}	Output impedance	20	Ohms
R_S	Driver series resistor	294	Ohms
R_P	Driver parallel resistor	121	Ohms
R_T	Receiver termination	100	Ohms
V_{OH}	Output high voltage	1.35	V
V_{OL}	Output low voltage	1.15	V
V_{OD}	Output differential voltage	0.20	V
V_{CM}	Output common mode voltage	1.25	V
Z_{BACK}	Back impedance	101.5	Ohms
I_{DC}	DC output current	3.66	mA

Typical Building Block Function Performance – HC/HE Devices[1]

Pin-to-Pin Performance (LVCMOS25 12mA Drive)

Function	-6 Timing	Units
Basic Functions		
16-bit decoder	8.9	ns
4:1 MUX	7.5	ns
16:1 MUX	8.3	ns

Register-to-Register Performance

Function	-6 Timing	Units
Basic Functions		
16:1 MUX	412	MHz
16-bit adder	297	MHz
16-bit counter	324	MHz
64-bit counter	161	MHz
Embedded Memory Functions		
1024x9 True-Dual Port RAM (Write Through or Normal, EBR output registers)	183	MHz
Distributed Memory Functions		
16x4 Pseudo-Dual Port RAM (one PFU)	500	MHz

1. The above timing numbers are generated using the Diamond design tool. Exact performance may vary with device and tool version. The tool uses internal parameters that have been characterized but are not tested on every device.

Typical Building Block Function Performance – ZE Devices[1]

Pin-to-Pin Performance (LVCMOS25 12mA Drive)

Function	-3 Timing	Units
Basic Functions		
16-bit decoder	13.9	ns
4:1 MUX	10.9	ns
16:1 MUX	12.0	ns

Register-to-Register Performance

Function	-3 Timing	Units
Basic Functions		
16:1 MUX	191	MHz
16-bit adder	134	MHz
16-bit counter	148	MHz
64-bit counter	77	MHz
Embedded Memory Functions		
1024x9 True-Dual Port RAM (Write Through or Normal, EBR output registers)	90	MHz
Distributed Memory Functions		
16x4 Pseudo-Dual Port RAM (one PFU)	214	MHz

1. The above timing numbers are generated using the Diamond design tool. Exact performance may vary with device and tool version. The tool uses internal parameters that have been characterized but are not tested on every device.

Derating Logic Timing

Logic timing provided in the following sections of the data sheet and the Lattice design tools are worst case numbers in the operating range. Actual delays may be much faster. Lattice design tools can provide logic timing numbers at a particular temperature and voltage.

Maximum sysIO Buffer Performance

I/O Standard	Max. Speed	Units
LVDS25	400	MHz
LVDS25E	150	MHz
RSDS25	150	MHz
RSDS25E	150	MHz
BLVDS25	150	MHz
BLVDS25E	150	MHz
MLVDS25	150	MHz
MLVDS25E	150	MHz
LVPECL33	150	MHz
LVPECL33E	150	MHz
SSTL25_I	150	MHz
SSTL25_II	150	MHz
SSTL25D_I	150	MHz
SSTL25D_II	150	MHz
SSTL18_I	150	MHz
SSTL18_II	150	MHz
SSTL18D_I	150	MHz
SSTL18D_II	150	MHz
HSTL18_I	150	MHz
HSTL18_II	150	MHz
HSTL18D_I	150	MHz
HSTL18D_II	150	MHz
PCI33	134	MHz
LVTTL33	150	MHz
LVTTL33D	150	MHz
LVCMOS33	150	MHz
LVCMOS33D	150	MHz
LVCMOS25	150	MHz
LVCMOS25D	150	MHz
LVCMOS25R33	150	MHz
LVCMOS18	150	MHz
LVCMOS18D	150	MHz
LVCMOS18R33	150	MHz
LVCMOS18R25	150	MHz
LVCMOS15	150	MHz
LVCMOS15D	150	MHz
LVCMOS15R33	150	MHz
LVCMOS15R25	150	MHz
LVCMOS12	91	MHz
LVCMOS12D	91	MHz

MachXO2 External Switching Characteristics – HC/HE Devices[1, 2, 3, 4, 5, 6, 7]

Over Recommended Operating Conditions

Parameter	Description	Device	-6		-5		-4		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
Clocks									
Primary Clocks									
f_{MAX_PRI}[8]	Frequency for Primary Clock Tree	All MachXO2 devices	—	388	—	323	—	269	MHz
t_{W_PRI}	Clock Pulse Width for Primary Clock	All MachXO2 devices	0.5	—	0.6	—	0.7	—	ns
t_{SKEW_PRI}	Primary Clock Skew Within a Device	MachXO2-256HC-HE	—	912	—	939	—	975	ps
		MachXO2-640HC-HE	—	844	—	871	—	908	ps
		MachXO2-1200HC-HE	—	868	—	902	—	951	ps
		MachXO2-2000HC-HE	—	867	—	897	—	941	ps
		MachXO2-4000HC-HE	—	865	—	892	—	931	ps
		MachXO2-7000HC-HE	—	902	—	942	—	989	ps
Edge Clock									
f_{MAX_EDGE}[8]	Frequency for Edge Clock	MachXO2-1200 and larger devices	—	400	—	333	—	278	MHz
Pin-LUT-Pin Propagation Delay									
t_{PD}	Best case propagation delay through one LUT-4	All MachXO2 devices	—	6.72	—	6.96	—	7.24	ns
General I/O Pin Parameters (Using Primary Clock without PLL)									
t_{CO}	Clock to Output - PIO Output Register	MachXO2-256HC-HE	—	7.13	—	7.30	—	7.57	ns
		MachXO2-640HC-HE	—	7.15	—	7.30	—	7.57	ns
		MachXO2-1200HC-HE	—	7.44	—	7.64	—	7.94	ns
		MachXO2-2000HC-HE	—	7.46	—	7.66	—	7.96	ns
		MachXO2-4000HC-HE	—	7.51	—	7.71	—	8.01	ns
		MachXO2-7000HC-HE	—	7.54	—	7.75	—	8.06	ns
t_{SU}	Clock to Data Setup - PIO Input Register	MachXO2-256HC-HE	-0.06	—	-0.06	—	-0.06	—	ns
		MachXO2-640HC-HE	-0.06	—	-0.06	—	-0.06	—	ns
		MachXO2-1200HC-HE	-0.17	—	-0.17	—	-0.17	—	ns
		MachXO2-2000HC-HE	-0.20	—	-0.20	—	-0.20	—	ns
		MachXO2-4000HC-HE	-0.23	—	-0.23	—	-0.23	—	ns
		MachXO2-7000HC-HE	-0.23	—	-0.23	—	-0.23	—	ns
t_{H}	Clock to Data Hold - PIO Input Register	MachXO2-256HC-HE	1.75	—	1.95	—	2.16	—	ns
		MachXO2-640HC-HE	1.75	—	1.95	—	2.16	—	ns
		MachXO2-1200HC-HE	1.88	—	2.12	—	2.36	—	ns
		MachXO2-2000HC-HE	1.89	—	2.13	—	2.37	—	ns
		MachXO2-4000HC-HE	1.94	—	2.18	—	2.43	—	ns
		MachXO2-7000HC-HE	1.98	—	2.23	—	2.49	—	ns

Parameter	Description	Device	-6		-5		-4		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{SU_DEL}	Clock to Data Setup - PIO Input Register with Data Input Delay	MachXO2-256HC-HE	1.42	—	1.59	—	1.96	—	ns
		MachXO2-640HC-HE	1.41	—	1.58	—	1.96	—	ns
		MachXO2-1200HC-HE	1.63	—	1.79	—	2.17	—	ns
		MachXO2-2000HC-HE	1.61	—	1.76	—	2.13	—	ns
		MachXO2-4000HC-HE	1.66	—	1.81	—	2.19	—	ns
		MachXO2-7000HC-HE	1.53	—	1.67	—	2.03	—	ns
t_{H_DEL}	Clock to Data Hold - PIO Input Register with Input Data Delay	MachXO2-256HC-HE	-0.24	—	-0.24	—	-0.24	—	ns
		MachXO2-640HC-HE	-0.23	—	-0.23	—	-0.23	—	ns
		MachXO2-1200HC-HE	-0.24	—	-0.24	—	-0.24	—	ns
		MachXO2-2000HC-HE	-0.23	—	-0.23	—	-0.23	—	ns
		MachXO2-4000HC-HE	-0.25	—	-0.25	—	-0.25	—	ns
		MachXO2-7000HC-HE	-0.21	—	-0.21	—	-0.21	—	ns
f_{MAX_IO}	Clock Frequency of I/O and PFU Register	All MachXO2 devices	—	388	—	323	—	269	MHz
General I/O Pin Parameters (Using Edge Clock without PLL)									
t_{COE}	Clock to Output - PIO Output Register	MachXO2-1200HC-HE	—	7.53	—	7.76	—	8.10	ns
		MachXO2-2000HC-HE	—	7.53	—	7.76	—	8.10	ns
		MachXO2-4000HC-HE	—	7.45	—	7.68	—	8.00	ns
		MachXO2-7000HC-HE	—	7.53	—	7.76	—	8.10	ns
t_{SUE}	Clock to Data Setup - PIO Input Register	MachXO2-1200HC-HE	-0.19	—	-0.19	—	-0.19	—	ns
		MachXO2-2000HC-HE	-0.19	—	-0.19	—	-0.19	—	ns
		MachXO2-4000HC-HE	-0.16	—	-0.16	—	-0.16	—	ns
		MachXO2-7000HC-HE	-0.19	—	-0.19	—	-0.19	—	ns
t_{HE}	Clock to Data Hold - PIO Input Register	MachXO2-1200HC-HE	1.97	—	2.24	—	2.52	—	ns
		MachXO2-2000HC-HE	1.97	—	2.24	—	2.52	—	ns
		MachXO2-4000HC-HE	1.89	—	2.16	—	2.43	—	ns
		MachXO2-7000HC-HE	1.97	—	2.24	—	2.52	—	ns
t_{SU_DELE}	Clock to Data Setup - PIO Input Register with Data Input Delay	MachXO2-1200HC-HE	1.56	—	1.69	—	2.05	—	ns
		MachXO2-2000HC-HE	1.56	—	1.69	—	2.05	—	ns
		MachXO2-4000HC-HE	1.74	—	1.88	—	2.25	—	ns
		MachXO2-7000HC-HE	1.66	—	1.81	—	2.17	—	ns
t_{H_DELE}	Clock to Data Hold - PIO Input Register with Input Data Delay	MachXO2-1200HC-HE	-0.23	—	-0.23	—	-0.23	—	ns
		MachXO2-2000HC-HE	-0.23	—	-0.23	—	-0.23	—	ns
		MachXO2-4000HC-HE	-0.34	—	-0.34	—	-0.34	—	ns
		MachXO2-7000HC-HE	-0.29	—	-0.29	—	-0.29	—	ns
General I/O Pin Parameters (Using Primary Clock with PLL)									
t_{COPLL}	Clock to Output - PIO Output Register	MachXO2-1200HC-HE	—	5.97	—	6.00	—	6.13	ns
		MachXO2-2000HC-HE	—	5.98	—	6.01	—	6.14	ns
		MachXO2-4000HC-HE	—	5.99	—	6.02	—	6.16	ns
		MachXO2-7000HC-HE	—	6.02	—	6.06	—	6.20	ns
t_{SUPLL}	Clock to Data Setup - PIO Input Register	MachXO2-1200HC-HE	0.36	—	0.36	—	0.65	—	ns
		MachXO2-2000HC-HE	0.36	—	0.36	—	0.63	—	ns
		MachXO2-4000HC-HE	0.35	—	0.35	—	0.62	—	ns
		MachXO2-7000HC-HE	0.34	—	0.34	—	0.59	—	ns

Parameter	Description	Device	-6		-5		-4		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{HPLL}	Clock to Data Hold - PIO Input Register	MachXO2-1200HC-HE	0.41	—	0.48	—	0.55	—	ns
		MachXO2-2000HC-HE	0.42	—	0.49	—	0.56	—	ns
		MachXO2-4000HC-HE	0.43	—	0.50	—	0.58	—	ns
		MachXO2-7000HC-HE	0.46	—	0.54	—	0.62	—	ns
t_{SU_DELPLL}	Clock to Data Setup - PIO Input Register with Data Input Delay	MachXO2-1200HC-HE	2.88	—	3.19	—	3.72	—	ns
		MachXO2-2000HC-HE	2.87	—	3.18	—	3.70	—	ns
		MachXO2-4000HC-HE	2.96	—	3.28	—	3.81	—	ns
		MachXO2-7000HC-HE	3.05	—	3.35	—	3.87	—	ns
t_{H_DELPLL}	Clock to Data Hold - PIO Input Register with Input Data Delay	MachXO2-1200HC-HE	-0.83	—	-0.83	—	-0.83	—	ns
		MachXO2-2000HC-HE	-0.83	—	-0.83	—	-0.83	—	ns
		MachXO2-4000HC-HE	-0.87	—	-0.87	—	-0.87	—	ns
		MachXO2-7000HC-HE	-0.91	—	-0.91	—	-0.91	—	ns
Generic DDRX1 Inputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX1_RX.SCLK.Aligned[9]									
t_{DVA}	Input Data Valid After CLK	All MachXO2 devices, all sides	—	0.317	—	0.344	—	0.368	UI
t_{DVE}	Input Data Hold After CLK		0.742	—	0.702	—	0.668	—	UI
f_{DATA}	DDRX1 Input Data Speed		—	300	—	250	—	208	Mbps
f_{DDRX1}	DDRX1 SCLK Frequency		—	150	—	125	—	104	MHz
Generic DDRX1 Inputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX1_RX.SCLK.Centered[9]									
t_{SU}	Input Data Setup Before CLK	All MachXO2 devices, all sides	0.566	—	0.560	—	0.538	—	ns
t_{HO}	Input Data Hold After CLK		0.778	—	0.879	—	1.090	—	ns
f_{DATA}	DDRX1 Input Data Speed		—	300	—	250	—	208	Mbps
f_{DDRX1}	DDRX1 SCLK Frequency		—	150	—	125	—	104	MHz
Generic DDRX2 Inputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX2_RX.ECLK.Aligned[9]									
t_{DVA}	Input Data Valid After CLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	—	0.316	—	0.342	—	0.364	UI
t_{DVE}	Input Data Hold After CLK		0.710	—	0.675	—	0.679	—	UI
f_{DATA}	DDRX2 Serial Input Data Speed		—	664	—	554	—	462	Mbps
f_{DDRX2}	DDRX2 ECLK Frequency		—	332	—	277	—	231	MHz
f_{SCLK}	SCLK Frequency		—	166	—	139	—	116	MHz
Generic DDRX2 Inputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX2_RX.ECLK.Centered[9]									
t_{SU}	Input Data Setup Before CLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	0.233	—	0.219	—	0.198	—	ns
t_{HO}	Input Data Hold After CLK		0.287	—	0.287	—	0.344	—	ns
f_{DATA}	DDRX2 Serial Input Data Speed		—	664	—	554	—	462	Mbps
f_{DDRX2}	DDRX2 ECLK Frequency		—	332	—	277	—	231	MHz
f_{SCLK}	SCLK Frequency		—	166	—	139	—	116	MHz

Parameter	Description	Device	-6		-5		-4		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
Generic DDR4 Inputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX4_RX.ECLK.Aligned[9]									
t_{DVA}	Input Data Valid After ECLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	—	0.290	—	0.320	—	0.345	UI
t_{DVE}	Input Data Hold After ECLK		0.739	—	0.699	—	0.703	—	UI
f_{DATA}	DDRX4 Serial Input Data Speed		—	756	—	630	—	524	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency		—	378	—	315	—	262	MHz
f_{SCLK}	SCLK Frequency		—	95	—	79	—	66	MHz
Generic DDR4 Inputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX4_RX.ECLK.Centered[9]									
t_{SU}	Input Data Setup Before ECLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	0.233	—	0.219	—	0.198	—	ns
t_{HO}	Input Data Hold After ECLK		0.287	—	0.287	—	0.344	—	ns
f_{DATA}	DDRX4 Serial Input Data Speed		—	756	—	630	—	524	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency		—	378	—	315	—	262	MHz
f_{SCLK}	SCLK Frequency		—	95	—	79	—	66	MHz
7:1 LVDS Inputs (GDDR71_RX.ECLK.7:1)[9]									
t_{DVA}	Input Data Valid After ECLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	—	0.290	—	0.320	—	0.345	UI
t_{DVE}	Input Data Hold After ECLK		0.739	—	0.699	—	0.703	—	UI
f_{DATA}	DDR71 Serial Input Data Speed		—	756	—	630	—	524	Mbps
f_{DDR71}	DDR71 ECLK Frequency		—	378	—	315	—	262	MHz
f_{CLKIN}	7:1 Input Clock Frequency (SCLK) (minimum limited by PLL)		—	108	—	90	—	75	MHz
Generic DDR Outputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX1_TX.SCLK.Aligned[9]									
t_{DIA}	Output Data Invalid After CLK Output	All MachXO2 devices, all sides	—	0.520	—	0.550	—	0.580	ns
t_{DIB}	Output Data Invalid Before CLK Output		—	0.520	—	0.550	—	0.580	ns
f_{DATA}	DDRX1 Output Data Speed		—	300	—	250	—	208	Mbps
f_{DDRX1}	DDRX1 SCLK frequency		—	150	—	125	—	104	MHz
Generic DDR Outputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX1_TX.SCLK.Centered[9]									
t_{DVB}	Output Data Valid Before CLK Output	All MachXO2 devices, all sides	1.210	—	1.510	—	1.870	—	ns
t_{DVA}	Output Data Valid After CLK Output		1.210	—	1.510	—	1.870	—	ns
f_{DATA}	DDRX1 Output Data Speed		—	300	—	250	—	208	Mbps
f_{DDRX1}	DDRX1 SCLK Frequency (minimum limited by PLL)		—	150	—	125	—	104	MHz
Generic DDRX2 Outputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX2_TX.ECLK.Aligned[9]									
t_{DIA}	Output Data Invalid After CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	—	0.200	—	0.215	—	0.230	ns
t_{DIB}	Output Data Invalid Before CLK Output		—	0.200	—	0.215	—	0.230	ns
f_{DATA}	DDRX2 Serial Output Data Speed		—	664	—	554	—	462	Mbps
f_{DDRX2}	DDRX2 ECLK frequency		—	332	—	277	—	231	MHz
f_{SCLK}	SCLK Frequency		—	166	—	139	—	116	MHz

Parameter	Description	Device	-6		-5		-4		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
Generic DDRX2 Outputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX2_TX.ECLK.Centered[9]									
t_{DVB}	Output Data Valid Before CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	0.535	—	0.670	—	0.830	—	ns
t_{DVA}	Output Data Valid After CLK Output		0.535	—	0.670	—	0.830	—	ns
f_{DATA}	DDRX2 Serial Output Data Speed		—	664	—	554	—	462	Mbps
f_{DDRX2}	DDRX2 ECLK Frequency (minimum limited by PLL)		—	332	—	277	—	231	MHz
f_{SCLK}	SCLK Frequency		—	166	—	139	—	116	MHz
Generic DDRX4 Outputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX4_TX.ECLK.Aligned[9]									
t_{DIA}	Output Data Invalid After CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	—	0.200	—	0.215	—	0.230	ns
t_{DIB}	Output Data Invalid Before CLK Output		—	0.200	—	0.215	—	0.230	ns
f_{DATA}	DDRX4 Serial Output Data Speed		—	756	—	630	—	524	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency		—	378	—	315	—	262	MHz
f_{SCLK}	SCLK Frequency		—	95	—	79	—	66	MHz
Generic DDRX4 Outputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX4_TX.ECLK.Centered[9]									
t_{DVB}	Output Data Valid Before CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	0.455	—	0.570	—	0.710	—	ns
t_{DVA}	Output Data Valid After CLK Output		0.455	—	0.570	—	0.710	—	ns
f_{DATA}	DDRX4 Serial Output Data Speed		—	756	—	630	—	524	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency (minimum limited by PLL)		—	378	—	315	—	262	MHz
f_{SCLK}	SCLK Frequency		—	95	—	79	—	66	MHz
7:1 LVDS Outputs – GDDR71_TX.ECLK.7:1[9]									
t_{DVB}	Output Data Valid Before CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only.	—	0.160	—	0.180	—	0.200	ns
t_{DVA}	Output Data Valid After CLK Output		—	0.160	—	0.180	—	0.200	ns
f_{DATA}	DDR71 Serial Output Data Speed		—	756	—	630	—	524	Mbps
f_{DDR71}	DDR71 ECLK Frequency		—	378	—	315	—	262	MHz
f_{CLKOUT}	7:1 Output Clock Frequency (SCLK) (minimum limited by PLL)		—	108	—	90	—	75	MHz

Parameter	Description	Device	-6		-5		-4		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
LPDDR[9]									
t_{DVADQ}	Input Data Valid After DQS Input	MachXO2-1200/U and larger devices, right side only.	—	0.369	—	0.395	—	0.421	UI
t_{DVEDQ}	Input Data Hold After DQS Input		0.529	—	0.530	—	0.527	—	UI
t_{DQVBS}	Output Data Invalid Before DQS Output		0.25	—	0.25	—	0.25	—	UI
t_{DQVAS}	Output Data Invalid After DQS Output		0.25	—	0.25	—	0.25	—	UI
f_{DATA}	MEM LPDDR Serial Data Speed		—	280	—	250	—	208	Mbps
f_{SCLK}	SCLK Frequency		—	140	—	125	—	104	MHz
f_{LPDDR}	LPDDR Data Transfer Rate		0	280	0	250	0	208	Mbps
DDR[9]									
t_{DVADQ}	Input Data Valid After DQS Input	MachXO2-1200/U and larger devices, right side only.	—	0.350	—	0.387	—	0.414	UI
t_{DVEDQ}	Input Data Hold After DQS Input		0.545	—	0.538	—	0.532	—	UI
t_{DQVBS}	Output Data Invalid Before DQS Output		0.25	—	0.25	—	0.25	—	UI
t_{DQVAS}	Output Data Invalid After DQS Output		0.25	—	0.25	—	0.25	—	UI
f_{DATA}	MEM DDR Serial Data Speed		—	300	—	250	—	208	Mbps
f_{SCLK}	SCLK Frequency		—	150	—	125	—	104	MHz
f_{MEM_DDR}	MEM DDR Data Transfer Rate		N/A	300	N/A	250	N/A	208	Mbps
DDR2[9]									
t_{DVADQ}	Input Data Valid After DQS Input	MachXO2-1200/U and larger devices, right side only.	—	0.360	—	0.378	—	0.406	UI
t_{DVEDQ}	Input Data Hold After DQS Input		0.555	—	0.549	—	0.542	—	UI
t_{DQVBS}	Output Data Invalid Before DQS Output		0.25	—	0.25	—	0.25	—	UI
t_{DQVAS}	Output Data Invalid After DQS Output		0.25	—	0.25	—	0.25	—	UI
f_{DATA}	MEM DDR Serial Data Speed		—	300	—	250	—	208	Mbps
f_{SCLK}	SCLK Frequency		—	150	—	125	—	104	MHz
f_{MEM_DDR2}	MEM DDR2 Data Transfer Rate		N/A	300	N/A	250	N/A	208	Mbps

1. Exact performance may vary with device and design implementation. Commercial timing numbers are shown at 85°C and 1.14V. Other operating conditions, including industrial, can be extracted from the Diamond software.
2. General I/O timing numbers based on LVCMOS 2.5, 8mA, 0pf load.
3. Generic DDR timing numbers based on LVDS I/O (for input, output, and clock ports).
4. DDR timing numbers based on SSTL25. DDR2 timing numbers based on SSTL18. LPDDR timing numbers based in LVCMOS18.
5. 7:1 LVDS (GDDR71) uses the LVDS I/O standard (for input, output, and clock ports).
6. For Generic DDRX1 mode $t_{SU} = t_{HO} = (t_{DVE} - t_{DVA} - 0.03ns)/2$.
7. The t_{SU_DEL} and t_{H_DEL} values use the SCLK_ZERHOLD default step size. Each step is 105ps (-6), 113ps (-5), 120ps (-4).
8. This number for general purpose usage. Duty cycle tolerance is +/-10%.
9. Duty cycle is +/- 5% for system usage.
10. The above timing numbers are generated using the Diamond design tool. Exact performance may vary with the device selected.

MachXO2 External Switching Characteristics – ZE Devices[1, 2, 3, 4, 5, 6, 7]

Over Recommended Operating Conditions

Parameter	Description	Device	-3		-2		-1		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
Clocks									
Primary Clocks									
f_{MAX_PRI}[8]	Frequency for Primary Clock Tree	All MachXO2 devices	—	150	—	125	—	104	MHz
t_{W_PRI}	Clock Pulse Width for Primary Clock	All MachXO2 devices	1.00	—	1.20	—	1.40	—	ns
t_{SKEW_PRI}	Primary Clock Skew Within a Device	MachXO2-256ZE	—	1250	—	1272	—	1296	ps
		MachXO2-640ZE	—	1161	—	1183	—	1206	ps
		MachXO2-1200ZE	—	1213	—	1267	—	1322	ps
		MachXO2-2000ZE	—	1204	—	1250	—	1296	ps
		MachXO2-4000ZE	—	1195	—	1233	—	1269	ps
		MachXO2-7000ZE	—	1243	—	1268	—	1296	ps
Edge Clock									
f_{MAX_EDGE}[8]	Frequency for Edge Clock	MachXO2-1200 and larger devices	—	210	—	175	—	146	MHz
Pin-LUT-Pin Propagation Delay									
t_{PD}	Best case propagation delay through one LUT-4	All MachXO2 devices	—	9.35	—	9.78	—	10.21	ns
General I/O Pin Parameters (Using Primary Clock without PLL)									
t_{CO}	Clock to Output - PIO Output Register	MachXO2-256ZE	—	10.46	—	10.86	—	11.25	ns
		MachXO2-640ZE	—	10.52	—	10.92	—	11.32	ns
		MachXO2-1200ZE	—	11.24	—	11.68	—	12.12	ns
		MachXO2-2000ZE	—	11.27	—	11.71	—	12.16	ns
		MachXO2-4000ZE	—	11.28	—	11.78	—	12.28	ns
		MachXO2-7000ZE	—	11.22	—	11.76	—	12.30	ns
t_{SU}	Clock to Data Setup - PIO Input Register	MachXO2-256ZE	-0.21	—	-0.21	—	-0.21	—	ns
		MachXO2-640ZE	-0.22	—	-0.22	—	-0.22	—	ns
		MachXO2-1200ZE	-0.25	—	-0.25	—	-0.25	—	ns
		MachXO2-2000ZE	-0.27	—	-0.27	—	-0.27	—	ns
		MachXO2-4000ZE	-0.31	—	-0.31	—	-0.31	—	ns
		MachXO2-7000ZE	-0.33	—	-0.33	—	-0.33	—	ns
t_{H}	Clock to Data Hold - PIO Input Register	MachXO2-256ZE	3.96	—	4.25	—	4.65	—	ns
		MachXO2-640ZE	4.01	—	4.31	—	4.71	—	ns
		MachXO2-1200ZE	3.95	—	4.29	—	4.73	—	ns
		MachXO2-2000ZE	3.94	—	4.29	—	4.74	—	ns
		MachXO2-4000ZE	3.96	—	4.36	—	4.87	—	ns
		MachXO2-7000ZE	3.93	—	4.37	—	4.91	—	ns

Parameter	Description	Device	-3		-2		-1		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{SU_DEL}	Clock to Data Setup - PIO Input Register with Data Input Delay	MachXO2-256ZE	2.62	—	2.91	—	3.14	—	ns
		MachXO2-640ZE	2.56	—	2.85	—	3.08	—	ns
		MachXO2-1200ZE	2.30	—	2.57	—	2.79	—	ns
		MachXO2-2000ZE	2.25	—	2.50	—	2.70	—	ns
		MachXO2-4000ZE	2.39	—	2.60	—	2.76	—	ns
		MachXO2-7000ZE	2.17	—	2.33	—	2.43	—	ns
t_{H_DEL}	Clock to Data Hold - PIO Input Register with Input Data Delay	MachXO2-256ZE	-0.44	—	-0.44	—	-0.44	—	ns
		MachXO2-640ZE	-0.43	—	-0.43	—	-0.43	—	ns
		MachXO2-1200ZE	-0.28	—	-0.28	—	-0.28	—	ns
		MachXO2-2000ZE	-0.31	—	-0.31	—	-0.31	—	ns
		MachXO2-4000ZE	-0.34	—	-0.34	—	-0.34	—	ns
		MachXO2-7000ZE	-0.21	—	-0.21	—	-0.21	—	ns
f_{MAX_IO}	Clock Frequency of I/O and PFU Register	All MachXO2 devices	—	150	—	125	—	104	MHz
General I/O Pin Parameters (Using Edge Clock without PLL)									
t_{COE}	Clock to Output - PIO Output Register	MachXO2-1200ZE	—	11.10	—	11.51	—	11.91	ns
		MachXO2-2000ZE	—	11.10	—	11.51	—	11.91	ns
		MachXO2-4000ZE	—	10.89	—	11.28	—	11.67	ns
		MachXO2-7000ZE	—	11.10	—	11.51	—	11.91	ns
t_{SUE}	Clock to Data Setup - PIO Input Register	MachXO2-1200ZE	-0.23	—	-0.23	—	-0.23	—	ns
		MachXO2-2000ZE	-0.23	—	-0.23	—	-0.23	—	ns
		MachXO2-4000ZE	-0.15	—	-0.15	—	-0.15	—	ns
		MachXO2-7000ZE	-0.23	—	-0.23	—	-0.23	—	ns
t_{HE}	Clock to Data Hold - PIO Input Register	MachXO2-1200ZE	3.81	—	4.11	—	4.52	—	ns
		MachXO2-2000ZE	3.81	—	4.11	—	4.52	—	ns
		MachXO2-4000ZE	3.60	—	3.89	—	4.28	—	ns
		MachXO2-7000ZE	3.81	—	4.11	—	4.52	—	ns
t_{SU_DELE}	Clock to Data Setup - PIO Input Register with Data Input Delay	MachXO2-1200ZE	2.78	—	3.11	—	3.40	—	ns
		MachXO2-2000ZE	2.78	—	3.11	—	3.40	—	ns
		MachXO2-4000ZE	3.11	—	3.48	—	3.79	—	ns
		MachXO2-7000ZE	2.94	—	3.30	—	3.60	—	ns
t_{H_DELE}	Clock to Data Hold - PIO Input Register with Input Data Delay	MachXO2-1200ZE	-0.29	—	-0.29	—	-0.29	—	ns
		MachXO2-2000ZE	-0.29	—	-0.29	—	-0.29	—	ns
		MachXO2-4000ZE	-0.46	—	-0.46	—	-0.46	—	ns
		MachXO2-7000ZE	-0.37	—	-0.37	—	-0.37	—	ns
General I/O Pin Parameters (Using Primary Clock with PLL)									
t_{COPLL}	Clock to Output - PIO Output Register	MachXO2-1200ZE	—	7.95	—	8.07	—	8.19	ns
		MachXO2-2000ZE	—	7.97	—	8.10	—	8.22	ns
		MachXO2-4000ZE	—	7.98	—	8.10	—	8.23	ns
		MachXO2-7000ZE	—	8.02	—	8.14	—	8.26	ns
t_{SUPLL}	Clock to Data Setup - PIO Input Register	MachXO2-1200ZE	0.85	—	0.85	—	0.89	—	ns
		MachXO2-2000ZE	0.84	—	0.84	—	0.86	—	ns
		MachXO2-4000ZE	0.84	—	0.84	—	0.85	—	ns
		MachXO2-7000ZE	0.83	—	0.83	—	0.81	—	ns

Parameter	Description	Device	-3		-2		-1		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{HPLL}	Clock to Data Hold - PIO Input Register	MachXO2-1200ZE	0.66	—	0.68	—	0.80	—	ns
		MachXO2-2000ZE	0.68	—	0.70	—	0.83	—	ns
		MachXO2-4000ZE	0.68	—	0.71	—	0.84	—	ns
		MachXO2-7000ZE	0.73	—	0.74	—	0.87	—	ns
t_{SU_DELPLL}	Clock to Data Setup - PIO Input Register with Data Input Delay	MachXO2-1200ZE	5.14	—	5.69	—	6.20	—	ns
		MachXO2-2000ZE	5.11	—	5.67	—	6.17	—	ns
		MachXO2-4000ZE	5.27	—	5.84	—	6.35	—	ns
		MachXO2-7000ZE	5.15	—	5.71	—	6.23	—	ns
t_{H_DELPLL}	Clock to Data Hold - PIO Input Register with Input Data Delay	MachXO2-1200ZE	-1.36	—	-1.36	—	-1.36	—	ns
		MachXO2-2000ZE	-1.35	—	-1.35	—	-1.35	—	ns
		MachXO2-4000ZE	-1.43	—	-1.43	—	-1.43	—	ns
		MachXO2-7000ZE	-1.41	—	-1.41	—	-1.41	—	ns
Generic DDRX1 Inputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX1_RX.SCLK.Aligned[9]									
t_{DVA}	Input Data Valid After CLK	All MachXO2 devices, all sides	—	0.382	—	0.401	—	0.417	UI
t_{DVE}	Input Data Hold After CLK		0.670	—	0.684	—	0.693	—	UI
f_{DATA}	DDRX1 Input Data Speed		—	140	—	116	—	98	Mbps
f_{DDRX1}	DDRX1 SCLK Frequency		—	70	—	58	—	49	MHz
Generic DDRX1 Inputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX1_RX.SCLK.Centered[9]									
t_{SU}	Input Data Setup Before CLK	All MachXO2 devices, all sides	1.319	—	1.412	—	1.462	—	ns
t_{HO}	Input Data Hold After CLK		0.717	—	1.010	—	1.340	—	ns
f_{DATA}	DDRX1 Input Data Speed		—	140	—	116	—	98	Mbps
f_{DDRX1}	DDRX1 SCLK Frequency		—	70	—	58	—	49	MHz
Generic DDRX2 Inputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX2_RX.ECLK.Aligned[9]									
t_{DVA}	Input Data Valid After CLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	—	0.361	—	0.346	—	0.334	UI
t_{DVE}	Input Data Hold After CLK		0.602	—	0.625	—	0.648	—	UI
f_{DATA}	DDRX2 Serial Input Data Speed		—	280	—	234	—	194	Mbps
f_{DDRX2}	DDRX2 ECLK Frequency		—	140	—	117	—	97	MHz
f_{SCLK}	SCLK Frequency		—	70	—	59	—	49	MHz
Generic DDRX2 Inputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX2_RX.ECLK.Centered[9]									
t_{SU}	Input Data Setup Before CLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	0.472	—	0.672	—	0.865	—	ns
t_{HO}	Input Data Hold After CLK		0.363	—	0.501	—	0.743	—	ns
f_{DATA}	DDRX2 Serial Input Data Speed		—	280	—	234	—	194	Mbps
f_{DDRX2}	DDRX2 ECLK Frequency		—	140	—	117	—	97	MHz
f_{SCLK}	SCLK Frequency		—	70	—	59	—	49	MHz
Generic DDR4 Inputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input - GDDRX4_RX.ECLK.Aligned[9]									
t_{DVA}	Input Data Valid After ECLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	—	0.307	—	0.316	—	0.326	UI
t_{DVE}	Input Data Hold After ECLK		0.662	—	0.650	—	0.649	—	UI
f_{DATA}	DDRX4 Serial Input Data Speed		—	420	—	352	—	292	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency		—	210	—	176	—	146	MHz
f_{SCLK}	SCLK Frequency		—	53	—	44	—	37	MHz

Parameter	Description	Device	-3		-2		-1		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
Generic DDR4 Inputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX4_RX.ECLK.Centered[9]									
t_{SU}	Input Data Setup Before ECLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	0.434	—	0.535	—	0.630	—	ns
t_{HO}	Input Data Hold After ECLK		0.385	—	0.395	—	0.463	—	ns
f_{DATA}	DDRX4 Serial Input Data Speed		—	420	—	352	—	292	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency		—	210	—	176	—	146	MHz
f_{SCLK}	SCLK Frequency		—	53	—	44	—	37	MHz
7:1 LVDS Inputs – GDDR71_RX.ECLK.7.1[9]									
t_{DVA}	Input Data Valid After ECLK	MachXO2-640U, MachXO2-1200/U and larger devices, bottom side only	—	0.307	—	0.316	—	0.326	UI
t_{DVE}	Input Data Hold After ECLK		0.662	—	0.650	—	0.649	—	UI
f_{DATA}	DDR71 Serial Input Data Speed		—	420	—	352	—	292	Mbps
f_{DDR71}	DDR71 ECLK Frequency		—	210	—	176	—	146	MHz
f_{CLKIN}	7:1 Input Clock Frequency (SCLK) (minimum limited by PLL)		—	60	—	50	—	42	MHz
Generic DDR Outputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX1_TX.SCLK.Aligned[9]									
t_{DIA}	Output Data Invalid After CLK Output	All MachXO2 devices, all sides	—	0.850	—	0.910	—	0.970	ns
t_{DIB}	Output Data Invalid Before CLK Output		—	0.850	—	0.910	—	0.970	ns
f_{DATA}	DDRX1 Output Data Speed		—	140	—	116	—	98	Mbps
f_{DDRX1}	DDRX1 SCLK frequency		—	70	—	58	—	49	MHz
Generic DDR Outputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX1_TX.SCLK.Centered[9]									
t_{DVB}	Output Data Valid Before CLK Output	All MachXO2 devices, all sides	2.720	—	3.380	—	4.140	—	ns
t_{DVA}	Output Data Valid After CLK Output		2.720	—	3.380	—	4.140	—	ns
f_{DATA}	DDRX1 Output Data Speed		—	140	—	116	—	98	Mbps
f_{DDRX1}	DDRX1 SCLK Frequency (minimum limited by PLL)		—	70	—	58	—	49	MHz
Generic DDRX2 Outputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX2_TX.ECLK.Aligned[9]									
t_{DIA}	Output Data Invalid After CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	—	0.270	—	0.300	—	0.330	ns
t_{DIB}	Output Data Invalid Before CLK Output		—	0.270	—	0.300	—	0.330	ns
f_{DATA}	DDRX2 Serial Output Data Speed		—	280	—	234	—	194	Mbps
f_{DDRX2}	DDRX2 ECLK frequency		—	140	—	117	—	97	MHz
f_{SCLK}	SCLK Frequency		—	70	—	59	—	49	MHz

Parameter	Description	Device	-3		-2		-1		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
Generic DDRX2 Outputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX2_TX.ECLK.Centered[9]									
t_{DVB}	Output Data Valid Before CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	1.445	—	1.760	—	2.140	—	ns
t_{DVA}	Output Data Valid After CLK Output		1.445	—	1.760	—	2.140	—	ns
f_{DATA}	DDRX2 Serial Output Data Speed		—	280	—	234	—	194	Mbps
f_{DDRX2}	DDRX2 ECLK Frequency (minimum limited by PLL)		—	140	—	117	—	97	MHz
f_{SCLK}	SCLK Frequency		—	70	—	59	—	49	MHz
Generic DDRX4 Outputs with Clock and Data Aligned at Pin Using PCLK Pin for Clock Input – GDDRX4_TX.ECLK.Aligned[9]									
t_{DIA}	Output Data Invalid After CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	—	0.270	—	0.300	—	0.330	ns
t_{DIB}	Output Data Invalid Before CLK Output		—	0.270	—	0.300	—	0.330	ns
f_{DATA}	DDRX4 Serial Output Data Speed		—	420	—	352	—	292	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency		—	210	—	176	—	146	MHz
f_{SCLK}	SCLK Frequency		—	53	—	44	—	37	MHz
Generic DDRX4 Outputs with Clock and Data Centered at Pin Using PCLK Pin for Clock Input – GDDRX4_TX.ECLK.Centered[9]									
t_{DVB}	Output Data Valid Before CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only	0.873	—	1.067	—	1.319	—	ns
t_{DVA}	Output Data Valid After CLK Output		0.873	—	1.067	—	1.319	—	ns
f_{DATA}	DDRX4 Serial Output Data Speed		—	420	—	352	—	292	Mbps
f_{DDRX4}	DDRX4 ECLK Frequency (minimum limited by PLL)		—	210	—	176	—	146	MHz
f_{SCLK}	SCLK Frequency		—	53	—	44	—	37	MHz
7:1 LVDS Outputs – GDDR71_TX.ECLK.7:1[9]									
t_{DVB}	Output Data Valid Before CLK Output	MachXO2-640U, MachXO2-1200/U and larger devices, top side only.	—	0.240	—	0.270	—	0.300	ns
t_{DVA}	Output Data Valid After CLK Output		—	0.240	—	0.270	—	0.300	ns
f_{DATA}	DDR71 Serial Output Data Speed		—	420	—	352	—	292	Mbps
f_{DDR71}	DDR71 ECLK Frequency		—	210	—	176	—	146	MHz
f_{CLKOUT}	7:1 Output Clock Frequency (SCLK) (minimum limited by PLL)		—	60	—	50	—	42	MHz

Parameter	Description	Device	-3		-2		-1		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
LPDDR[9]									
t_{DVADQ}	Input Data Valid After DQS Input	MachXO2-1200/U and larger devices, right side only.	—	0.349	—	0.381	—	0.396	UI
t_{DVEDQ}	Input Data Hold After DQS Input		0.665	—	0.630	—	0.613	—	UI
t_{DQVBS}	Output Data Invalid Before DQS Output		0.25	—	0.25	—	0.25	—	UI
t_{DQVAS}	Output Data Invalid After DQS Output		0.25	—	0.25	—	0.25	—	UI
f_{DATA}	MEM LPDDR Serial Data Speed		—	120	—	110	—	96	Mbps
f_{SCLK}	SCLK Frequency		—	60	—	55	—	48	MHz
f_{LPDDR}	LPDDR Data Transfer Rate		0	120	0	110	0	96	Mbps
DDR[9]									
t_{DVADQ}	Input Data Valid After DQS Input	MachXO2-1200/U and larger devices, right side only.	—	0.347	—	0.374	—	0.393	UI
t_{DVEDQ}	Input Data Hold After DQS Input		0.665	—	0.637	—	0.616	—	UI
t_{DQVBS}	Output Data Invalid Before DQS Output		0.25	—	0.25	—	0.25	—	UI
t_{DQVAS}	Output Data Invalid After DQS Output		0.25	—	0.25	—	0.25	—	UI
f_{DATA}	MEM DDR Serial Data Speed		—	140	—	116	—	98	Mbps
f_{SCLK}	SCLK Frequency		—	70	—	58	—	49	MHz
f_{MEM_DDR}	MEM DDR Data Transfer Rate		N/A	140	N/A	116	N/A	98	Mbps
DDR2[9]									
t_{DVADQ}	Input Data Valid After DQS Input	MachXO2-1200/U and larger devices, right side only.	—	0.372	—	0.394	—	0.410	UI
t_{DVEDQ}	Input Data Hold After DQS Input		0.690	—	0.658	—	0.618	—	UI
t_{DQVBS}	Output Data Invalid Before DQS Output		0.25	—	0.25	—	0.25	—	UI
t_{DQVAS}	Output Data Invalid After DQS Output		0.25	—	0.25	—	0.25	—	UI
f_{DATA}	MEM DDR Serial Data Speed		—	140	—	116	—	98	Mbps
f_{SCLK}	SCLK Frequency		—	70	—	58	—	49	MHz
f_{MEM_DDR2}	MEM DDR2 Data Transfer Rate		N/A	140	N/A	116	N/A	98	Mbps

1. Exact performance may vary with device and design implementation. Commercial timing numbers are shown at 85°C and 1.14V. Other operating conditions, including industrial, can be extracted from the Diamond software.
2. General I/O timing numbers based on LVCMOS 2.5, 8mA, 0pf load.
3. Generic DDR timing numbers based on LVDS I/O (for input, output, and clock ports).
4. DDR timing numbers based on SSTL25. DDR2 timing numbers based on SSTL18. LPDDR timing numbers based in LVCMOS18.
5. 7:1 LVDS (GDDR71) uses the LVDS I/O standard (for input, output, and clock ports).
6. For Generic DDRX1 mode $t_{SU} = t_{HO} = (t_{DVE} - t_{DVA} - 0.03ns)/2$.
7. The t_{SU_DEL} and t_{H_DEL} values use the SCLK_ZERHOLD default step size. Each step is 167ps (-3), 182ps (-2), 195ps (-1).
8. This number for general purpose usage. Duty cycle tolerance is +/-10%.
9. Duty cycle is +/- 5% for system usage.
10. The above timing numbers are generated using the Diamond design tool. Exact performance may vary with the device selected.

Figure 3-5. Receiver RX.CLK.Aligned and MEM DDR Input Waveforms

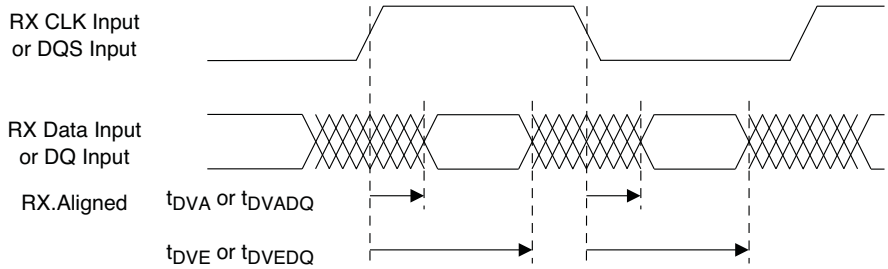


Figure 3-6. Receiver RX.CLK.Centered Waveforms

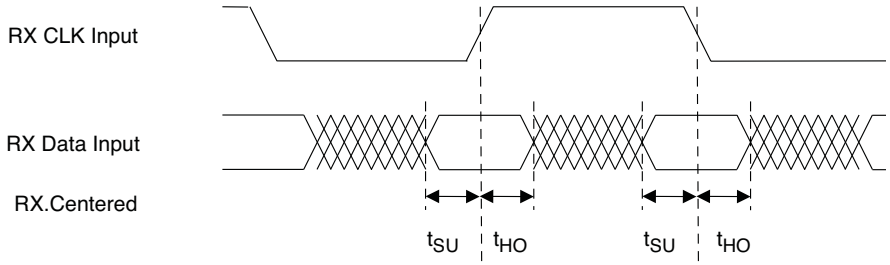


Figure 3-7. Transmitter TX.CLK.Aligned Waveforms

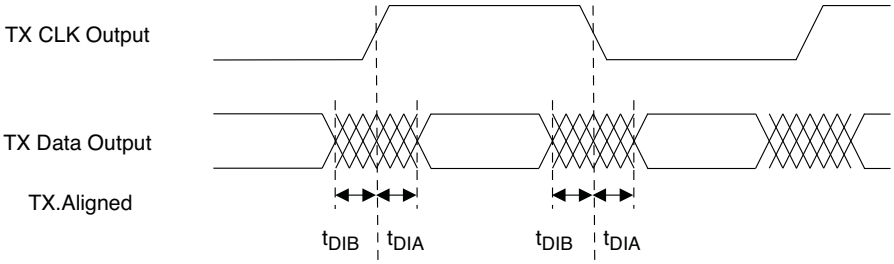


Figure 3-8. Transmitter TX.CLK.Centered and MEM DDR Output Waveforms

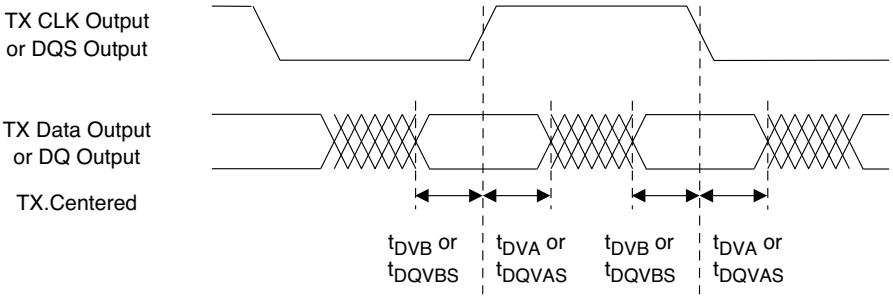


Figure 3-9. GDDR71 Video Timing Waveforms

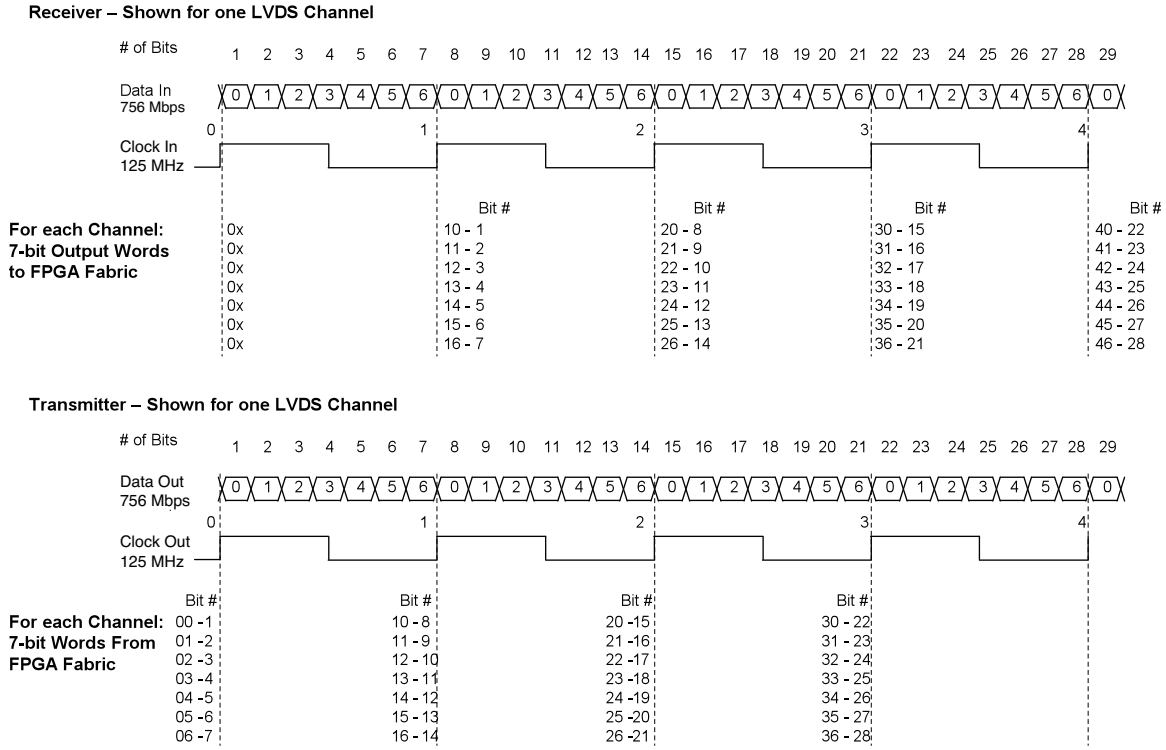


Figure 3-10. Receiver GDDR71_RX. Waveforms

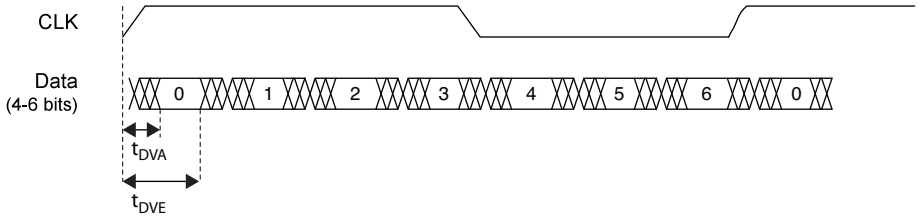
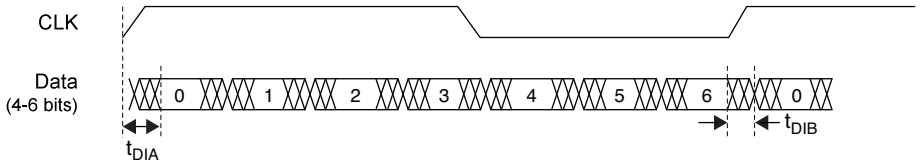


Figure 3-11. Transmitter GDDR71_TX. Waveforms



sysCLOCK PLL Timing

Over Recommended Operating Conditions

Parameter	Descriptions	Conditions	Min.	Max.	Units
f_{IN}	Input Clock Frequency (CLKI, CLKFB)		7	400	MHz
f_{OUT}	Output Clock Frequency (CLKOP, CLKOS, CLKOS2)		1.5625	400	MHz
f_{OUT2}	Output Frequency (CLKOS3 cascaded from CLKOS2)		0.0122	400	MHz
f_{VCO}	PLL VCO Frequency		200	800	MHz
f_{PFD}	Phase Detector Input Frequency		7	400	MHz
AC Characteristics					
t_{DT}	Output Clock Duty Cycle	Without duty trim selected[3]	45	55	%
t_{DT_TRIM}[7]	Edge Duty Trim Accuracy		-75	75	%
t_{PH}[4]	Output Phase Accuracy		-6	6	%
t_{OPJIT}[1, 8]	Output Clock Period Jitter	f_{OUT} > 100MHz	—	150	ps p-p
		f_{OUT} < 100MHz	—	0.007	UIPP
	Output Clock Cycle-to-cycle Jitter	f_{OUT} > 100MHz	—	180	ps p-p
		f_{OUT} < 100MHz	—	0.009	UIPP
	Output Clock Phase Jitter	f_{PFD} > 100MHz	—	160	ps p-p
		f_{PFD} < 100MHz	—	0.011	UIPP
	Output Clock Period Jitter (Fractional-N)	f_{OUT} > 100MHz	—	230	ps p-p
		f_{OUT} < 100MHz	—	0.12	UIPP
	Output Clock Cycle-to-cycle Jitter (Fractional-N)	f_{OUT} > 100MHz	—	230	ps p-p
		f_{OUT} < 100MHz	—	0.12	UIPP
t_{SPO}	Static Phase Offset	Divider ratio = integer	-120	120	ps
t_{W}	Output Clock Pulse Width	At 90% or 10%[3]	0.9	—	ns
t_{LOCK}[2, 5]	PLL Lock-in Time		—	15	ms
t_{UNLOCK}	PLL Unlock Time		—	50	ns
t_{IPJIT}[6]	Input Clock Period Jitter	$f_{PFD} \geq$ 20 MHz	—	1,000	ps p-p
		f_{PFD} < 20 MHz	—	0.02	UIPP
t_{HI}	Input Clock High Time	90% to 90%	0.5	—	ns
t_{LO}	Input Clock Low Time	10% to 10%	0.5	—	ns
t_{STABLE}[5]	STANDBY High to PLL Stable		—	15	ms
t_{RST}	RST/RESETM Pulse Width		1	—	ns
t_{RSTREC}	RST Recovery Time		1	—	ns
t_{RST_DIV}	RESETC/D Pulse Width		10	—	ns
t_{RSTREC_DIV}	RESETC/D Recovery Time		1	—	ns
$t_{ROTATE-SETUP}$	PHASESTEP Setup Time		10	—	ns

sysCLOCK PLL Timing (Continued)

Over Recommended Operating Conditions

Parameter	Descriptions	Conditions	Min.	Max.	Units
t_{ROTATE_WD}	PHASESTEP Pulse Width		4	—	VCO Cycles

1. Period jitter sample is taken over 10,000 samples of the primary PLL output with a clean reference clock. Cycle-to-cycle jitter is taken over 1000 cycles. Phase jitter is taken over 2000 cycles. All values per JESD65B.
2. Output clock is valid after t_{LOCK} for PLL reset and dynamic delay adjustment.
3. Using LVDS output buffers.
4. CLKOS as compared to CLKOP output for one phase step at the maximum VCO frequency. See TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide for more details.
5. At minimum $f_{PFD.}$ As the f_{PFD} increases the time will decrease to approximately 60% the value listed.
6. Maximum allowed jitter on an input clock. PLL unlock may occur if the input jitter exceeds this specification. Jitter on the input clock may be transferred to the output clocks, resulting in jitter measurements outside the output specifications listed in this table.
7. Edge Duty Trim Accuracy is a percentage of the setting value. Settings available are 70 ps, 140 ps, and 280 ps in addition to the default value of none.
8. Jitter values measured with the internal oscillator operating. The jitter values will increase with loading of the PLD fabric and in the presence of SSO noise.

MachXO2 Oscillator Output Frequency

Symbol	Parameter	Min.	Typ.	Max	Units
f_{MAX}	Oscillator Output Frequency (Commercial Grade Devices, 0 to 85°C)	125.685	133	140.315	MHz
	Oscillator Output Frequency (Industrial Grade Devices, -40 to 100°C)	124.355	133	141.645	MHz
t_{DT}	Output Clock Duty Cycle	43	50	57	%
t_{OPJIT}[1]	Output Clock Period Jitter	0.01	0.012	0.02	UIPP
$t_{STABLEOSC}$	STDBY Low to Oscillator Stable	0.01	0.05	0.1	µs

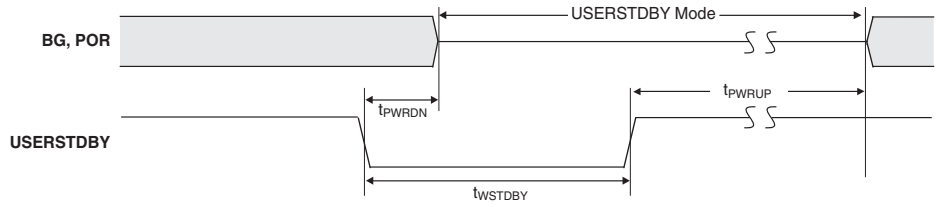
1. Output Clock Period Jitter specified at 133MHz. The values for lower frequencies will be smaller UIPP. The typical value for 133MHz is 95ps and for 2.08MHz the typical value is 1.54ns.

MachXO2 Standby Mode Timing – ZE Devices

Symbol	Parameter	Device	Min.	Typ.	Max	Units
t_{PWRDN}	USERSTDBY High to Stop	All	—	—	13	ns
t_{PWRUP}	USERSTDBY Low to Power Up	LCMXO2-256		—		µs
		LCMXO2-640		—		µs
		LCMXO2-1200	20	—	50	µs
		LCMXO2-2000		—		µs
		LCMXO2-4000		—		µs
		LCMXO2-7000		—		µs
t_{WSTDBY}	USERSTDBY Pulse Width	All	19	—	—	ns
$t_{BNDGAPSTBL}$	USERSTDBY High to Bandgap Stable	All	—	—	15	ns

MachXO2 Standby Mode Timing – HC/HE Devices

Symbol	Parameter	Device	Min.	Typ.	Max	Units
t_{PWRDN}	USERSTDBY High to Stop	All	—	—	9	ns
t_{PWRUP}	USERSTDBY Low to Power Up	LCMXO2-256		—		µs
		LCMXO2-640		—		µs
		LCMXO2-640U		—		µs
		LCMXO2-1200	20	—	50	µs
		LCMXO2-1200U		—		µs
		LCMXO2-2000		—		µs
		LCMXO2-2000U		—		µs
		LCMXO2-4000		—		µs
		LCMXO2-7000		—		µs
t_{WSTDBY}	USERSTDBY Pulse Width	All	18	—	—	ns



Flash Download Time[1, 2]

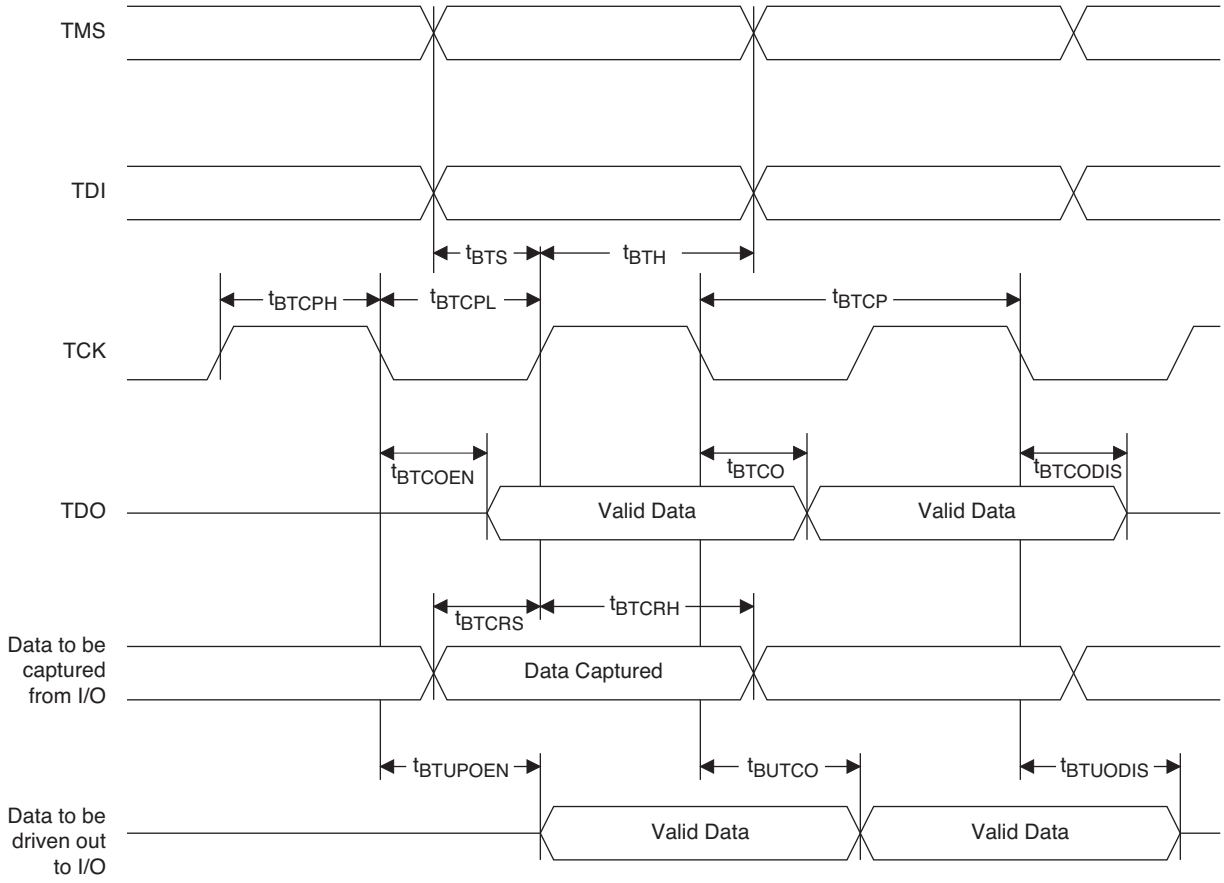
Symbol	Parameter	Device	Typ.	Units
$t_{REFRESH}$	POR to Device I/O Active	LCMXO2-256	0.6	ms
		LCMXO2-640	1.0	ms
		LCMXO2-640U	1.9	ms
		LCMXO2-1200	1.9	ms
		LCMXO2-1200U	1.4	ms
		LCMXO2-2000	1.4	ms
		LCMXO2-2000U	2.4	ms
		LCMXO2-4000	2.4	ms
		LCMXO2-7000	3.8	ms

1. Assumes sysMEM EBR initialized to an all zero pattern if they are used.
2. The Flash download time is measured starting from the maximum voltage of POR trip point.

JTAG Port Timing Specifications

Symbol	Parameter	Min.	Max.	Units
f_{MAX}	TCK clock frequency	—	25	MHz
t_{BTCPH}	TCK [BSCAN] clock pulse width high	20	—	ns
t_{BTCPL}	TCK [BSCAN] clock pulse width low	20	—	ns
t_{BTS}	TCK [BSCAN] setup time	10	—	ns
t_{BTH}	TCK [BSCAN] hold time	8	—	ns
t_{BTCO}	TAP controller falling edge of clock to valid output	—	10	ns
$t_{BTCODIS}$	TAP controller falling edge of clock to valid disable	—	10	ns
t_{BTCOEN}	TAP controller falling edge of clock to valid enable	—	10	ns
t_{BTCRS}	BSCAN test capture register setup time	8	—	ns
t_{BTCRH}	BSCAN test capture register hold time	20	—	ns
t_{BUTCO}	BSCAN test update register, falling edge of clock to valid output	—	25	ns
$t_{BTUODIS}$	BSCAN test update register, falling edge of clock to valid disable	—	25	ns
$t_{BTUPOEN}$	BSCAN test update register, falling edge of clock to valid enable	—	25	ns

Figure 3-12. JTAG Port Timing Waveforms



sysCONFIG Port Timing Specifications

Symbol	Parameter	Min.	Max.	Units
All Configuration Modes				
t_{PRGM}	PROGRAMN low pulse accept	55	—	ns
t_{PRGMJ}	PROGRAMN low pulse rejection	—	25	ns
t_{INITL}	INITN low time	—	55	us
$t_{DPPINIT}$	PROGRAMN low to INITN low	—	70	ns
$t_{DPPDONE}$	PROGRAMN low to DONE low	—	80	ns
t_{IODISS}	PROGRAMN low to I/O disable	—	120	ns
Slave SPI				
f_{MAX}	CCLK clock frequency	—	66	MHz
t_{CCLKH}	CCLK clock pulse width high	7.5	—	ns
t_{CCLKL}	CCLK clock pulse width low	7.5	—	ns
t_{STSU}	CCLK setup time	2	—	ns
t_{STH}	CCLK hold time	0	—	ns
t_{STCO}	CCLK falling edge to valid output	—	10	ns
t_{STOZ}	CCLK falling edge to valid disable	—	10	ns
t_{STOV}	CCLK falling edge to valid enable	—	10	ns
t_{SCS}	Chip select high time	25	—	ns
t_{SCSS}	Chip select setup time	3	—	ns
t_{SCSH}	Chip select hold time	3	—	ns
Master SPI				
f_{MAX}	MCLK clock frequency	—	133	MHz
t_{MCLKH}	MCLK clock pulse width high	3.75	—	ns
t_{MCLKL}	MCLK clock pulse width low	3.75	—	ns
t_{STSU}	MCLK setup time	5	—	ns
t_{STH}	MCLK hold time	1	—	ns
t_{CSSPI}	INITN high to chip select low	100	200	ns
t_{MCLK}	INITN high to first MCLK edge	0.75	1	us

I²C Port Timing Specifications[1, 2]

Symbol	Parameter	Min.	Max.	Units
f_{MAX}	Maximum SCL clock frequency	—	400	KHz

1. MachXO2 supports the following modes:
   - Standard-mode (Sm), with a bit rate up to 100 kbit/s (user and configuration mode)
   - Fast-mode (Fm), with a bit rate up to 400 kbit/s (user and configuration mode)
2. Refer to the I²C specification for timing requirements.

SPI Port Timing Specifications[1]

Symbol	Parameter	Min.	Max.	Units
f_{MAX}	Maximum SCK clock frequency	—	45	MHz

1. Applies to user mode only. For configuration mode timing specifications, refer to sysCONFIG Port Timing Specifications table in this data sheet.

Switching Test Conditions

Figure 3-13 shows the output test load used for AC testing. The specific values for resistance, capacitance, voltage, and other test conditions are shown in Table 3-5.

Figure 3-13. Output Test Load, LVTTL and LVCMOS Standards

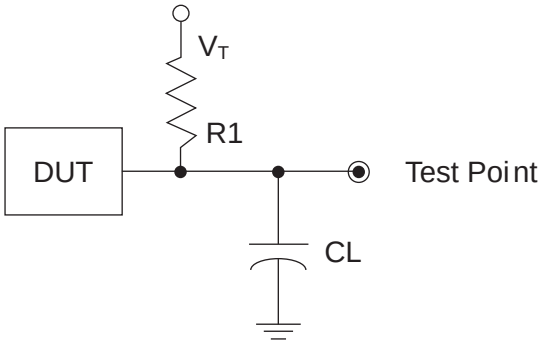


Table 3-5. Test Fixture Required Components, Non-Terminated Interfaces

Test Condition	R1	CL	Timing Ref.	VT
LVTTL and LVCMOS settings (L -> H, H -> L)	∞	0pF	LVTTL, LVCMOS 3.3 = 1.5V	—
			LVCMOS 2.5 = $V_{CCIO}/2$	—
			LVCMOS 1.8 = $V_{CCIO}/2$	—
			LVCMOS 1.5 = $V_{CCIO}/2$	—
			LVCMOS 1.2 = $V_{CCIO}/2$	—
LVTTL and LVCMOS 3.3 (Z -> H)	188	0pF	1.5	V_{OL}
LVTTL and LVCMOS 3.3 (Z -> L)			1.5	V_{OH}
Other LVCMOS (Z -> H)			$V_{CCIO}/2$	V_{OL}
Other LVCMOS (Z -> L)			$V_{CCIO}/2$	V_{OH}
LVTTL + LVCMOS (H -> Z)			V_{OH} - 0.15	V_{OL}
LVTTL + LVCMOS (L -> Z)			V_{OL} - 0.15	V_{OH}

Note: Output test conditions for all other interfaces are determined by the respective standards.

Signal Descriptions

Signal Name	I/O	Descriptions
General Purpose		
P[Edge] [Row/Column Number]_[A/B/C/D]	I/O	[Edge] indicates the edge of the device on which the pad is located. Valid edge designations are L (Left), B (Bottom), R (Right), T (Top). [Row/Column Number] indicates the PFU row or the column of the device on which the PIO Group exists. When Edge is T (Top) or (Bottom), only need to specify Row Number. When Edge is L (Left) or R (Right), only need to specify Column Number. [A/B/C/D] indicates the PIO within the group to which the pad is connected. Some of these user-programmable pins are shared with special function pins. When not used as special function pins, these pins can be programmed as I/Os for user logic. During configuration of the user-programmable I/Os, the user has an option to tri-state the I/Os and enable an internal pull-up, pull-down or buskeeper resistor. This option also applies to unused pins (or those not bonded to a package pin). The default during configuration is for user-programmable I/Os to be tri-stated with an internal pull-down resistor enabled. When the device is erased, I/Os will be tri-stated with an internal pull-down resistor enabled. Some pins, such as PROGRAMN and JTAG pins, default to tri-stated I/Os with pull-up resistors enabled when the device is erased.
NC	—	No connect.
GND	—	GND – Ground. Dedicated pins. It is recommended that all GNDs are tied together.
VCC	—	V_{CC} – The power supply pins for core logic. Dedicated pins. It is recommended that all VCCs are tied to the same supply.
VCCIOx	—	VCCIO – The power supply pins for I/O Bank x. Dedicated pins. It is recommended that all VCCIOs located in the same bank are tied to the same supply.
PLL and Clock Functions (Used as user-programmable I/O pins when not used for PLL or clock pins)		
[LOC]_GPLL[T, C]_IN	—	Reference Clock (PLL) input pads: [LOC] indicates location. Valid designations are L (Left PLL) and R (Right PLL). T = true and C = complement.
[LOC]_GPLL[T, C]_FB	—	Optional Feedback (PLL) input pads: [LOC] indicates location. Valid designations are L (Left PLL) and R (Right PLL). T = true and C = complement.
PCLK [n]_[2:0]	—	Primary Clock pads. One to three clock pads per side.
Test and Programming (Dual function pins used for test access port and during sysCONFIG™)		
TMS	I	Test Mode Select input pin, used to control the 1149.1 state machine.
TCK	I	Test Clock input pin, used to clock the 1149.1 state machine.
TDI	I	Test Data input pin, used to load data into the device using an 1149.1 state machine.
TDO	O	Output pin – Test Data output pin used to shift data out of the device using 1149.1.
JTAGENB	I	Optionally controls behavior of TDI, TDO, TMS, TCK. If the device is configured to use the JTAG pins (TDI, TDO, TMS, TCK) as general purpose I/O, then: If JTAGENB is low: TDI, TDO, TMS and TCK can function a general purpose I/O. If JTAGENB is high: TDI, TDO, TMS and TCK function as JTAG pins. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.
Configuration (Dual function pins used during sysCONFIG)		
PROGRAMN	I	Initiates configuration sequence when asserted low. This pin always has an active pull-up.
INITN	I/O	Open Drain pin. Indicates the FPGA is ready to be configured. During configuration, a pull-up is enabled.

Signal Name	I/O	Descriptions
General Purpose		
DONE	I/O	Open Drain pin. Indicates that the configuration sequence is complete, and the start-up sequence is in progress.
MCLK/CCLK	I/O	Input Configuration Clock for configuring an FPGA in Slave SPI mode. Output Configuration Clock for configuring an FPGA in SPI and SPIm configuration modes.
SN	I	Slave SPI active low chip select input.
CSSPIN	I/O	Master SPI active low chip select output.
SI/SISPI	I/O	Slave SPI serial data input and master SPI serial data output.
SO/SPISO	I/O	Slave SPI serial data output and master SPI serial data input.
SCL	I/O	Slave I^2C clock input and master I^2C clock output.
SDA	I/O	Slave I^2C data input and master I^2C data output.

Pin Information Summary

	MachXO2-256				MachXO2-640		MachXO2-640U
	32 QFN[1]	64 ucBGA	100 TQFP	132 csBGA	100 TQFP	132 csBGA	144 TQFP
General Purpose I/O per Bank							
Bank 0	8	9	13	13	18	19	27
Bank 1	2	12	14	14	20	20	26
Bank 2	9	11	14	14	20	20	28
Bank 3	2	12	14	14	20	20	26
Bank 4	0	0	0	0	0	0	0
Bank 5	0	0	0	0	0	0	0
Total General Purpose Single Ended I/O	21	44	55	55	78	79	107
Differential I/O per Bank							
Bank 0	4	5	7	7	9	10	14
Bank 1	1	6	7	7	10	10	13
Bank 2	4	5	7	7	10	10	14
Bank 3	1	6	7	7	10	10	13
Bank 4	0	0	0	0	0	0	0
Bank 5	0	0	0	0	0	0	0
Total General Purpose Differential I/O	10	22	28	28	39	40	54
Dual Function I/O	22	27	29	29	29	29	33
High-speed Differential I/O							
Bank 0	0	0	0	0	0	0	7
Gearboxes							
Number of 7:1 or 8:1 Output Gearbox Available (Bank 0)	0	0	0	0	0	0	7
Number of 7:1 or 8:1 Input Gearbox Available (Bank 2)	0	0	0	0	0	0	7
DQS Groups							
Bank 1	0	0	0	0	0	0	2
VCCIO Pins							
Bank 0	2	2	2	2	2	2	3
Bank 1	1	2	2	2	2	2	3
Bank 2	2	2	2	2	2	2	3
Bank 3	1	2	2	2	2	2	3
Bank 4	0	0	0	0	0	0	0
Bank 5	0	0	0	0	0	0	0
VCC	2	2	2	2	2	2	4
GND	2	8	8	8	8	10	12
NC	0	1	26	58	3	32	8
Total Count of Bonded Pins	31	62	73	73	96	99	135

1. Lattice recommends soldering the central thermal pad onto the top PCB ground for improved thermal resistance.

	MachXO2-1200				MachXO2-1200U
	100 TQFP	132 csBGA	144 TQFP	25 WLCSP	256 ftBGA
General Purpose I/O per Bank					
Bank 0	18	25	27	11	50
Bank 1	21	26	26	0	52
Bank 2	20	28	28	7	52
Bank 3	20	25	26	0	16
Bank 4	0	0	0	0	16
Bank 5	0	0	0	0	20
Total General Purpose Single Ended I/O	79	104	107	18	206
Differential I/O per Bank					
Bank 0	9	13	14	5	25
Bank 1	10	13	13	0	26
Bank 2	10	14	14	2	26
Bank 3	10	12	13	0	8
Bank 4	0	0	0	0	8
Bank 5	0	0	0	0	10
Total General Purpose Differential I/O	39	52	54	7	103
Dual Function I/O	31	33	33	18	33
High-speed Differential I/O					
Bank 0	4	7	7	0	14
Gearboxes					
Number of 7:1 or 8:1 Output Gearbox Available (Bank 0)	4	7	7	0	14
Number of 7:1 or 8:1 Input Gearbox Available (Bank 2)	5	7	7	0	14
DQS Groups					
Bank 1	1	2	2	0	2
VCCIO Pins					
Bank 0	2	3	3	1	4
Bank 1	2	3	3	0	4
Bank 2	2	3	3	1	4
Bank 3	3	3	3	0	1
Bank 4	0	0	0	0	2
Bank 5	0	0	0	0	1
VCC	2	4	4	2	8
GND	8	10	12	2	24
NC	1	1	8	0	1
Total Count of Bonded Pins	98	130	135	24	254

	MachXO2-2000					MachXO2-2000U
	100 TQFP	132 csBGA	144 TQFP	256 caBGA	256 ftBGA	484 ftBGA
General Purpose I/O per Bank						
Bank 0	18	25	27	50	50	70
Bank 1	21	26	28	52	52	68
Bank 2	20	28	28	52	52	72
Bank 3	6	7	8	16	16	24
Bank 4	6	8	10	16	16	16
Bank 5	8	10	10	20	20	28
Total General Purpose Single-Ended I/O	79	104	111	206	206	278
Differential I/O per Bank						
Bank 0	9	13	14	25	25	35
Bank 1	10	13	14	26	26	34
Bank 2	10	14	14	26	26	36
Bank 3	3	3	4	8	8	12
Bank 4	3	4	5	8	8	8
Bank 5	4	5	5	10	10	14
Total General Purpose Differential I/O	39	52	56	103	103	139
Dual Function I/O	31	33	33	33	33	37
High-speed Differential I/O						
Bank 0	4	8	9	14	14	18
Gearboxes						
Number of 7:1 or 8:1 Output Gearbox Available (Bank 0)	4	8	9	14	14	18
Number of 7:1 or 8:1 Input Gearbox Available (Bank 2)	10	14	14	14	14	18
DQS Groups						
Bank 1	1	2	2	2	2	2
VCCIO Pins						
Bank 0	2	3	3	4	4	10
Bank 1	2	3	3	4	4	10
Bank 2	2	3	3	4	4	10
Bank 3	1	1	1	1	1	3
Bank 4	1	1	1	2	2	4
Bank 5	1	1	1	1	1	3
VCC	2	4	4	8	8	12
GND	8	10	12	24	24	48
NC	1	1	4	1	1	105
Total Count of Bonded Pins	98	130	139	254	254	378

	MachXO2-4000						
	132 csBGA	144 TQFP	184 csBGA	256 caBGA	256 ftBGA	332 caBGA	484 fpBGA
General Purpose I/O per Bank							
Bank 0	25	27	37	50	50	68	70
Bank 1	26	29	37	52	52	68	68
Bank 2	28	29	39	52	52	70	72
Bank 3	7	9	10	16	16	24	24
Bank 4	8	10	12	16	16	16	16
Bank 5	10	10	15	20	20	28	28
Total General Purpose Single Ended I/O	104	114	150	206	206	274	278
Differential I/O per Bank							
Bank 0	13	14	18	25	25	34	35
Bank 1	13	14	18	26	26	34	34
Bank 2	14	14	19	26	26	35	36
Bank 3	3	4	4	8	8	12	12
Bank 4	4	5	6	8	8	8	8
Bank 5	5	5	7	10	10	14	14
Total General Purpose Differential I/O	52	56	72	103	103	137	139
Dual Function I/O	37	37	37	37	37	37	37
High-speed Differential I/O							
Bank 0	8	9	8	18	18	18	18
Gearboxes							
Number of 7:1 or 8:1 Output Gearbox Available (Bank 0)	8	9	9	18	18	18	18
Number of 7:1 or 8:1 Input Gearbox Available (Bank 2)	14	14	12	18	18	18	18
DQS Groups							
Bank 1	2	2	2	2	2	2	2
VCCIO Pins							
Bank 0	3	3	3	4	4	4	10
Bank 1	3	3	3	4	4	4	10
Bank 2	3	3	3	4	4	4	10
Bank 3	1	1	1	1	1	2	3
Bank 4	1	1	1	2	2	1	4
Bank 5	1	1	1	1	1	2	3
VCC	4	4	4	8	8	8	12
GND	10	12	16	24	24	27	48
NC	1	1	1	1	1	5	105
Total Count of Bonded Pins	130	142	182	254	254	326	378

	MachXO2-7000				
	144 TQFP	**256 caBGA**	**256 ftBGA**	**332 caBGA**	**484 fpBGA**
General Purpose I/O per Bank					
Bank 0	27	50	50	68	82
Bank 1	29	52	52	70	84
Bank 2	29	52	52	70	84
Bank 3	9	16	16	24	28
Bank 4	10	16	16	16	24
Bank 5	10	20	20	30	32
Total General Purpose Single Ended I/O	114	206	206	278	334
Differential I/O per Bank					
Bank 0	14	25	25	34	41
Bank 1	14	26	26	35	42
Bank 2	14	26	26	35	42
Bank 3	4	8	8	12	14
Bank 4	5	8	8	8	12
Bank 5	5	10	10	15	16
Total General Purpose Differential I/O	56	103	103	139	167
Dual Function I/O	37	37	37	37	37
High-speed Differential I/O					
Bank 0	9	20	20	21	21
Gearboxes					
Number of 7:1 or 8:1 Output Gearbox Available (Bank 0)	9	20	20	21	21
Number of 7:1 or 8:1 Input Gearbox Available (Bank 2)	14	20	20	21	21
DQS Groups					
Bank 1	2	2	2	2	2
VCCIO Pins					
Bank 0	3	4	4	4	10
Bank 1	3	4	4	4	10
Bank 2	3	4	4	4	10
Bank 3	1	1	1	2	3
Bank 4	1	2	2	1	4
Bank 5	1	1	1	2	3
VCC	4	8	8	8	12
GND	12	24	24	27	48
NC	1	1	1	1	49
Total Count of Bonded Pins	142	254	254	330	434

For Further Information

For further information regarding logic signal connections for various packages please refer to the MachXO2 Device Pinout Files.

Thermal Management

Thermal management is recommended as part of any sound FPGA design methodology. To assess the thermal characteristics of a system, Lattice specifies a maximum allowable junction temperature in all device data sheets. Users must complete a thermal analysis of their specific design to ensure that the device and package do not exceed the junction temperature limits. Refer to the Thermal Management document to find the device/package specific thermal values.

For Further Information

For further information regarding Thermal Management, refer to the following:

- Thermal Management document
- TN1198, Power Estimation and Management for MachXO2 Devices
- The Power Calculator tool is included with the Lattice design tools, or as a standalone download from www.latticesemi.com/software

MachXO2 Family Data Sheet Ordering Information

January 2013 Data Sheet DS1035

MachXO2 Part Number Description

LCMXO2 – XXXX X X X – X XXXXXX X XX XX

Device Family
MachXO2 PLD

Logic Capacity
256 = 256 LUTs
640 = 640 LUTs
1200 = 1280 LUTs
2000 = 2112 LUTs
4000 = 4320 LUTs
7000 = 6864 LUTs

I/O Count
Blank = Standard Device
U = Ultra High I/O Device

Power/Performance
Z = Low Power
H = High Performance

Supply Voltage
C = 2.5V/3.3V
E = 1.2V

Speed
1 = Slowest
2
3 = Fastest
} Low Power

4 = Slowest
5
6 = Fastest
} High Performance

Device Status
Blank = Production Device
ES = Engineering Sample
R1 = Production Release 1 Device
50 = WLCSP package, 50 parts per reel

Shipping Method
Blank = Trays
TR = Tape and Reel

Grade
C = Commercial
I = Industrial

Package
UWG25 = 25-Ball Halogen-Free WLCSP (0.4 mm Pitch)
SG32 = 32-Pin Halogen-Free QFN (0.5 mm Pitch)
UMG64 = 64-Ball Halogen-Free ucBGA (0.4 mm Pitch)
TG100 = 100-Pin Halogen-Free TQFP
TG144 = 144-Pin Halogen-Free TQFP
MG132 = 132-Ball Halogen-Free csBGA (0.5 mm Pitch)
MG184 = 184-Ball Halogen-Free csBGA (0.5mm Pitch)
BG256 = 256-Ball Halogen-Free caBGA (0.8 mm Pitch)
FTG256= 256-Ball Halogen-Free ftBGA (1.0 mm Pitch)
BG332 = 332-Ball Halogen-Free caBGA
FG484 = 484-Ball Halogen-Free fpBGA (1.0 mm Pitch)

Ordering Information

MachXO2 devices have top-side markings, for commercial and industrial grades, as shown below:

Notes:

1. Markings are abbreviated for small packages.

2. See PCN 05A-12 for information regarding a change to the top-side mark logo.

Ultra Low Power Commercial Grade Devices, Halogen Free (RoHS) Packaging

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-256ZE-1SG32C	256	1.2V	-1	Halogen-Free QFN	32	COM
LCMXO2-256ZE-2SG32C	256	1.2V	-2	Halogen-Free QFN	32	COM
LCMXO2-256ZE-3SG32C	256	1.2V	-3	Halogen-Free QFN	32	COM
LCMXO2-256ZE-1UMG64C	256	1.2V	-1	Halogen-Free ucBGA	64	COM
LCMXO2-256ZE-2UMG64C	256	1.2V	-2	Halogen-Free ucBGA	64	COM
LCMXO2-256ZE-3UMG64C	256	1.2V	-3	Halogen-Free ucBGA	64	COM
LCMXO2-256ZE-1TG100C	256	1.2V	-1	Halogen-Free TQFP	100	COM
LCMXO2-256ZE-2TG100C	256	1.2V	-2	Halogen-Free TQFP	100	COM
LCMXO2-256ZE-3TG100C	256	1.2V	-3	Halogen-Free TQFP	100	COM
LCMXO2-256ZE-1MG132C	256	1.2V	-1	Halogen-Free csBGA	132	COM
LCMXO2-256ZE-2MG132C	256	1.2V	-2	Halogen-Free csBGA	132	COM
LCMXO2-256ZE-3MG132C	256	1.2V	-3	Halogen-Free csBGA	132	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-640ZE-1TG100C	640	1.2V	-1	Halogen-Free TQFP	100	COM
LCMXO2-640ZE-2TG100C	640	1.2V	-2	Halogen-Free TQFP	100	COM
LCMXO2-640ZE-3TG100C	640	1.2V	-3	Halogen-Free TQFP	100	COM
LCMXO2-640ZE-1MG132C	640	1.2V	-1	Halogen-Free csBGA	132	COM
LCMXO2-640ZE-2MG132C	640	1.2V	-2	Halogen-Free csBGA	132	COM
LCMXO2-640ZE-3MG132C	640	1.2V	-3	Halogen-Free csBGA	132	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200ZE-1TG100C	1280	1.2V	-1	Halogen-Free TQFP	100	COM
LCMXO2-1200ZE-2TG100C	1280	1.2V	-2	Halogen-Free TQFP	100	COM
LCMXO2-1200ZE-3TG100C	1280	1.2V	-3	Halogen-Free TQFP	100	COM
LCMXO2-1200ZE-1MG132C	1280	1.2V	-1	Halogen-Free csBGA	132	COM
LCMXO2-1200ZE-2MG132C	1280	1.2V	-2	Halogen-Free csBGA	132	COM
LCMXO2-1200ZE-3MG132C	1280	1.2V	-3	Halogen-Free csBGA	132	COM
LCMXO2-1200ZE-1TG144C	1280	1.2V	-1	Halogen-Free TQFP	144	COM
LCMXO2-1200ZE-2TG144C	1280	1.2V	-2	Halogen-Free TQFP	144	COM
LCMXO2-1200ZE-3TG144C	1280	1.2V	-3	Halogen-Free TQFP	144	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000ZE-1TG100C	2112	1.2V	-1	Halogen-Free TQFP	100	COM
LCMXO2-2000ZE-2TG100C	2112	1.2V	-2	Halogen-Free TQFP	100	COM
LCMXO2-2000ZE-3TG100C	2112	1.2V	-3	Halogen-Free TQFP	100	COM
LCMXO2-2000ZE-1MG132C	2112	1.2V	-1	Halogen-Free csBGA	132	COM
LCMXO2-2000ZE-2MG132C	2112	1.2V	-2	Halogen-Free csBGA	132	COM
LCMXO2-2000ZE-3MG132C	2112	1.2V	-3	Halogen-Free csBGA	132	COM
LCMXO2-2000ZE-1TG144C	2112	1.2V	-1	Halogen-Free TQFP	144	COM
LCMXO2-2000ZE-2TG144C	2112	1.2V	-2	Halogen-Free TQFP	144	COM
LCMXO2-2000ZE-3TG144C	2112	1.2V	-3	Halogen-Free TQFP	144	COM
LCMXO2-2000ZE-1BG256C	2112	1.2V	-1	Halogen-Free caBGA	256	COM
LCMXO2-2000ZE-2BG256C	2112	1.2V	-2	Halogen-Free caBGA	256	COM
LCMXO2-2000ZE-3BG256C	2112	1.2V	-3	Halogen-Free caBGA	256	COM
LCMXO2-2000ZE-1FTG256C	2112	1.2V	-1	Halogen-Free ftBGA	256	COM
LCMXO2-2000ZE-2FTG256C	2112	1.2V	-2	Halogen-Free ftBGA	256	COM
LCMXO2-2000ZE-3FTG256C	2112	1.2V	-3	Halogen-Free ftBGA	256	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000ZE-1MG132C	4320	1.2V	-1	Halogen-Free csBGA	132	COM
LCMXO2-4000ZE-2MG132C	4320	1.2V	-2	Halogen-Free csBGA	132	COM
LCMXO2-4000ZE-3MG132C	4320	1.2V	-3	Halogen-Free csBGA	132	COM
LCMXO2-4000ZE-1TG144C	4320	1.2V	-1	Halogen-Free TQFP	144	COM
LCMXO2-4000ZE-2TG144C	4320	1.2V	-2	Halogen-Free TQFP	144	COM
LCMXO2-4000ZE-3TG144C	4320	1.2V	-3	Halogen-Free TQFP	144	COM
LCMXO2-4000ZE-1BG256C	4320	1.2V	-1	Halogen-Free caBGA	256	COM
LCMXO2-4000ZE-2BG256C	4320	1.2V	-2	Halogen-Free caBGA	256	COM
LCMXO2-4000ZE-3BG256C	4320	1.2V	-3	Halogen-Free caBGA	256	COM
LCMXO2-4000ZE-1FTG256C	4320	1.2V	-1	Halogen-Free ftBGA	256	COM
LCMXO2-4000ZE-2FTG256C	4320	1.2V	-2	Halogen-Free ftBGA	256	COM
LCMXO2-4000ZE-3FTG256C	4320	1.2V	-3	Halogen-Free ftBGA	256	COM
LCMXO2-4000ZE-1BG332C	4320	1.2V	-1	Halogen-Free caBGA	332	COM
LCMXO2-4000ZE-2BG332C	4320	1.2V	-2	Halogen-Free caBGA	332	COM
LCMXO2-4000ZE-3BG332C	4320	1.2V	-3	Halogen-Free caBGA	332	COM
LCMXO2-4000ZE-1FG484C	4320	1.2V	-1	Halogen-Free fpBGA	484	COM
LCMXO2-4000ZE-2FG484C	4320	1.2V	-2	Halogen-Free fpBGA	484	COM
LCMXO2-4000ZE-3FG484C	4320	1.2V	-3	Halogen-Free fpBGA	484	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-7000ZE-1TG144C	6864	1.2V	-1	Halogen-Free TQFP	144	COM
LCMXO2-7000ZE-2TG144C	6864	1.2V	-2	Halogen-Free TQFP	144	COM
LCMXO2-7000ZE-3TG144C	6864	1.2V	-3	Halogen-Free TQFP	144	COM
LCMXO2-7000ZE-1BG256C	6864	1.2V	-1	Halogen-Free caBGA	256	COM
LCMXO2-7000ZE-2BG256C	6864	1.2V	-2	Halogen-Free caBGA	256	COM
LCMXO2-7000ZE-3BG256C	6864	1.2V	-3	Halogen-Free caBGA	256	COM
LCMXO2-7000ZE-1FTG256C	6864	1.2V	-1	Halogen-Free ftBGA	256	COM
LCMXO2-7000ZE-2FTG256C	6864	1.2V	-2	Halogen-Free ftBGA	256	COM
LCMXO2-7000ZE-3FTG256C	6864	1.2V	-3	Halogen-Free ftBGA	256	COM
LCMXO2-7000ZE-1BG332C	6864	1.2V	-1	Halogen-Free caBGA	332	COM
LCMXO2-7000ZE-2BG332C	6864	1.2V	-2	Halogen-Free caBGA	332	COM
LCMXO2-7000ZE-3BG332C	6864	1.2V	-3	Halogen-Free caBGA	332	COM
LCMXO2-7000ZE-1FG484C	6864	1.2V	-1	Halogen-Free fpBGA	484	COM
LCMXO2-7000ZE-2FG484C	6864	1.2V	-2	Halogen-Free fpBGA	484	COM
LCMXO2-7000ZE-3FG484C	6864	1.2V	-3	Halogen-Free fpBGA	484	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200ZE-1TG100CR1[1]	1280	1.2V	-1	Halogen-Free TQFP	100	COM
LCMXO2-1200ZE-2TG100CR1[1]	1280	1.2V	-2	Halogen-Free TQFP	100	COM
LCMXO2-1200ZE-3TG100CR1[1]	1280	1.2V	-3	Halogen-Free TQFP	100	COM
LCMXO2-1200ZE-1MG132CR1[1]	1280	1.2V	-1	Halogen-Free csBGA	132	COM
LCMXO2-1200ZE-2MG132CR1[1]	1280	1.2V	-2	Halogen-Free csBGA	132	COM
LCMXO2-1200ZE-3MG132CR1[1]	1280	1.2V	-3	Halogen-Free csBGA	132	COM
LCMXO2-1200ZE-1TG144CR1[1]	1280	1.2V	-1	Halogen-Free TQFP	144	COM
LCMXO2-1200ZE-2TG144CR1[1]	1280	1.2V	-2	Halogen-Free TQFP	144	COM
LCMXO2-1200ZE-3TG144CR1[1]	1280	1.2V	-3	Halogen-Free TQFP	144	COM

1. Specifications for the “LCMXO2-1200ZE-speed package CR1” are the same as the “LCMXO2-1200ZE-speed package C” devices respectively, except as specified in the R1 Device Specifications section on page 5-18 of this data sheet.

High-Performance Commercial Grade Devices with Voltage Regulator, Halogen Free (RoHS) Packaging

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-256HC-4SG32C	256	2.5V/3.3V	-4	Halogen-Free QFN	32	COM
LCMXO2-256HC-5SG32C	256	2.5V/3.3V	-5	Halogen-Free QFN	32	COM
LCMXO2-256HC-6SG32C	256	2.5V/3.3V	-6	Halogen-Free QFN	32	COM
LCMXO2-256HC-4UMG64C	256	2.5V/3.3V	-4	Halogen-Free ucBGA	64	COM
LCMXO2-256HC-5UMG64C	256	2.5V/3.3V	-5	Halogen-Free ucBGA	64	COM
LCMXO2-256HC-6UMG64C	256	2.5V/3.3V	-6	Halogen-Free ucBGA	64	COM
LCMXO2-256HC-4TG100C	256	2.5V/3.3V	-4	Halogen-Free TQFP	100	COM
LCMXO2-256HC-5TG100C	256	2.5V/3.3V	-5	Halogen-Free TQFP	100	COM
LCMXO2-256HC-6TG100C	256	2.5V/3.3V	-6	Halogen-Free TQFP	100	COM
LCMXO2-256HC-4MG132C	256	2.5V/3.3V	-4	Halogen-Free csBGA	132	COM
LCMXO2-256HC-5MG132C	256	2.5V/3.3V	-5	Halogen-Free csBGA	132	COM
LCMXO2-256HC-6MG132C	256	2.5V/3.3V	-6	Halogen-Free csBGA	132	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-640HC-4TG100C	640	2.5V/3.3V	-4	Halogen-Free TQFP	100	COM
LCMXO2-640HC-5TG100C	640	2.5V/3.3V	-5	Halogen-Free TQFP	100	COM
LCMXO2-640HC-6TG100C	640	2.5V/3.3V	-6	Halogen-Free TQFP	100	COM
LCMXO2-640HC-4MG132C	640	2.5V/3.3V	-4	Halogen-Free csBGA	132	COM
LCMXO2-640HC-5MG132C	640	2.5V/3.3V	-5	Halogen-Free csBGA	132	COM
LCMXO2-640HC-6MG132C	640	2.5V/3.3V	-6	Halogen-Free csBGA	132	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-640UHC-4TG144C	640	2.5V/3.3V	-4	Halogen-Free TQFP	144	COM
LCMXO2-640UHC-5TG144C	640	2.5V/3.3V	-5	Halogen-Free TQFP	144	COM
LCMXO2-640UHC-6TG144C	640	2.5V/3.3V	-6	Halogen-Free TQFP	144	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200HC-4TG100C	1280	2.5V/3.3V	-4	Halogen-Free TQFP	100	COM
LCMXO2-1200HC-5TG100C	1280	2.5V/3.3V	-5	Halogen-Free TQFP	100	COM
LCMXO2-1200HC-6TG100C	1280	2.5V/3.3V	-6	Halogen-Free TQFP	100	COM
LCMXO2-1200HC-4MG132C	1280	2.5V/3.3V	-4	Halogen-Free csBGA	132	COM
LCMXO2-1200HC-5MG132C	1280	2.5V/3.3V	-5	Halogen-Free csBGA	132	COM
LCMXO2-1200HC-6MG132C	1280	2.5V/3.3V	-6	Halogen-Free csBGA	132	COM
LCMXO2-1200HC-4TG144C	1280	2.5V/3.3V	-4	Halogen-Free TQFP	144	COM
LCMXO2-1200HC-5TG144C	1280	2.5V/3.3V	-5	Halogen-Free TQFP	144	COM
LCMXO2-1200HC-6TG144C	1280	2.5V/3.3V	-6	Halogen-Free TQFP	144	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200UHC-4FTG256C	1280	2.5V/3.3V	-4	Halogen-Free ftBGA	256	COM
LCMXO2-1200UHC-5FTG256C	1280	2.5V/3.3V	-5	Halogen-Free ftBGA	256	COM
LCMXO2-1200UHC-6FTG256C	1280	2.5V/3.3V	-6	Halogen-Free ftBGA	256	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000HC-4TG100C	2112	2.5V/3.3V	-4	Halogen-Free TQFP	100	COM
LCMXO2-2000HC-5TG100C	2112	2.5V/3.3V	-5	Halogen-Free TQFP	100	COM
LCMXO2-2000HC-6TG100C	2112	2.5V/3.3V	-6	Halogen-Free TQFP	100	COM
LCMXO2-2000HC-4MG132C	2112	2.5V/3.3V	-4	Halogen-Free csBGA	132	COM
LCMXO2-2000HC-5MG132C	2112	2.5V/3.3V	-5	Halogen-Free csBGA	132	COM
LCMXO2-2000HC-6MG132C	2112	2.5V/3.3V	-6	Halogen-Free csBGA	132	COM
LCMXO2-2000HC-4TG144C	2112	2.5V/3.3V	-4	Halogen-Free TQFP	144	COM
LCMXO2-2000HC-5TG144C	2112	2.5V/3.3V	-5	Halogen-Free TQFP	144	COM
LCMXO2-2000HC-6TG144C	2112	2.5V/3.3V	-6	Halogen-Free TQFP	144	COM
LCMXO2-2000HC-4BG256C	2112	2.5V/3.3V	-4	Halogen-Free caBGA	256	COM
LCMXO2-2000HC-5BG256C	2112	2.5V/3.3V	-5	Halogen-Free caBGA	256	COM
LCMXO2-2000HC-6BG256C	2112	2.5V/3.3V	-6	Halogen-Free caBGA	256	COM
LCMXO2-2000HC-4FTG256C	2112	2.5V/3.3V	-4	Halogen-Free ftBGA	256	COM
LCMXO2-2000HC-5FTG256C	2112	2.5V/3.3V	-5	Halogen-Free ftBGA	256	COM
LCMXO2-2000HC-6FTG256C	2112	2.5V/3.3V	-6	Halogen-Free ftBGA	256	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000UHC-4FG484C	2112	2.5V/3.3V	-4	Halogen-Free fpBGA	484	COM
LCMXO2-2000UHC-5FG484C	2112	2.5V/3.3V	-5	Halogen-Free fpBGA	484	COM
LCMXO2-2000UHC-6FG484C	2112	2.5V/3.3V	-6	Halogen-Free fpBGA	484	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000HC-4MG132C	4320	2.5V/3.3V	-4	Halogen-Free csBGA	132	COM
LCMXO2-4000HC-5MG132C	4320	2.5V/3.3V	-5	Halogen-Free csBGA	132	COM
LCMXO2-4000HC-6MG132C	4320	2.5V/3.3V	-6	Halogen-Free csBGA	132	COM
LCMXO2-4000HC-4TG144C	4320	2.5V/3.3V	-4	Halogen-Free TQFP	144	COM
LCMXO2-4000HC-5TG144C	4320	2.5V/3.3V	-5	Halogen-Free TQFP	144	COM
LCMXO2-4000HC-6TG144C	4320	2.5V/3.3V	-6	Halogen-Free TQFP	144	COM
LCMXO2-4000HC-4BG256C	4320	2.5V/3.3V	-4	Halogen-Free caBGA	256	COM
LCMXO2-4000HC-5BG256C	4320	2.5V/3.3V	-5	Halogen-Free caBGA	256	COM
LCMXO2-4000HC-6BG256C	4320	2.5V/3.3V	-6	Halogen-Free caBGA	256	COM
LCMXO2-4000HC-4FTG256C	4320	2.5V/3.3V	-4	Halogen-Free ftBGA	256	COM
LCMXO2-4000HC-5FTG256C	4320	2.5V/3.3V	-5	Halogen-Free ftBGA	256	COM
LCMXO2-4000HC-6FTG256C	4320	2.5V/3.3V	-6	Halogen-Free ftBGA	256	COM
LCMXO2-4000HC-4BG332C	4320	2.5V/3.3V	-4	Halogen-Free caBGA	332	COM
LCMXO2-4000HC-5BG332C	4320	2.5V/3.3V	-5	Halogen-Free caBGA	332	COM
LCMXO2-4000HC-6BG332C	4320	2.5V/3.3V	-6	Halogen-Free caBGA	332	COM
LCMXO2-4000HC-4FG484C	4320	2.5V/3.3V	-4	Halogen-Free fpBGA	484	COM
LCMXO2-4000HC-5FG484C	4320	2.5V/3.3V	-5	Halogen-Free fpBGA	484	COM
LCMXO2-4000HC-6FG484C	4320	2.5V/3.3V	-6	Halogen-Free fpBGA	484	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-7000HC-4TG144C	6864	2.5V/3.3V	-4	Halogen-Free TQFP	144	COM
LCMXO2-7000HC-5TG144C	6864	2.5V/3.3V	-5	Halogen-Free TQFP	144	COM
LCMXO2-7000HC-6TG144C	6864	2.5V/3.3V	-6	Halogen-Free TQFP	144	COM
LCMXO2-7000HC-4BG256C	6864	2.5V/3.3V	-4	Halogen-Free caBGA	256	COM
LCMXO2-7000HC-5BG256C	6864	2.5V/3.3V	-5	Halogen-Free caBGA	256	COM
LCMXO2-7000HC-6BG256C	6864	2.5V/3.3V	-6	Halogen-Free caBGA	256	COM
LCMXO2-7000HC-4FTG256C	6864	2.5V/3.3V	-4	Halogen-Free ftBGA	256	COM
LCMXO2-7000HC-5FTG256C	6864	2.5V/3.3V	-5	Halogen-Free ftBGA	256	COM
LCMXO2-7000HC-6FTG256C	6864	2.5V/3.3V	-6	Halogen-Free ftBGA	256	COM
LCMXO2-7000HC-4BG332C	6864	2.5V/3.3V	-4	Halogen-Free caBGA	332	COM
LCMXO2-7000HC-5BG332C	6864	2.5V/3.3V	-5	Halogen-Free caBGA	332	COM
LCMXO2-7000HC-6BG332C	6864	2.5V/3.3V	-6	Halogen-Free caBGA	332	COM
LCMXO2-7000HC-4FG484C	6864	2.5V/3.3V	-4	Halogen-Free fpBGA	484	COM
LCMXO2-7000HC-5FG484C	6864	2.5V/3.3V	-5	Halogen-Free fpBGA	484	COM
LCMXO2-7000HC-6FG484C	6864	2.5V/3.3V	-6	Halogen-Free fpBGA	484	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200HC-4TG100CR1[1]	1280	2.5V/3.3V	-4	Halogen-Free TQFP	100	COM
LCMXO2-1200HC-5TG100CR1[1]	1280	2.5V/3.3V	-5	Halogen-Free TQFP	100	COM
LCMXO2-1200HC-6TG100CR1[1]	1280	2.5V/3.3V	-6	Halogen-Free TQFP	100	COM
LCMXO2-1200HC-4MG132CR1[1]	1280	2.5V/3.3V	-4	Halogen-Free csBGA	132	COM
LCMXO2-1200HC-5MG132CR1[1]	1280	2.5V/3.3V	-5	Halogen-Free csBGA	132	COM
LCMXO2-1200HC-6MG132CR1[1]	1280	2.5V/3.3V	-6	Halogen-Free csBGA	132	COM
LCMXO2-1200HC-4TG144CR1[1]	1280	2.5V/3.3V	-4	Halogen-Free TQFP	144	COM
LCMXO2-1200HC-5TG144CR1[1]	1280	2.5V/3.3V	-5	Halogen-Free TQFP	144	COM
LCMXO2-1200HC-6TG144CR1[1]	1280	2.5V/3.3V	-6	Halogen-Free TQFP	144	COM

1. Specifications for the “LCMXO2-1200HC-speed package CR1” are the same as the “LCMXO2-1200HC-speed package C” devices respectively, except as specified in the R1 Device Specifications section on page 5-18 of this data sheet.

High-Performance Commercial Grade Devices without Voltage Regulator, Halogen Free (RoHS) Packaging

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000HE-4TG100C	2112	1.2V	-4	Halogen-Free TQFP	100	COM
LCMXO2-2000HE-5TG100C	2112	1.2V	-5	Halogen-Free TQFP	100	COM
LCMXO2-2000HE-6TG100C	2112	1.2V	-6	Halogen-Free TQFP	100	COM
LCMXO2-2000HE-4TG144C	2112	1.2V	-4	Halogen-Free TQFP	144	COM
LCMXO2-2000HE-5TG144C	2112	1.2V	-5	Halogen-Free TQFP	144	COM
LCMXO2-2000HE-6TG144C	2112	1.2V	-6	Halogen-Free TQFP	144	COM
LCMXO2-2000HE-4MG132C	2112	1.2V	-4	Halogen-Free csBGA	132	COM
LCMXO2-2000HE-5MG132C	2112	1.2V	-5	Halogen-Free csBGA	132	COM
LCMXO2-2000HE-6MG132C	2112	1.2V	-6	Halogen-Free csBGA	132	COM
LCMXO2-2000HE-4BG256C	2112	1.2V	-4	Halogen-Free caBGA	256	COM
LCMXO2-2000HE-5BG256C	2112	1.2V	-5	Halogen-Free caBGA	256	COM
LCMXO2-2000HE-6BG256C	2112	1.2V	-6	Halogen-Free caBGA	256	COM
LCMXO2-2000HE-4FTG256C	2112	1.2V	-4	Halogen-Free ftBGA	256	COM
LCMXO2-2000HE-5FTG256C	2112	1.2V	-5	Halogen-Free ftBGA	256	COM
LCMXO2-2000HE-6FTG256C	2112	1.2V	-6	Halogen-Free ftBGA	256	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000UHE-4FG484C	2112	1.2V	-4	Halogen-Free fpBGA	484	COM
LCMXO2-2000UHE-5FG484C	2112	1.2V	-5	Halogen-Free fpBGA	484	COM
LCMXO2-2000UHE-6FG484C	2112	1.2V	-6	Halogen-Free fpBGA	484	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000HE-4TG144C	4320	1.2V	-4	Halogen-Free TQFP	144	COM
LCMXO2-4000HE-5TG144C	4320	1.2V	-5	Halogen-Free TQFP	144	COM
LCMXO2-4000HE-6TG144C	4320	1.2V	-6	Halogen-Free TQFP	144	COM
LCMXO2-4000HE-4MG132C	4320	1.2V	-4	Halogen-Free csBGA	132	COM
LCMXO2-4000HE-5MG132C	4320	1.2V	-5	Halogen-Free csBGA	132	COM
LCMXO2-4000HE-6MG132C	4320	1.2V	-6	Halogen-Free csBGA	132	COM
LCMXO2-4000HE-4BG256C	4320	1.2V	-4	Halogen-Free caBGA	256	COM
LCMXO2-4000HE-4MG184C	4320	1.2V	-4	Halogen-Free csBGA	184	COM
LCMXO2-4000HE-5MG184C	4320	1.2V	-5	Halogen-Free csBGA	184	COM
LCMXO2-4000HE-6MG184C	4320	1.2V	-6	Halogen-Free csBGA	184	COM
LCMXO2-4000HE-5BG256C	4320	1.2V	-5	Halogen-Free caBGA	256	COM
LCMXO2-4000HE-6BG256C	4320	1.2V	-6	Halogen-Free caBGA	256	COM
LCMXO2-4000HE-4FTG256C	4320	1.2V	-4	Halogen-Free ftBGA	256	COM
LCMXO2-4000HE-5FTG256C	4320	1.2V	-5	Halogen-Free ftBGA	256	COM
LCMXO2-4000HE-6FTG256C	4320	1.2V	-6	Halogen-Free ftBGA	256	COM
LCMXO2-4000HE-4BG332C	4320	1.2V	-4	Halogen-Free caBGA	332	COM
LCMXO2-4000HE-5BG332C	4320	1.2V	-5	Halogen-Free caBGA	332	COM
LCMXO2-4000HE-6BG332C	4320	1.2V	-6	Halogen-Free caBGA	332	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000HE-4FG484C	4320	1.2V	-4	Halogen-Free fpBGA	484	COM
LCMXO2-4000HE-5FG484C	4320	1.2V	-5	Halogen-Free fpBGA	484	COM
LCMXO2-4000HE-6FG484C	4320	1.2V	-6	Halogen-Free fpBGA	484	COM

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-7000HE-4TG144C	6864	1.2V	-4	Halogen-Free TQFP	144	COM
LCMXO2-7000HE-5TG144C	6864	1.2V	-5	Halogen-Free TQFP	144	COM
LCMXO2-7000HE-6TG144C	6864	1.2V	-6	Halogen-Free TQFP	144	COM
LCMXO2-7000HE-4BG256C	6864	1.2V	-4	Halogen-Free caBGA	256	COM
LCMXO2-7000HE-5BG256C	6864	1.2V	-5	Halogen-Free caBGA	256	COM
LCMXO2-7000HE-6BG256C	6864	1.2V	-6	Halogen-Free caBGA	256	COM
LCMXO2-7000HE-4FTG256C	6864	1.2V	-4	Halogen-Free ftBGA	256	COM
LCMXO2-7000HE-5FTG256C	6864	1.2V	-5	Halogen-Free ftBGA	256	COM
LCMXO2-7000HE-6FTG256C	6864	1.2V	-6	Halogen-Free ftBGA	256	COM
LCMXO2-7000HE-4BG332C	6864	1.2V	-4	Halogen-Free caBGA	332	COM
LCMXO2-7000HE-5BG332C	6864	1.2V	-5	Halogen-Free caBGA	332	COM
LCMXO2-7000HE-6BG332C	6864	1.2V	-6	Halogen-Free caBGA	332	COM
LCMXO2-7000HE-4FG484C	6864	1.2V	-4	Halogen-Free fpBGA	484	COM
LCMXO2-7000HE-5FG484C	6864	1.2V	-5	Halogen-Free fpBGA	484	COM
LCMXO2-7000HE-6FG484C	6864	1.2V	-6	Halogen-Free fpBGA	484	COM

Ultra Low Power Industrial Grade Devices, Halogen Free (RoHS) Packaging

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-256ZE-1SG32I	256	1.2V	-1	Halogen-Free QFN	32	IND
LCMXO2-256ZE-2SG32I	256	1.2V	-2	Halogen-Free QFN	32	IND
LCMXO2-256ZE-3SG32I	256	1.2V	-3	Halogen-Free QFN	32	IND
LCMXO2-256ZE-1UMG64I	256	1.2V	-1	Halogen-Free ucBGA	64	IND
LCMXO2-256ZE-2UMG64I	256	1.2V	-2	Halogen-Free ucBGA	64	IND
LCMXO2-256ZE-3UMG64I	256	1.2V	-3	Halogen-Free ucBGA	64	IND
LCMXO2-256ZE-1TG100I	256	1.2V	-1	Halogen-Free TQFP	100	IND
LCMXO2-256ZE-2TG100I	256	1.2V	-2	Halogen-Free TQFP	100	IND
LCMXO2-256ZE-3TG100I	256	1.2V	-3	Halogen-Free TQFP	100	IND
LCMXO2-256ZE-1MG132I	256	1.2V	-1	Halogen-Free csBGA	132	IND
LCMXO2-256ZE-2MG132I	256	1.2V	-2	Halogen-Free csBGA	132	IND
LCMXO2-256ZE-3MG132I	256	1.2V	-3	Halogen-Free csBGA	132	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-640ZE-1TG100I	640	1.2V	-1	Halogen-Free TQFP	100	IND
LCMXO2-640ZE-2TG100I	640	1.2V	-2	Halogen-Free TQFP	100	IND
LCMXO2-640ZE-3TG100I	640	1.2V	-3	Halogen-Free TQFP	100	IND
LCMXO2-640ZE-1MG132I	640	1.2V	-1	Halogen-Free csBGA	132	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-640ZE-2MG132I	640	1.2V	-2	Halogen-Free csBGA	132	IND
LCMXO2-640ZE-3MG132I	640	1.2V	-3	Halogen-Free csBGA	132	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000HE-4MG184I	4320	1.2V	-4	Halogen-Free csBGA	184	IND
LCMXO2-4000HE-5MG184I	4320	1.2V	-5	Halogen-Free csBGA	184	IND
LCMXO2-4000HE-6MG184I	4320	1.2V	-6	Halogen-Free caBGA	184	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200ZE-1UWG25ITR[1]	1280	1.2V	-1	Halogen-Free WLCSP	25	IND
LCMXO2-1200ZE-1UWG25ITR50[2]	1280	1.2V	-1	Halogen-Free WLCSP	25	IND
LCMXO2-1200ZE-1TG100I	1280	1.2V	-1	Halogen-Free TQFP	100	IND
LCMXO2-1200ZE-2TG100I	1280	1.2V	-2	Halogen-Free TQFP	100	IND
LCMXO2-1200ZE-3TG100I	1280	1.2V	-3	Halogen-Free TQFP	100	IND
LCMXO2-1200ZE-1MG132I	1280	1.2V	-1	Halogen-Free csBGA	132	IND
LCMXO2-1200ZE-2MG132I	1280	1.2V	-2	Halogen-Free csBGA	132	IND
LCMXO2-1200ZE-3MG132I	1280	1.2V	-3	Halogen-Free csBGA	132	IND
LCMXO2-1200ZE-1TG144I	1280	1.2V	-1	Halogen-Free TQFP	144	IND
LCMXO2-1200ZE-2TG144I	1280	1.2V	-2	Halogen-Free TQFP	144	IND
LCMXO2-1200ZE-3TG144I	1280	1.2V	-3	Halogen-Free TQFP	144	IND

1. This part number has a tape and reel quantity of 5,000 units with a minimum order quantity of 10,000 units. Order quantities must be in increments of 10,000 units. For example, a 10,000 unit order will be shipped in two reels with one reel containing 5,000 units and the other reel with less than 5,000 units (depending on test yields). Unserviced backlog will be canceled.
2. This part number has a tape and reel quantity of 50 units with a minimum order quantity of 50. Order quantities must be in increments of 50 units. For example, a 1000 unit order will be shipped as 20 reels of 50 units each.

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000ZE-1TG100I	2112	1.2V	-1	Halogen-Free TQFP	100	IND
LCMXO2-2000ZE-2TG100I	2112	1.2V	-2	Halogen-Free TQFP	100	IND
LCMXO2-2000ZE-3TG100I	2112	1.2V	-3	Halogen-Free TQFP	100	IND
LCMXO2-2000ZE-1MG132I	2112	1.2V	-1	Halogen-Free csBGA	132	IND
LCMXO2-2000ZE-2MG132I	2112	1.2V	-2	Halogen-Free csBGA	132	IND
LCMXO2-2000ZE-3MG132I	2112	1.2V	-3	Halogen-Free csBGA	132	IND
LCMXO2-2000ZE-1TG144I	2112	1.2V	-1	Halogen-Free TQFP	144	IND
LCMXO2-2000ZE-2TG144I	2112	1.2V	-2	Halogen-Free TQFP	144	IND
LCMXO2-2000ZE-3TG144I	2112	1.2V	-3	Halogen-Free TQFP	144	IND
LCMXO2-2000ZE-1BG256I	2112	1.2V	-1	Halogen-Free caBGA	256	IND
LCMXO2-2000ZE-2BG256I	2112	1.2V	-2	Halogen-Free caBGA	256	IND
LCMXO2-2000ZE-3BG256I	2112	1.2V	-3	Halogen-Free caBGA	256	IND
LCMXO2-2000ZE-1FTG256I	2112	1.2V	-1	Halogen-Free ftBGA	256	IND
LCMXO2-2000ZE-2FTG256I	2112	1.2V	-2	Halogen-Free ftBGA	256	IND
LCMXO2-2000ZE-3FTG256I	2112	1.2V	-3	Halogen-Free ftBGA	256	IND

1. Samples can be ordered in minimum order quantities and increments of 50 units. Production volumes can be ordered in minimum order quantities and increments of 10,000 units for the LCMXO2-1200ZE in the 25-ball WLCSP package.

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000ZE-1MG132I	4320	1.2V	-1	Halogen-Free csBGA	132	IND
LCMXO2-4000ZE-2MG132I	4320	1.2V	-2	Halogen-Free csBGA	132	IND
LCMXO2-4000ZE-3MG132I	4320	1.2V	-3	Halogen-Free csBGA	132	IND
LCMXO2-4000ZE-1TG144I	4320	1.2V	-1	Halogen-Free TQFP	144	IND
LCMXO2-4000ZE-2TG144I	4320	1.2V	-2	Halogen-Free TQFP	144	IND
LCMXO2-4000ZE-3TG144I	4320	1.2V	-3	Halogen-Free TQFP	144	IND
LCMXO2-4000ZE-1BG256I	4320	1.2V	-1	Halogen-Free caBGA	256	IND
LCMXO2-4000ZE-2BG256I	4320	1.2V	-2	Halogen-Free caBGA	256	IND
LCMXO2-4000ZE-3BG256I	4320	1.2V	-3	Halogen-Free caBGA	256	IND
LCMXO2-4000ZE-1FTG256I	4320	1.2V	-1	Halogen-Free ftBGA	256	IND
LCMXO2-4000ZE-2FTG256I	4320	1.2V	-2	Halogen-Free ftBGA	256	IND
LCMXO2-4000ZE-3FTG256I	4320	1.2V	-3	Halogen-Free ftBGA	256	IND
LCMXO2-4000ZE-1BG332I	4320	1.2V	-1	Halogen-Free caBGA	332	IND
LCMXO2-4000ZE-2BG332I	4320	1.2V	-2	Halogen-Free caBGA	332	IND
LCMXO2-4000ZE-3BG332I	4320	1.2V	-3	Halogen-Free caBGA	332	IND
LCMXO2-4000ZE-1FG484I	4320	1.2V	-1	Halogen-Free fpBGA	484	IND
LCMXO2-4000ZE-2FG484I	4320	1.2V	-2	Halogen-Free fpBGA	484	IND
LCMXO2-4000ZE-3FG484I	4320	1.2V	-3	Halogen-Free fpBGA	484	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-7000ZE-1TG144I	6864	1.2V	-1	Halogen-Free TQFP	144	IND
LCMXO2-7000ZE-2TG144I	6864	1.2V	-2	Halogen-Free TQFP	144	IND
LCMXO2-7000ZE-3TG144I	6864	1.2V	-3	Halogen-Free TQFP	144	IND
LCMXO2-7000ZE-1BG256I	6864	1.2V	-1	Halogen-Free caBGA	256	IND
LCMXO2-7000ZE-2BG256I	6864	1.2V	-2	Halogen-Free caBGA	256	IND
LCMXO2-7000ZE-3BG256I	6864	1.2V	-3	Halogen-Free caBGA	256	IND
LCMXO2-7000ZE-1FTG256I	6864	1.2V	-1	Halogen-Free ftBGA	256	IND
LCMXO2-7000ZE-2FTG256I	6864	1.2V	-2	Halogen-Free ftBGA	256	IND
LCMXO2-7000ZE-3FTG256I	6864	1.2V	-3	Halogen-Free ftBGA	256	IND
LCMXO2-7000ZE-1BG332I	6864	1.2V	-1	Halogen-Free caBGA	332	IND
LCMXO2-7000ZE-2BG332I	6864	1.2V	-2	Halogen-Free caBGA	332	IND
LCMXO2-7000ZE-3BG332I	6864	1.2V	-3	Halogen-Free caBGA	332	IND
LCMXO2-7000ZE-1FG484I	6864	1.2V	-1	Halogen-Free fpBGA	484	IND
LCMXO2-7000ZE-2FG484I	6864	1.2V	-2	Halogen-Free fpBGA	484	IND
LCMXO2-7000ZE-3FG484I	6864	1.2V	-3	Halogen-Free fpBGA	484	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200ZE-1TG100IR1[1]	1280	1.2V	-1	Halogen-Free TQFP	100	IND
LCMXO2-1200ZE-2TG100IR1[1]	1280	1.2V	-2	Halogen-Free TQFP	100	IND
LCMXO2-1200ZE-3TG100IR1[1]	1280	1.2V	-3	Halogen-Free TQFP	100	IND
LCMXO2-1200ZE-1MG132IR1[1]	1280	1.2V	-1	Halogen-Free csBGA	132	IND
LCMXO2-1200ZE-2MG132IR1[1]	1280	1.2V	-2	Halogen-Free csBGA	132	IND
LCMXO2-1200ZE-3MG132IR1[1]	1280	1.2V	-3	Halogen-Free csBGA	132	IND
LCMXO2-1200ZE-1TG144IR1[1]	1280	1.2V	-1	Halogen-Free TQFP	144	IND
LCMXO2-1200ZE-2TG144IR1[1]	1280	1.2V	-2	Halogen-Free TQFP	144	IND
LCMXO2-1200ZE-3TG144IR1[1]	1280	1.2V	-3	Halogen-Free TQFP	144	IND

1. Specifications for the “LCMXO2-1200ZE-speed package IR1” are the same as the “LCMXO2-1200ZE-speed package I” devices respectively, except as specified in the R1 Device Specifications section on page 5-18 of this data sheet.

High-Performance Industrial Grade Devices with Voltage Regulator, Halogen Free (RoHS) Packaging

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-256HC-4SG32I	256	2.5V/3.3V	-4	Halogen-Free QFN	32	IND
LCMXO2-256HC-5SG32I	256	2.5V/3.3V	-5	Halogen-Free QFN	32	IND
LCMXO2-256HC-6SG32I	256	2.5V/3.3V	-6	Halogen-Free QFN	32	IND
LCMXO2-256HC-4UMG64I	256	2.5V/3.3V	-4	Halogen-Free ucBGA	64	IND
LCMXO2-256HC-5UMG64I	256	2.5V/3.3V	-5	Halogen-Free ucBGA	64	IND
LCMXO2-256HC-6UMG64I	256	2.5V/3.3V	-6	Halogen-Free ucBGA	64	IND
LCMXO2-256HC-4TG100I	256	2.5V/3.3V	-4	Halogen-Free TQFP	100	IND
LCMXO2-256HC-5TG100I	256	2.5V/3.3V	-5	Halogen-Free TQFP	100	IND
LCMXO2-256HC-6TG100I	256	2.5V/3.3V	-6	Halogen-Free TQFP	100	IND
LCMXO2-256HC-4MG132I	256	2.5V/3.3V	-4	Halogen-Free csBGA	132	IND
LCMXO2-256HC-5MG132I	256	2.5V/3.3V	-5	Halogen-Free csBGA	132	IND
LCMXO2-256HC-6MG132I	256	2.5V/3.3V	-6	Halogen-Free csBGA	132	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-640HC-4TG100I	640	2.5V/3.3V	-4	Halogen-Free TQFP	100	IND
LCMXO2-640HC-5TG100I	640	2.5V/3.3V	-5	Halogen-Free TQFP	100	IND
LCMXO2-640HC-6TG100I	640	2.5V/3.3V	-6	Halogen-Free TQFP	100	IND
LCMXO2-640HC-4MG132I	640	2.5V/3.3V	-4	Halogen-Free csBGA	132	IND
LCMXO2-640HC-5MG132I	640	2.5V/3.3V	-5	Halogen-Free csBGA	132	IND
LCMXO2-640HC-6MG132I	640	2.5V/3.3V	-6	Halogen-Free csBGA	132	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-640UHC-4TG144I	640	2.5V/3.3V	-4	Halogen-Free TQFP	144	IND
LCMXO2-640UHC-5TG144I	640	2.5V/3.3V	-5	Halogen-Free TQFP	144	IND
LCMXO2-640UHC-6TG144I	640	2.5V/3.3V	-6	Halogen-Free TQFP	144	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200HC-4TG100I	1280	2.5V/3.3V	-4	Halogen-Free TQFP	100	IND
LCMXO2-1200HC-5TG100I	1280	2.5V/3.3V	-5	Halogen-Free TQFP	100	IND
LCMXO2-1200HC-6TG100I	1280	2.5V/3.3V	-6	Halogen-Free TQFP	100	IND
LCMXO2-1200HC-4MG132I	1280	2.5V/3.3V	-4	Halogen-Free csBGA	132	IND
LCMXO2-1200HC-5MG132I	1280	2.5V/3.3V	-5	Halogen-Free csBGA	132	IND
LCMXO2-1200HC-6MG132I	1280	2.5V/3.3V	-6	Halogen-Free csBGA	132	IND
LCMXO2-1200HC-4TG144I	1280	2.5V/3.3V	-4	Halogen-Free TQFP	144	IND
LCMXO2-1200HC-5TG144I	1280	2.5V/3.3V	-5	Halogen-Free TQFP	144	IND
LCMXO2-1200HC-6TG144I	1280	2.5V/3.3V	-6	Halogen-Free TQFP	144	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200UHC-4FTG256I	1280	2.5V/3.3V	-4	Halogen-Free ftBGA	256	IND
LCMXO2-1200UHC-5FTG256I	1280	2.5V/3.3V	-5	Halogen-Free ftBGA	256	IND
LCMXO2-1200UHC-6FTG256I	1280	2.5V/3.3V	-6	Halogen-Free ftBGA	256	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000HC-4TG100I	2112	2.5V/3.3V	-4	Halogen-Free TQFP	100	IND
LCMXO2-2000HC-5TG100I	2112	2.5V/3.3V	-5	Halogen-Free TQFP	100	IND
LCMXO2-2000HC-6TG100I	2112	2.5V/3.3V	-6	Halogen-Free TQFP	100	IND
LCMXO2-2000HC-4MG132I	2112	2.5V/3.3V	-4	Halogen-Free csBGA	132	IND
LCMXO2-2000HC-5MG132I	2112	2.5V/3.3V	-5	Halogen-Free csBGA	132	IND
LCMXO2-2000HC-6MG132I	2112	2.5V/3.3V	-6	Halogen-Free csBGA	132	IND
LCMXO2-2000HC-4TG144I	2112	2.5V/3.3V	-4	Halogen-Free TQFP	144	IND
LCMXO2-2000HC-5TG144I	2112	2.5V/3.3V	-5	Halogen-Free TQFP	144	IND
LCMXO2-2000HC-6TG144I	2112	2.5V/3.3V	-6	Halogen-Free TQFP	144	IND
LCMXO2-2000HC-4BG256I	2112	2.5V/3.3V	-4	Halogen-Free caBGA	256	IND
LCMXO2-2000HC-5BG256I	2112	2.5V/3.3V	-5	Halogen-Free caBGA	256	IND
LCMXO2-2000HC-6BG256I	2112	2.5V/3.3V	-6	Halogen-Free caBGA	256	IND
LCMXO2-2000HC-4FTG256I	2112	2.5V/3.3V	-4	Halogen-Free ftBGA	256	IND
LCMXO2-2000HC-5FTG256I	2112	2.5V/3.3V	-5	Halogen-Free ftBGA	256	IND
LCMXO2-2000HC-6FTG256I	2112	2.5V/3.3V	-6	Halogen-Free ftBGA	256	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000UHC-4FG484I	2112	2.5V/3.3V	-4	Halogen-Free fpBGA	484	IND
LCMXO2-2000UHC-5FG484I	2112	2.5V/3.3V	-5	Halogen-Free fpBGA	484	IND
LCMXO2-2000UHC-6FG484I	2112	2.5V/3.3V	-6	Halogen-Free fpBGA	484	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000HC-4TG144I	4320	2.5V/3.3V	-4	Halogen-Free TQFP	144	IND
LCMXO2-4000HC-5TG144I	4320	2.5V/3.3V	-5	Halogen-Free TQFP	144	IND
LCMXO2-4000HC-6TG144I	4320	2.5V/3.3V	-6	Halogen-Free TQFP	144	IND
LCMXO2-4000HC-4MG132I	4320	2.5V/3.3V	-4	Halogen-Free csBGA	132	IND
LCMXO2-4000HC-5MG132I	4320	2.5V/3.3V	-5	Halogen-Free csBGA	132	IND
LCMXO2-4000HC-6MG132I	4320	2.5V/3.3V	-6	Halogen-Free csBGA	132	IND
LCMXO2-4000HC-4BG256I	4320	2.5V/3.3V	-4	Halogen-Free caBGA	256	IND
LCMXO2-4000HC-5BG256I	4320	2.5V/3.3V	-5	Halogen-Free caBGA	256	IND
LCMXO2-4000HC-6BG256I	4320	2.5V/3.3V	-6	Halogen-Free caBGA	256	IND
LCMXO2-4000HC-4FTG256I	4320	2.5V/3.3V	-4	Halogen-Free ftBGA	256	IND
LCMXO2-4000HC-5FTG256I	4320	2.5V/3.3V	-5	Halogen-Free ftBGA	256	IND
LCMXO2-4000HC-6FTG256I	4320	2.5V/3.3V	-6	Halogen-Free ftBGA	256	IND
LCMXO2-4000HC-4BG332I	4320	2.5V/3.3V	-4	Halogen-Free caBGA	332	IND
LCMXO2-4000HC-5BG332I	4320	2.5V/3.3V	-5	Halogen-Free caBGA	332	IND
LCMXO2-4000HC-6BG332I	4320	2.5V/3.3V	-6	Halogen-Free caBGA	332	IND
LCMXO2-4000HC-4FG484I	4320	2.5V/3.3V	-4	Halogen-Free fpBGA	484	IND
LCMXO2-4000HC-5FG484I	4320	2.5V/3.3V	-5	Halogen-Free fpBGA	484	IND
LCMXO2-4000HC-6FG484I	4320	2.5V/3.3V	-6	Halogen-Free fpBGA	484	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-7000HC-4TG144I	6864	2.5V/3.3V	-4	Halogen-Free TQFP	144	IND
LCMXO2-7000HC-5TG144I	6864	2.5V/3.3V	-5	Halogen-Free TQFP	144	IND
LCMXO2-7000HC-6TG144I	6864	2.5V/3.3V	-6	Halogen-Free TQFP	144	IND
LCMXO2-7000HC-4BG256I	6864	2.5V/3.3V	-4	Halogen-Free caBGA	256	IND
LCMXO2-7000HC-5BG256I	6864	2.5V/3.3V	-5	Halogen-Free caBGA	256	IND
LCMXO2-7000HC-6BG256I	6864	2.5V/3.3V	-6	Halogen-Free caBGA	256	IND
LCMXO2-7000HC-4FTG256I	6864	2.5V/3.3V	-4	Halogen-Free ftBGA	256	IND
LCMXO2-7000HC-5FTG256I	6864	2.5V/3.3V	-5	Halogen-Free ftBGA	256	IND
LCMXO2-7000HC-6FTG256I	6864	2.5V/3.3V	-6	Halogen-Free ftBGA	256	IND
LCMXO2-7000HC-4BG332I	6864	2.5V/3.3V	-4	Halogen-Free caBGA	332	IND
LCMXO2-7000HC-5BG332I	6864	2.5V/3.3V	-5	Halogen-Free caBGA	332	IND
LCMXO2-7000HC-6BG332I	6864	2.5V/3.3V	-6	Halogen-Free caBGA	332	IND
LCMXO2-7000HC-4FG484I	6864	2.5V/3.3V	-4	Halogen-Free fpBGA	484	IND
LCMXO2-7000HC-5FG484I	6864	2.5V/3.3V	-5	Halogen-Free fpBGA	484	IND
LCMXO2-7000HC-6FG484I	6864	2.5V/3.3V	-6	Halogen-Free fpBGA	484	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-1200HC-4TG100IR1[1]	1280	2.5V/3.3V	-4	Halogen-Free TQFP	100	IND
LCMXO2-1200HC-5TG100IR1[1]	1280	2.5V/3.3V	-5	Halogen-Free TQFP	100	IND
LCMXO2-1200HC-6TG100IR1[1]	1280	2.5V/3.3V	-6	Halogen-Free TQFP	100	IND
LCMXO2-1200HC-4MG132IR1[1]	1280	2.5V/3.3V	-4	Halogen-Free csBGA	132	IND
LCMXO2-1200HC-5MG132IR1[1]	1280	2.5V/3.3V	-5	Halogen-Free csBGA	132	IND
LCMXO2-1200HC-6MG132IR1[1]	1280	2.5V/3.3V	-6	Halogen-Free csBGA	132	IND
LCMXO2-1200HC-4TG144IR1[1]	1280	2.5V/3.3V	-4	Halogen-Free TQFP	144	IND
LCMXO2-1200HC-5TG144IR1[1]	1280	2.5V/3.3V	-5	Halogen-Free TQFP	144	IND
LCMXO2-1200HC-6TG144IR1[1]	1280	2.5V/3.3V	-6	Halogen-Free TQFP	144	IND

1. Specifications for the “LCMXO2-1200HC-speed package IR1” are the same as the “LCMXO2-1200ZE-speed package I” devices respectively, except as specified in the R1 Device Specifications section on page 5-18 of this data sheet.

High Performance Industrial Grade Devices Without Voltage Regulator, Halogen Free (RoHS) Packaging

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000HE-4TG100I	2112	1.2V	-4	Halogen-Free TQFP	100	IND
LCMXO2-2000HE-5TG100I	2112	1.2V	-5	Halogen-Free TQFP	100	IND
LCMXO2-2000HE-6TG100I	2112	1.2V	-6	Halogen-Free TQFP	100	IND
LCMXO2-2000HE-4MG132I	2112	1.2V	-4	Halogen-Free csBGA	132	IND
LCMXO2-2000HE-5MG132I	2112	1.2V	-5	Halogen-Free csBGA	132	IND
LCMXO2-2000HE-6MG132I	2112	1.2V	-6	Halogen-Free csBGA	132	IND
LCMXO2-2000HE-4TG144I	2112	1.2V	-4	Halogen-Free TQFP	144	IND
LCMXO2-2000HE-5TG144I	2112	1.2V	-5	Halogen-Free TQFP	144	IND
LCMXO2-2000HE-6TG144I	2112	1.2V	-6	Halogen-Free TQFP	144	IND
LCMXO2-2000HE-4BG256I	2112	1.2V	-4	Halogen-Free caBGA	256	IND
LCMXO2-2000HE-5BG256I	2112	1.2V	-5	Halogen-Free caBGA	256	IND
LCMXO2-2000HE-6BG256I	2112	1.2V	-6	Halogen-Free caBGA	256	IND
LCMXO2-2000HE-4FTG256I	2112	1.2V	-4	Halogen-Free ftBGA	256	IND
LCMXO2-2000HE-5FTG256I	2112	1.2V	-5	Halogen-Free ftBGA	256	IND
LCMXO2-2000HE-6FTG256I	2112	1.2V	-6	Halogen-Free ftBGA	256	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-2000UHE-4FG484I	2112	1.2V	-4	Halogen-Free fpBGA	484	IND
LCMXO2-2000UHE-5FG484I	2112	1.2V	-5	Halogen-Free fpBGA	484	IND
LCMXO2-2000UHE-6FG484I	2112	1.2V	-6	Halogen-Free fpBGA	484	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-4000HE-4MG132I	4320	1.2V	-4	Halogen-Free csBGA	132	IND
LCMXO2-4000HE-5MG132I	4320	1.2V	-5	Halogen-Free csBGA	132	IND
LCMXO2-4000HE-6MG132I	4320	1.2V	-6	Halogen-Free csBGA	132	IND
LCMXO2-4000HE-4TG144I	4320	1.2V	-4	Halogen-Free TQFP	144	IND
LCMXO2-4000HE-5TG144I	4320	1.2V	-5	Halogen-Free TQFP	144	IND
LCMXO2-4000HE-6TG144I	4320	1.2V	-6	Halogen-Free TQFP	144	IND
LCMXO2-4000HE-4BG256I	4320	1.2V	-4	Halogen-Free caBGA	256	IND
LCMXO2-4000HE-5BG256I	4320	1.2V	-5	Halogen-Free caBGA	256	IND
LCMXO2-4000HE-6BG256I	4320	1.2V	-6	Halogen-Free caBGA	256	IND
LCMXO2-4000HE-4FTG256I	4320	1.2V	-4	Halogen-Free ftBGA	256	IND
LCMXO2-4000HE-5FTG256I	4320	1.2V	-5	Halogen-Free ftBGA	256	IND
LCMXO2-4000HE-6FTG256I	4320	1.2V	-6	Halogen-Free ftBGA	256	IND
LCMXO2-4000HE-4BG332I	4320	1.2V	-4	Halogen-Free caBGA	332	IND
LCMXO2-4000HE-5BG332I	4320	1.2V	-5	Halogen-Free caBGA	332	IND
LCMXO2-4000HE-6BG332I	4320	1.2V	-6	Halogen-Free caBGA	332	IND
LCMXO2-4000HE-4FG484I	4320	1.2V	-4	Halogen-Free fpBGA	484	IND
LCMXO2-4000HE-5FG484I	4320	1.2V	-5	Halogen-Free fpBGA	484	IND
LCMXO2-4000HE-6FG484I	4320	1.2V	-6	Halogen-Free fpBGA	484	IND

Part Number	LUTs	Supply Voltage	Grade	Package	Leads	Temp.
LCMXO2-7000HE-4TG144I	6864	1.2V	-4	Halogen-Free TQFP	144	IND
LCMXO2-7000HE-5TG144I	6864	1.2V	-5	Halogen-Free TQFP	144	IND
LCMXO2-7000HE-6TG144I	6864	1.2V	-6	Halogen-Free TQFP	144	IND
LCMXO2-7000HE-4BG256I	6864	1.2V	-4	Halogen-Free caBGA	256	IND
LCMXO2-7000HE-5BG256I	6864	1.2V	-5	Halogen-Free caBGA	256	IND
LCMXO2-7000HE-6BG256I	6864	1.2V	-6	Halogen-Free caBGA	256	IND
LCMXO2-7000HE-4FTG256I	6864	1.2V	-4	Halogen-Free ftBGA	256	IND
LCMXO2-7000HE-5FTG256I	6864	1.2V	-5	Halogen-Free ftBGA	256	IND
LCMXO2-7000HE-6FTG256I	6864	1.2V	-6	Halogen-Free ftBGA	256	IND
LCMXO2-7000HE-4BG332I	6864	1.2V	-4	Halogen-Free caBGA	332	IND
LCMXO2-7000HE-5BG332I	6864	1.2V	-5	Halogen-Free caBGA	332	IND
LCMXO2-7000HE-6BG332I	6864	1.2V	-6	Halogen-Free caBGA	332	IND
LCMXO2-7000HE-4FG484I	6864	1.2V	-4	Halogen-Free fpBGA	484	IND
LCMXO2-7000HE-5FG484I	6864	1.2V	-5	Halogen-Free fpBGA	484	IND
LCMXO2-7000HE-6FG484I	6864	1.2V	-6	Halogen-Free fpBGA	484	IND

R1 Device Specifications

The LCMXO2-1200ZE/HC "R1" devices have the same specifications as their Standard (non-R1) counterparts except as listed below. For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard Non-R1) Devices.

- The User Flash Memory (UFM) cannot be programmed through the internal WISHBONE interface. It can still be programmed through the JTAG/SPI/I^2C ports.
- The on-chip differential input termination resistor value is higher than intended. It is approximately 200Ω as opposed to the intended 100Ω. It is recommended to use external termination resistors for differential inputs. The on-chip termination resistors can be disabled through Lattice design software.
- Soft Error Detection logic may not produce the correct result when it is run for the first time after configuration. To use this feature, discard the result from the first operation. Subsequent operations will produce the correct result.
- Under certain conditions, IIH exceeds data sheet specifications. The following table provides more details:

Condition	Clamp	Pad Rising IIH Max.	Pad Falling IIH Min.	Steady State Pad High IIH	Steady State Pad Low IIL
VPAD > VCCIO	OFF	1mA	-1mA	1mA	10µA
VPAD = VCCIO	ON	10µA	-10µA	10µA	10µA
VPAD = VCCIO	OFF	1mA	-1mA	1mA	10µA
VPAD < VCCIO	OFF	10µA	-10µA	10µA	10µA

- The user SPI interface does not operate correctly in some situations. During master read access and slave write access, the last byte received does not generate the RRDY interrupt.
- In GDDRX2, GDDRX4 and GDDR71 modes, ECLKSYNC may have a glitch in the output under certain conditions, leading to possible loss of synchronization.
- When using the hard I^2C IP core, the I^2C status registers I2C_1_SR and I2C_2_SR may not update correctly.
- PLL Lock signal will glitch high when coming out of standby. This glitch lasts for about 10µsec before returning low.
- Dual boot only available on HC devices, requires tying VCC and VCCIO2 to the same 3.3V or 2.5V supply.

MachXO2 Family Data Sheet Supplemental Information

For Further Information

A variety of technical notes for the MachXO2 family are available on the Lattice web site.

- TN1198, Power Estimation and Management for MachXO2 Devices
- TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide
- TN1201, Memory Usage Guide for MachXO2 Devices
- TN1202, MachXO2 sysIO Usage Guide
- TN1203, Implementing High-Speed Interfaces with MachXO2 Devices
- TN1204, MachXO2 Programming and Configuration Usage Guide
- TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices
- TN1206, MachXO2 SRAM CRC Error Detection Usage Guide
- TN1207, Using TraceID in MachXO2 Devices
- TN1074, PCB Layout Recommendations for BGA Packages
- TN1087, Minimizing System Interruption During Configuration Using TransFR Technology
- AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (non-R1) Devices
- AN8066, Boundary Scan Testability with Lattice sysIO Capability
- MachXO2 Device Pinout Files
- Thermal Management document
- Lattice design tools

For further information on interface standards, refer to the following web sites:

- JEDEC Standards (LVTTL, LVCMOS, LVDS, DDR, DDR2, LPDDR): www.jedec.org
- PCI: www.pcisig.com

MachXO2 Family Data Sheet Revision History

January 2013 **Data Sheet DS1035**

Date	Version	Section	Change Summary
November 2010	01.0	—	Initial release.
January 2011	01.1	All	Included ultra-high I/O devices.
		DC and Switching Characteristics	Recommended Operating Conditions table – Added footnote 3.
			DC Electrical Characteristics table – Updated data for I_{IL}, I_{IH}. V_{HYST} typical values updated.
			Generic DDRX2 Outputs with Clock and Data Aligned at Pin (GDDRX2_TX.ECLK.Aligned) Using PCLK Pin for Clock Input tables – Updated data for T_{DIA} and T_{DIB}.
			Generic DDRX4 Outputs with Clock and Data Aligned at Pin (GDDRX4_TX.ECLK.Aligned) Using PCLK Pin for Clock Input tables – Updated data for T_{DIA} and T_{DIB}.
			Power-On-Reset Voltage Levels table - clarified note 3.
			Clarified VCCIO related recommended operating conditions specifications.
			Added power supply ramp rate requirements.
			Added Power Supply Ramp Rates table.
			Updated Programming/Erase Specifications table.
			Removed references to V_{CCP}.
		Pinout Information	Included number of 7:1 and 8:1 gearboxes (input and output) in the pin information summary tables.
			Removed references to V_{CCP}.
April 2011	01.2	—	Data sheet status changed from Advance to Preliminary.
		Introduction	Updated MachXO2 Family Selection Guide table.
		Architecture	Updated Supported Input Standards table.
			Updated sysMEM Memory Primitives diagram.
			Added differential SSTL and HSTL IO standards.
		DC and Switching Characteristics	Updates following parameters: POR voltage levels, DC electrical characteristics, static supply current for ZE/HE/HC devices, static power consumption contribution of different components – ZE devices, programming and erase Flash supply current.
			Added VREF specifications to sysIO recommended operating conditions.
			Updating timing information based on characterization.
			Added differential SSTL and HSTL IO standards.
		Ordering Information	Added Ordering Part Numbers for R1 devices, and devices in WLCSP packages.
			Added R1 device specifications.
May 2011	01.3	Multiple	Replaced “SED” with “SRAM CRC Error Detection” throughout the document.
		DC and Switching Characteristics	Added footnote 1 to Program Erase Specifications table.
		Pinout Information	Updated Pin Information Summary tables.
			Signal name SO/SISPISO changed to SO/SPISO in the Signal Descriptions table.

Date	Version	Section	Change Summary
August 2011	01.4	Architecture	Updated information in Clock/Control Distribution Network and sysCLOCK Phase Locked Loops (PLLs).
		DC and Switching Characteristics	Updated I_{IL} and I_{IH} conditions in the DC Electrical Characteristics table.
		Pinout Information	Included number of 7:1 and 8:1 gearboxes (input and output) in the pin information summary tables.
			Updated Pin Information Summary table: Dual Function I/O, DQS Groups Bank 1, Total General Purpose Single-Ended I/O, Differential I/O Per Bank, Total Count of Bonded Pins, Gearboxes. Added column of data for MachXO2-2000 49 WLCSP.
		Ordering Information	Updated R1 Device Specifications text section with information on migration from MachXO2-1200-R1 to Standard (non-R1) devices.
			Corrected Supply Voltage typo for part numbers: LCMX02-2000UHE-4FG484I, LCMX02-2000UHE-5FG484I, LCMX02-2000UHE-6FG484I.
			Added footnote for WLCSP package parts.
		Supplemental Information	Removed reference to Stand-alone Power Calculator for MachXO2 Devices. Added reference to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (non-R1) Devices.
August 2011	01.5	DC and Switching Characteristics	Updated ESD information.
		Ordering Information	Updated footnote for ordering WLCSP devices.
February 2012	01.6	—	Data sheet status changed from preliminary to final.
		Introduction	MachXO2 Family Selection Guide table – Removed references to 49-ball WLCSP.
		DC and Switching Characteristics	Updated Flash Download Time table.
			Modified Storage Temperature in the Absolute Maximum Ratings section.
			Updated I_{DK} max in Hot Socket Specifications table.
			Modified Static Supply Current tables for ZE and HC/HE devices.
			Updated Power Supply Ramp Rates table.
			Updated Programming and Erase Supply Current tables.
			Updated data in the External Switching Characteristics table.
			Corrected Absolute Maximum Ratings for Dedicated Input Voltage Applied for LCMXO2 HC.
			DC Electrical Characteristics table – Minor corrections to conditions for I_{IL}, I_{IH}.
		Pinout Information	Removed references to 49-ball WLCSP.
			Signal Descriptions table – Updated description for GND, VCC, and VCCIOx.
			Updated Pin Information Summary table – Number of VCCIOs, GNDs, VCCs, and Total Count of Bonded Pins for MachXO2-256, 640, and 640U and Dual Function I/O for MachXO2-4000 332caBGA.
		Ordering Information	Removed references to 49-ball WLCSP
February 2012	01.7	All	Updated document with new corporate logo.
March 2012	01.8	Introduction	Added 32 QFN packaging information to Features bullets and MachXO2 Family Selection Guide table.
		DC and Switching Characteristics	Changed ‘STANDBY’ to ‘USERSTDBY’ in Standby Mode timing diagram.
		Pinout Information	Removed footnote from Pin Information Summary tables.

Date	Version	Section	Change Summary
March 2012 (cont.)	01.8 (cont.)	Pinout Information (cont.)	Added 32 QFN package to Pin Information Summary table.
		Ordering Information	Updated Part Number Description and Ordering Information tables for 32 QFN package.
			Updated topside mark diagram in the Ordering Information section.
April 2012	01.9	Architecture	Removed references to TN1200.
		Ordering Information	Updated the Device Status portion of the MachXO2 Part Number Description to include the 50 parts per reel for the WLCSP package.
			Added new part number and footnote 2 for LCMXO2-1200ZE-1UWG25ITR50.
			Updated footnote 1 for LCMXO2-1200ZE-1UWG25ITR.
		Supplemental Information	Removed references to TN1200.
January 2013	02.0	Introduction	Updated the total number IOs to include JTAGENB.
		Architecture	Supported Output Standards table – Added 3.3 V_{CCIO} (Typ.) to LVDS row.
			Changed SRAM CRC Error Detection to Soft Error Detection.
		DC and Switching Characteristics	Power Supply Ramp Rates table – Updated Units column for t_{RAMP} symbol.
			Added new Maximum sysIO Buffer Performance table.
			sysCLOCK PLL Timing table – Updated Min. column values for f_{IN}, f_{OUT}, f_{OUT2} and f_{PFD} parameters. Added t_{SPO} parameter. Updated footnote 6.
			MachXO2 Oscillator Output Frequency table – Updated symbol name for $t_{STABLEOSC.}$
			DC Electrical Characteristics table – Updated conditions for $I_{IL,}$ I_{IH} symbols.
			Corrected parameters tDQVBS and tDQVAS
			Corrected MachXO2 ZE parameters tDVADQ and tDVEDQ
		Pinout Information	Included the MachXO2-4000HE 184 csBGA package.
		Ordering Information	Updated part number.

Section II. MachXO2 Family Technical Notes

Power Estimation and Management for MachXO2 Devices

December 2012 Technical Note TN1198

Introduction

A key requirement for many of today's high volume FPGA applications is low power consumption. The MachXO2™ PLD provides many power-saving features including Power Controller, Bank Controller and Power Guard. This technical note provides users with detail for using the MachXO2 low power architectural features including power supply considerations and power estimations provided by the Power Calculator tool.

Power Modes

FPGA designers often minimize power consumption by turning off subsystems while configured and operational. Design modes of operation are typically categorized into the following:

Normal operation:

- Device is fully operational and all circuits are active
- Highest power consumption

Low power operation:

- Subsystems are dynamically shut down when not required
- Average to low power consumption

Ultra low power standby:

- All subsystems are shut down
- Lowest power consumption provides the best option for prolonged battery life

The MachXO2 offers a flexible architecture that allows many on-chip components to be dynamically turned off during the low power operation modes. These features are listed in Table 8-1.

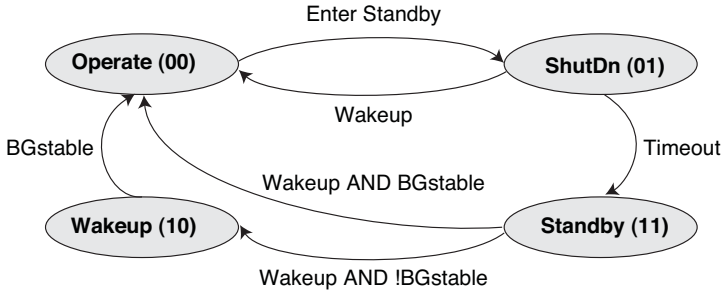
Table 8-1. MachXO2 Components with Low Power Features

Device Component	Description
Bandgap	The Bandgap can be turned off in standby mode. When the Bandgap is turned off analog circuitry such as the PLLs, on-chip oscillator, and referenced and differential I/O buffers are also turned off.
Power-On-Reset (POR)	The POR can be turned off in standby mode. This circuit monitors V_{CC} levels. In the event of unsafe V_{CC} drops, this circuit reconfigures the MachXO2 device. When the POR circuitry is turned off, limited power detection circuitry is still active. This option is only recommended for applications in which the power supply rails are reliable.
On-Chip Oscillator	The on-chip oscillator has two power-saving features. It may be statically switched off when not used in a design. It can also be turned off in standby mode.
PLL	Similar to the on-chip oscillator, the PLL also has two power-saving features. It can be statically switched off if it is not needed in a design. It can also be turned off in standby mode.
I/O Bank Controller	Referenced and differential I/O buffers (used to implement standards such as HSTL, SSTL and LVDS) consume more than ratioed single-ended I/Os such as LVCMOS. The I/O bank controller allows the designer to turn these I/Os off dynamically on a per-bank basis.
Dynamic Clock Enable for Primary Clock Nets	Each primary clock net can be dynamically disabled to save power.
Power Guard	Power Guard is a feature implemented in input buffers. This feature allows designers to switch off the input buffer when it is not needed. This feature can be used in both clock and data paths. Its biggest impact is in the standby mode when it can be used to switch off clock inputs that are distributed using general routing resources.

Power Controller

The MachXO2 PLD includes a Power Controller to ensure smooth transitions into and out of standby mode. The Power Controller's two primary signals, STOP and STDBY, transitions are shown in Figure 8-1.

Figure 8-1. Power ControllerFigure 8-1 State Diagram (STDBY, STOP)



The detailed Power Controller block diagram is shown in Figure 8-2 and its ports are defined in Table 8-2. The time-out signal is generated from an optional 8-bit Timer Counter which divides the input clock source by 2^8 (or 256). This injects a delay between the STOP and STDBY signals.

Figure 8-2. Power Controller Block Diagram

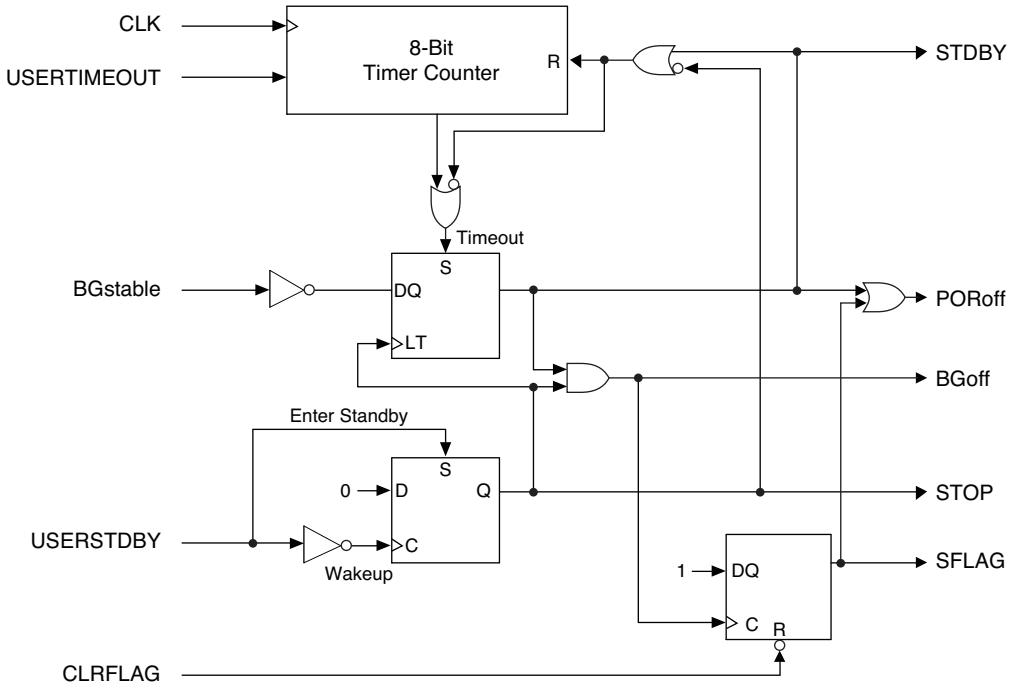


Table 8-2. Power Controller Signals

Port	I/O	Optional	Description
CLK	Input	Yes	Clock source to the Timer Delay block.
USERTIMEOUT	Input	Yes	An active high enable signal for the Timer Counter block.
USERSTDBY	Input or Hardwired	Yes	A rising edge of this signal begins the shut-down sequence to enter standby. A falling edge of this signal begins the wake up sequence from standby. The USERSTDBY signal is from user logic or hardwire from the Configuration logic. The signals CFGSTDBY and CFGWAKE are used to provide simulation support for the Configuration I^2C and SPI standby and wake commands. These ports should be connected to the matching ports on the EFB.
CLRFLAG	Input	Yes	Asynchronous active high reset for the standby flag. User logic should assert a high pulse on this signal once device has woken up to clear the SFLAG.
STOP	Output	Yes	Active high signal which is a precursor to the STDBY signal. The delay from STOP to STDBY is determined by the 8-bit Timer Counter block. The STOP signal is used to prepare logic for standby by switching off clocks, signals etc.
STDBY	Output	No	Active high signal through general routing to user logic, I/O pads, oscillator, and PLLs. Used to place logic in standby.
SFLAG	Output	Yes	This flag signal goes high to alert user logic that the device is in standby.
PORoff[1]	Hardwired	Yes	Shut-off signal to the Power Detector circuitry (POR). Power Detector circuitry is used to determine if there has been a drop in V_{CC}. If so, the device will be reconfigured.
BGoff[1]	Hardwired	Yes	Shut-off signal to bandgap circuitry found in the MachXO2 ZE and HE versions for additional power savings. When the bandgap circuitry is turned off, POR circuitry, analog circuits (PLL, oscillator, referenced LVCMOS I/Os, SSTL I/Os, HTSTL I/Os, and differential I/Os) are turned off.
BGstable	Hardwired	No	The BGstable signal is a signal from the bandgap circuit. This signal will only be released after the bandgap circuitry is stable.

1. When POR is shuf off, limited Power Detector circuitry is still active. This option is only recommended for applications where the power supply rails are reliable. V_{CC} must remain within the data sheet recommended range, otherwise device functionality cannot be guaranteed.

The Power Controller can be configured using IPexpress™ as shown in Figure 8-3.

Figure 8-3. IPexpress Power Controller

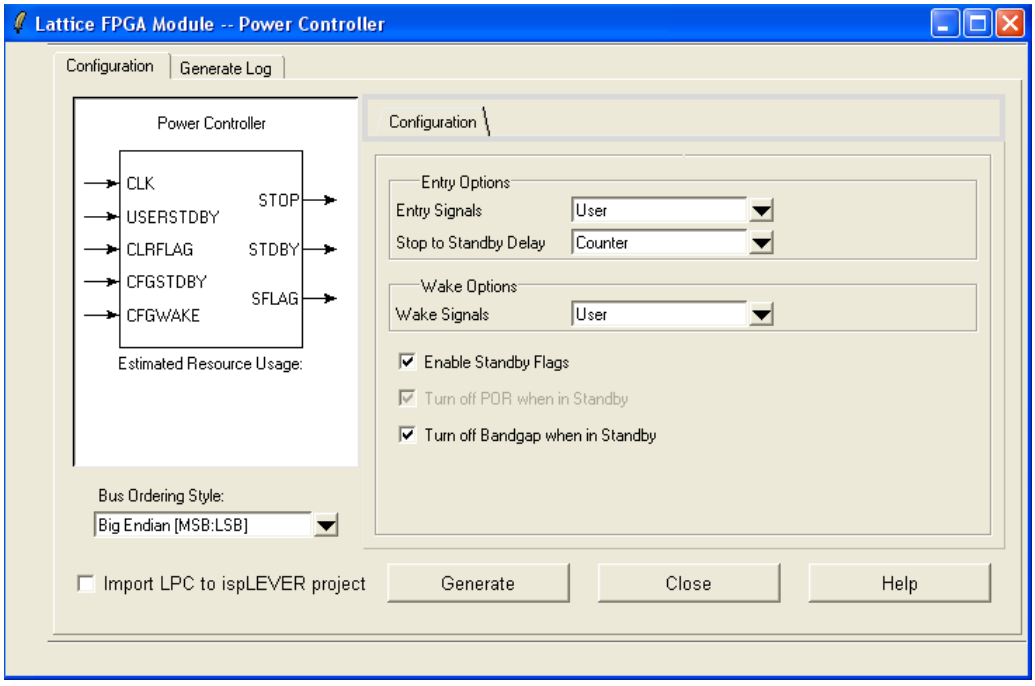


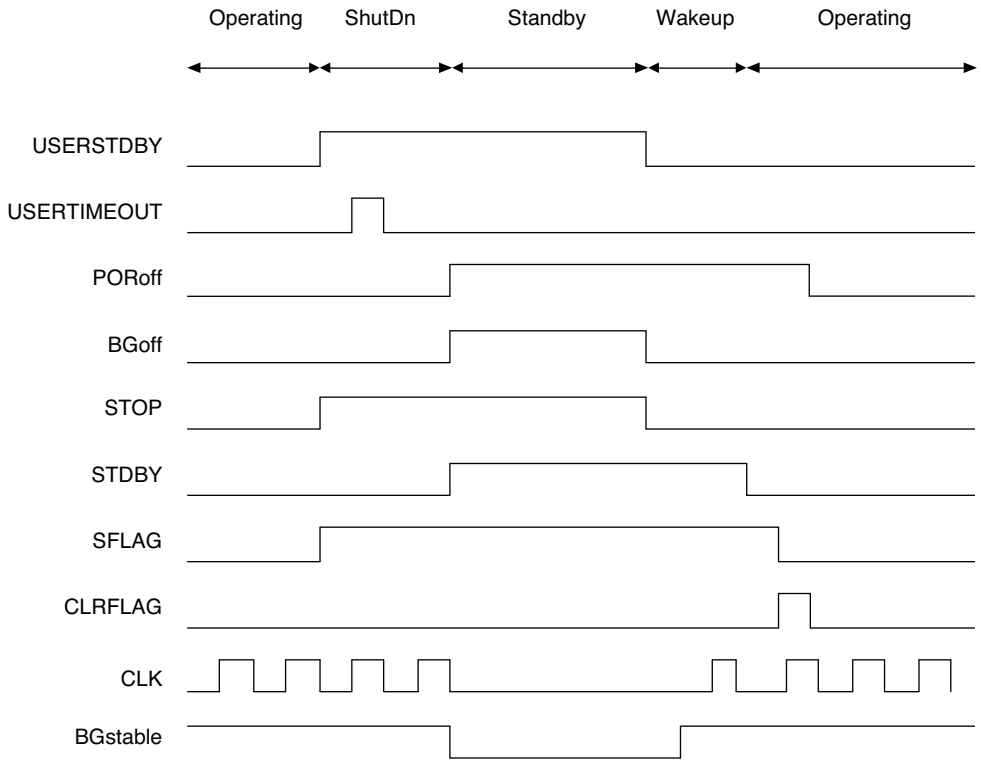
Table 8-3. IPexpress Power Controller Descriptions

Entry	Type	Values	Default Value	Comment
Entry Signals	Combo Box	User, Configuration	User	Configuration includes JTAG and I^2C
Stop to Standby Delay	Combo Box	User, Counter, Bypass	Bypass	User enables the USERTIMEOUT signal for the Timer Counter. When Bypass is selected there will be no delay between the STOP and STDBY signals.
Wake				
Wake Signals	Combo Box	User, Configuration	User	
Standby				
Enable Standby Flags	Check Box	TRUE, FALSE	TRUE	
Turn off Bandgap when in Standby[1]	Check Box	TRUE, FALSE	FALSE	When the bandgap circuitry is turned off, POR circuitry, analog circuits (PLL, oscillator, referenced LVCMOS I/Os, SSTL I/Os, HTSTL I/Os, and differential I/Os) are turned off.
Turn off POR when in Standby[1]	Check Box	TRUE, FALSE	FALSE	Power Detector circuitry is used determine if there has been a drop on V_{CC}. If there has been, then the device will be reconfigured.

1. When POR is shuf off, limited Power Detector circuitry is still active. This option is only recommended for applications where the power supply rails are reliable. VCC must remain within the data sheet recommended range, otherwise device functionality cannot be guaranteed.

The Power Controller sequence for entering and exiting standby is shown in Figure 8-4.

Figure 8-4. Power Controller Waveform



When designing with the Power Controller, users should be aware of the following:

- The software will produce an error if the Power Controller USRSTDBY signal is turned off during standby. This is to prevent a lock-up situation.
- When the Power Detector circuitry is turned off, there is still some limited circuitry within the Power Detector that is active. It is recommended to turn off the Power Detector circuitry only if the power supply rails are reliable. V_{CC} must remain within the data sheet recommended range or functionality cannot be guaranteed.
- When the Bandgap circuitry is turned off, Power Detector circuitry, analog circuits (PLL, oscillator, referenced LVCMOS I/Os, SSTL I/Os, HTSTL I/Os, and differential I/Os) are turned off.

Bank Controller

Referenced, differential and LVDS I/O standards consume more power then other I/O standards and are not always required to be active. The active high Bank Controller allows the designer to turn these I/Os off dynamically on a per-bank selection. The Dynamic InRD (input referenced and differential I/Os) is used to turn off referenced and differential inputs. Dynamic LVDS control is used to turn off the LVDS output driver. The Bank Controller can be instantiated using the primitives shown (BCINRD for dynamic InRD, BCLVDSO for dynamic LVDS) in Figure 8-5 or using IPexpress, as shown in Figure 8-6.

Figure 8-5. INRDB, LVDSOB Primitive

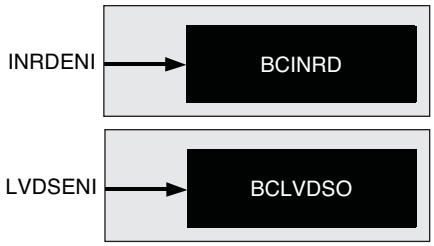
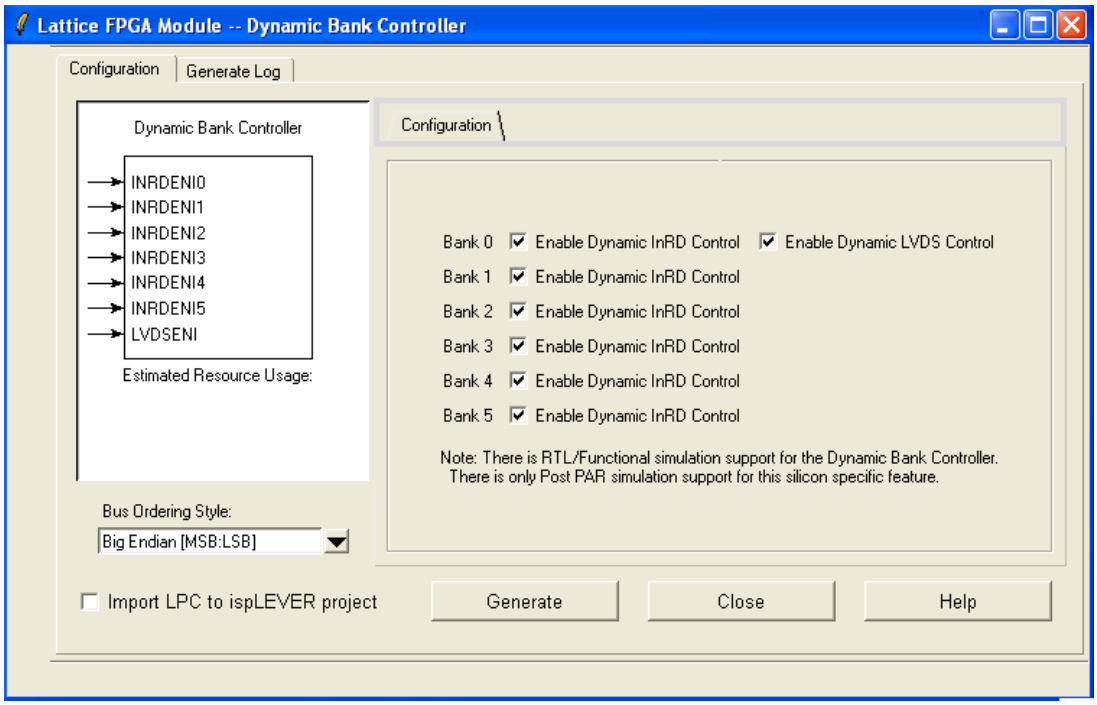


Figure 8-6. IPexpress Dynamic Bank Controller



When designing with the Bank Controller, users should be aware of the following:

- The software will produce an error if the Bank Controller control signal is from a referenced or differential I/O in the bank which is enabled by the Bank Controller. This is in order to prevent a lock-up situation.
- Powering off the bandgap overrides the Bank Controller

Power Guard

The Power Guard (PG) feature minimizes dynamic power consumption on routing by gating signals at the input pin. To prevent this loss, especially on large fan-out or heavily loaded nets like clocks, inputs can be “Power Guarded”. Power Guard prevents logic from getting on nets using an active high control signal. The Power Guard primitive component, as shown in Figure 8-7, can be included in your clock or data paths. For large buses, IPexpress can be used as shown in Figure 8-8.

Figure 8-7. Power Guard Primitive

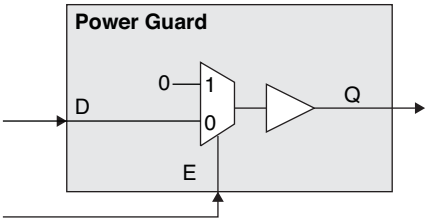
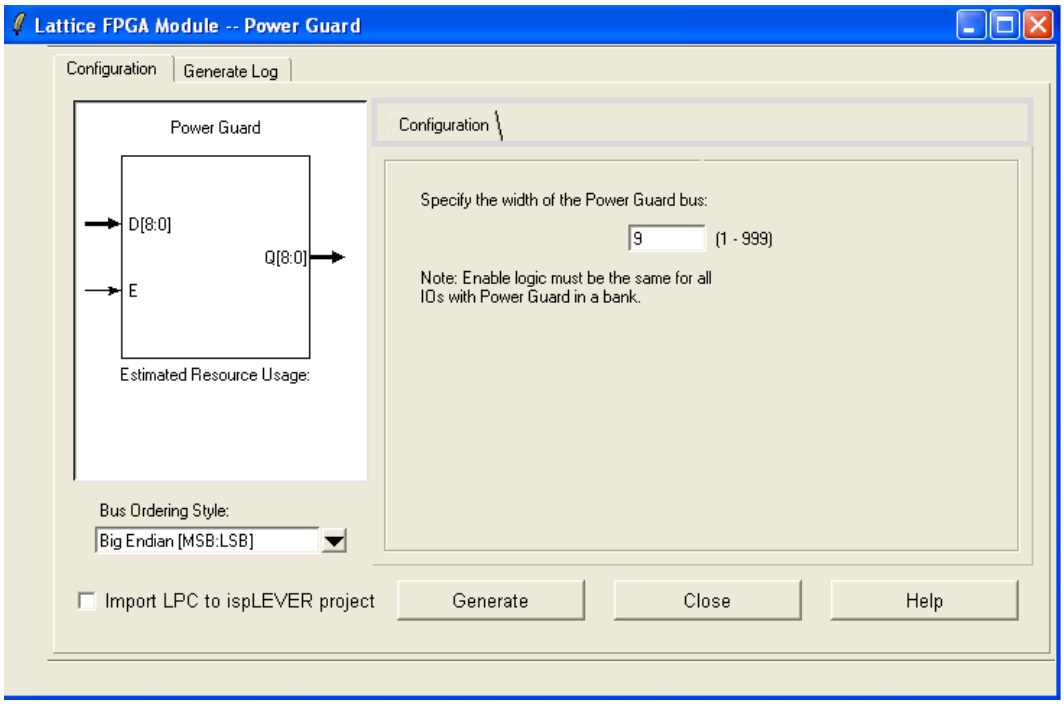


Figure 8-8. IPexpress Power Guard



When designing with Power Guard, users should be aware of the following:

- The software will produce an error if the Power Guard control signal is from the output of the Power Guard component. This is to prevent a lock-up situation.

Low Power Design Implementation

Figure 8-9 shows an example design implementation utilizing the MachXO2 low-power architectural components.

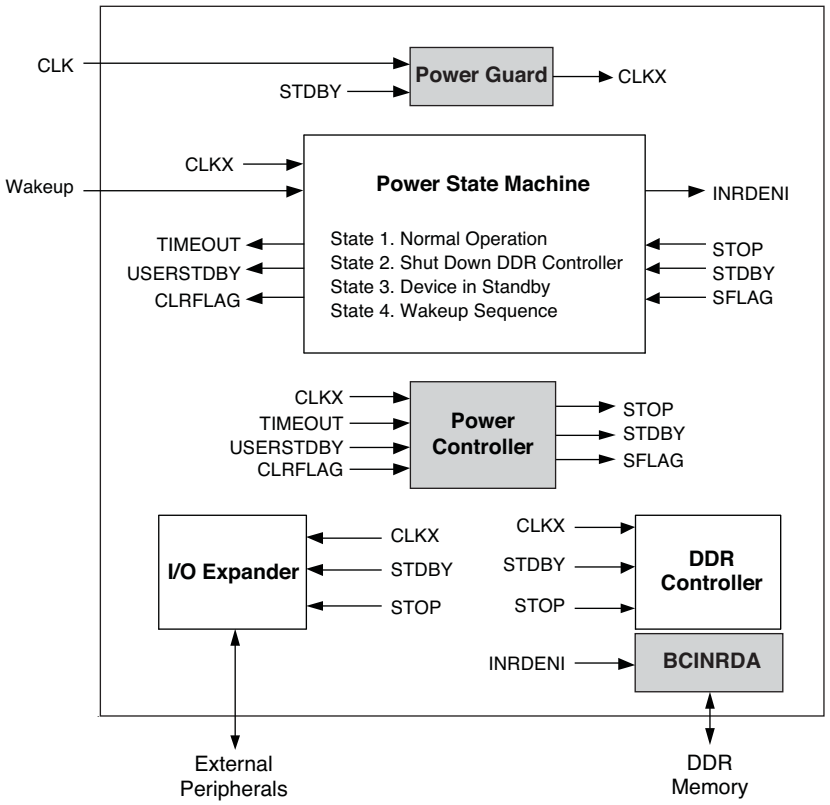
In this design, there are four power level states:

1. Normal operation
2. Normal operation except DDR memory controller turned off
3. Preparing for standby
4. Standby

In this example design:

- White blocks are normal FPGA logic.
- Gray blocks are MachXO2 components associated with low power consumption.
- The clock CLK goes through the Power Guard component because it is a high fan-net in the design. The STDBY signal is used to block the clock signal from getting onto the routing, thus reducing dynamic power.
- The signal INRDENI is used to turn off the referenced SSTL I/Os of the DDR interface using the Bank Controller BCINRDA, reducing static and dynamic power.
- A Power State Machine is used to step through the various states of the design. The bullets below describe what happens at each state:
  - Normal operation DDR Controller and I/O Expander are communicating with external memory and peripherals.
  - DDR memory is not being used and the INRDENI signal is asserted high. The referenced SSTL I/Os are disabled, thereby reducing static and dynamic power.
  - The device begins the low power standby sequence:
    - USERSDBY is asserted high and is used by the Power Controller to assert high STOP.
    - The DDR Controller and I/O Expander logic prepare for standby.
    - After the delay CLKX/256 from the Power Controller Timer Counter the STDBY signal is asserted high.
    - SFLAG is asserted high. Power Guard blocks the CLK signal from getting on high fan-out routing. Bank Controller turns off the referenced SSTL I/Os. DDR Controller and I/O Expander logic are in standby reaching the lowest power state.
  - When in the low power standby state the device waits for the signal Wakeup before resuming normal operation.

Figure 8-9. Low Power Design Implementation



Power Supply Sequencing and Hot Socketing

MachXO2 devices are designed to ensure predictable behavior during power-up and power-down. During power-up and power-down sequences, the I/Os remain in tri-state until the power supply voltage is high enough (VCC-MIN) to ensure reliable operation. In addition, leakage into I/O pins is controlled to within the limits specified in the MachXO2 Family Data Sheet, allowing for easy integration with the rest of the system.

Recommended Power-up Sequence

Refer to the DC and Switching Characteristics section of the MachXO2 Family Data Sheet for more information on power-up sequences for the MachXO2 family.

Power Calculator

The Power Calculator is a powerful tool that allows users to estimate the power consumption of a device. This tool offers an Estimation mode for “what-if” analysis, and enables designers to import NCD design files to accurately estimate power for their designs. The background engine performs each calculation quickly and accurately.

When running the Power Calculator tool in Estimation mode, designers provide estimates of the utilization of various components and the tool provides an estimate of the power consumption. This is a good start, especially for “what-if” analysis and device selection.

Calculation mode is a more accurate approach, where the designer imports the actual device utilization by importing the post Place and Route netlist design file (or NCD file).

Users can also import a Trace Report (or TWR) file where the frequencies for various clocks are also imported. Note that the Trace Report only includes frequencies of the clock nets that are constrained in the Preference file.

The default Activity Factor (AF%) for dynamic power calculation is set to 10% in the Power Calculator. Users can change the default AF for the entire project or for each clock net individually. Activity Factor is discussed in more detail later in this document.

Power Calculator Hardware Assumptions

The power consumption of a device can be broken down coarsely into the static (or DC) element and the dynamic (or AC) element. These elements have the following dependencies with respect to the junction temperature (T_J) of the die.

- Static power is a result of the leakage associated with the transistors. There are two types of static leakage.
  - Static leakage which has a strong temperature dependency
  - DC bias which is fairly constant across temperature
- Dynamic power is caused by the toggling of signals in the transistor.
  - Dynamic power is fairly constant across temperature

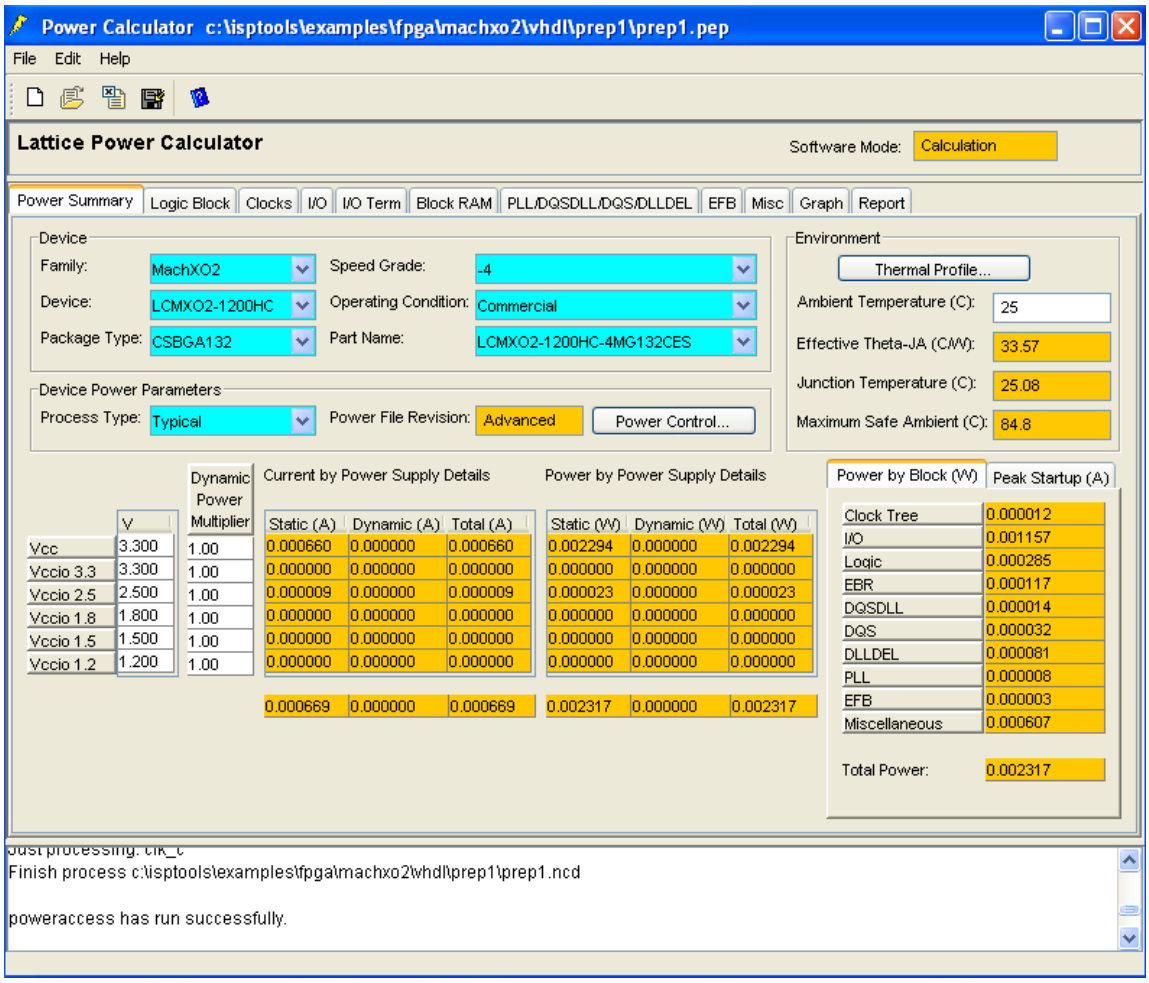
Each component of the device (e.g., LUT, register, EBR, I/O etc.) has its own coefficients for static and dynamic power. Certain selections in the Power Calculator tool affect some of these coefficients as discussed in the next section.

Power Calculator and Power Equations

Please refer to the ispLEVER Tutorial for launching and using the Power Calculator tool under **Help > ispLEVER Help**.

Once you step through the procedure, you will see the window illustrated in Figure 8-10.

Figure 8-10. Power Calculator Main Window



It is important to understand how the options available with the Power Calculator affect the power consumption of a device. For example, if the ambient temperature is changed, it affects the junction temperature, according to the following equation:

$$T_J = T_A + \Theta_{JA\_EFFECTIVE} * P \quad (1)$$

Where T_J and T_A are the junction and ambient temperatures, respectively, and P is the power.

$\Theta_{JA\_EFFECTIVE}$ is the effective thermal impedance between the die and its environment.

The junction temperature is directly proportional to the ambient temperature. An increase in T_A will increase T_J and result in an increase of the static leakage component.

Selecting the Process Type again affects the static leakage; in particular the static leakage coefficient changes.

The DC Bias component is constant across the range.

For dynamic power, increasing the frequency of toggling will increase the dynamic component of power.

Typical and Worst Case Process Power/I_{CC}

Another factor that affects DC power is process variation. This variation, in turn, causes variation in quiescent power.

Power Calculator takes these factors into account and allows designers to specify either a typical process or a worst case process.

Junction Temperature

Junction temperature is the temperature of the die during operation. It is one of the most important factors that affect the device power. For a fixed junction temperature, voltage and device package combination, quiescent power is fixed.

Ambient temperature affects the junction temperature as shown in Equation 1. Devices operating in a high-temperature environment have higher leakage since their junction temperature will be higher. Power Calculator models this ambient-to-junction temperature dependency. When the user provides an ambient temperature, it is rolled into an algorithm that calculates the junction temperature and power through an iterative process to find the thermal equilibrium of the system (device running with the design) with respect to its environment (T_A, airflow etc.).

Maximum Safe Ambient Temperature

Maximum Safe Ambient Temperature is one of the most important numbers displayed in the Summary tab of the Power Calculator. This is the maximum ambient temperature at which the design can run without violating the junction temperature limits for commercial or industrial devices.

Power Calculator uses an algorithm to accurately predict this temperature. The algorithm adjusts itself as the user changes options such as voltage, process, frequency, AF% etc. (or any factor that may affect the power dissipation of the device).

Operating Temperature Range

When designing a system, users must make sure that a device operates at specified temperatures within the system environment. This is particularly important to consider before a system is designed. With Power Calculator, users can predict device thermodynamics and estimate the dynamic power budget. The ability to estimate a device's operating temperature prior to board design also allows the designer to better plan for power budgeting and airflow.

Although total power, ambient temperature, thermal resistance and airflow all contribute to device thermodynamics, the junction temperature (as specified in the MachXO2 Family Data Sheet) is the key to device operation. The allowed junction temperature range is 0°C to 85°C for commercial grade devices and -40°C to 105°C for industrial grade devices. If the junction temperature of the die is not within these temperature ranges, the performance and reliability of the device's functionality cannot be guaranteed.

Dynamic Power Multiplier (DPM)

The user-defined frequency of operation makes this problem even more complex. To help resolve this issue, the Dynamic Power Multiplier provides some guard bands for system and board designers.

The Dynamic Power Multiplier is defaulted to "1" which means the dynamic power is what it is. If the user wishes to add 20% additional dynamic power, the DPM can be set to 1.2 (1 + 20%) and it can be placed against the appropriate power supply. This increases the dynamic power for that supply by 20% and provides designers with some guard band (if needed).

Power Budgeting

The Power Calculator provides the power dissipation of a design under a given set of conditions. It also predicts the junction temperature (T_J) for the design. If the junction temperature is outside the limits specified in the MachXO2 Family Data Sheet, the viability of operating the device at this junction temperature must be re-evaluated.

A commercial grade device is likely to show speed degradation with a junction temperature above 85°C and an industrial grade device at a junction temperature will degrade above 100°C. It is required that the die temperature be kept below these limits to achieve the guaranteed speed operation.

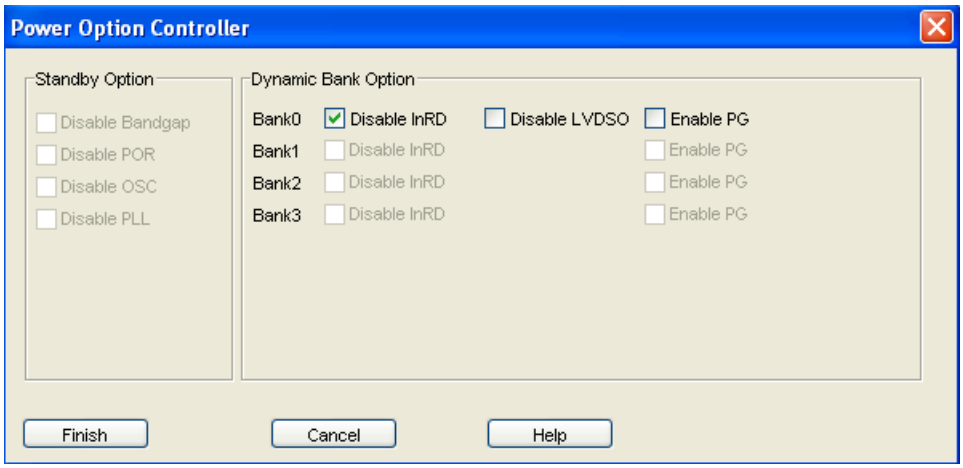
Operating a device at a higher temperature also means a higher Static I_{CC}. The difference between the Total I_{CC} and the Static I_{CC} (both Static I_{CC} and Dynamic I_{CC}) at a given temperature provides the Dynamic Icc budget available. If the device runs at a Dynamic I_{CC} higher than this budget, the total I_{CC} is also higher. This causes the die temperature to rise above the specified operating conditions.

The four factors of power, ambient temperature, thermal resistance and airflow, can also be varied and controlled to reduce the junction temperature of the device. The Power Calculator is a powerful tool to help system designers to properly budget the FPGA power that, in turn, helps improve the overall system reliability.

Dynamic Power Savings

The Power Calculator dynamically estimates the power when the Power Controller, Bank Controller and Power Guard are implemented in a design by simply enabling or disabling the components.

Figure 8-11. Dynamic Power Options



Activity Factor Calculation

The Activity Factor % (or AF%) is defined as the percentage of frequency (or time) that a signal is active or toggling the output. Most resources associated with a clock domain are running or toggling at some percentage of the frequency at which the clock is running. Users must provide this value as a percentage under the AF% column in the Power Calculator tool.

Another term for I/Os is the I/O Toggle Rate. The AF% is applicable to the PFU, Routing, and Memory Read Write Ports, etc. The activity of I/Os is determined by the signals provided by the user (in the case of inputs) or as an output of the design (in the case of outputs). The rates at which the I/Os toggle define their activity. The I/O Toggle Rate or the I/O Toggle Frequency is a better measure of their activity.

The Toggle Rate (or TR) in MHz of the output is defined in the following equation:

$$\text{Toggle Rate (MHz)} = 1/2 * f * AF\% \quad (2)$$

Users are required to provide the TR (MHz) value for the I/O instead of providing the frequency and AF% for other resources. AF can be calculated for each routing resource, output or PFU. However, this involves long calculations. The general recommendation for a design occupying roughly 30% to 70% of the device is an AF% between 15% and 25%. This is an average value. The accurate value of an AF depends upon clock frequency, stimulus to the design and the final output.

Thermal Impedance and Airflow

A common method for characterizing a packaged device's thermal performance is with Thermal Resistance, T. For a semiconductor device, thermal resistance indicates the steady state temperature rise of the die junction above a given reference for each watt of power (heat) dissipated at the die surface. Its units are °C/W.

The most common examples are Θ_{JA}, Thermal Resistance Junction-to-Ambient (in °C/W) and Θ_{JC}, Thermal Resistance Junction-to-Case (also in °C/W). Another factor is Θ_{JB}, Thermal Resistance Junction-to-Board (in °C/W).

Knowing the reference (i.e. ambient, case, or board) temperature, the power, and the relevant T value, the junction temperature can be calculated per the following equations.

$$T_J = T_A + \Theta_{JA} * P \quad (3)$$

$$T_J = T_C + \Theta_{JC} * P \quad (4)$$

$$T_J = T_B + \Theta_{JB} * P \quad (5)$$

Where T_J, T_A, T_C and T_B are the junction, ambient, case (or package) and board temperatures (in °C), respectively. P is the total power dissipation of the device.

Θ_{JA} is commonly used with natural and forced convection air-cooled systems. Θ_{JC} is useful when the package has a high conductivity case mounted directly to a PCB or heatsink. And Θ_{JB} applies when the board temperature adjacent to the package is known.

Power Calculator utilizes the ambient temperature (°C) to calculate the junction temperature (°C) based on the Θ_{JA} for the targeted device. Users can also provide the airflow values (in LFM) to obtain a more accurate junction temperature value.

To improve airflow effectiveness, it is important to maximize the amount of air that flows over the device or the surface area of the heat sink. The airflow around the device can be increased by providing an additional fan or increasing the output of the existing fan. If this is not possible, baffling the airflow to direct it across the device may help. This means the addition of sheet metal or objects to provide the mechanical airflow guides to guide air to the target device. Often the addition of simple baffles can eliminate the need for an extra fan. In addition, the order in which air passes over devices can impact the amount of heat dissipated.

Reducing Power Consumption

One of the most critical challenges for designers today is reducing the system power consumption. A low-order reduction in power consumption goes a long way, especially in modern hand-held devices and electronics. There are several design techniques that can be used to significantly reduce overall system power consumption. Some of these include:

- Using the MachXO2 power saving architecture features like Power Controller, Bank Controller and Power Guard.
- Reducing operating voltage while staying within data sheet limits.
- Operating within the specified package temperature limitations.
- PLL jitter/power option within IPexpress
- Confirm if input-only bank saves power
- Using optimum clock frequency reduces power consumption, as the dynamic power is directly proportional to the frequency of operation. Designers must determine if some portions of the design can be clocked at a lower rate that will reduce power.

- Reducing the span of the design across the device. A more closely-placed design uses fewer routing resources and therefore less power.
- Reducing the voltage swing of the I/Os where possible.
- Ensuring input logic levels are not left floating but pulled either up or down.
- Ensuring no I/O pull-up/down conflicts with other components on the board.
- Using optimum encoding where possible. For example, a 16-bit binary counter has, on average, only 12% activity factor and a 7-bit binary counter has an average of 28% activity factor. On the other hand, a 7-bit LFSR counter will toggle at an activity factor of 50%, which causes higher power consumption. A gray code counter, where only one bit changes at each clock edge, will use the least amount of power, as the activity factor is less than 10%.
- Minimizing the operating temperature by the following methods:
  - Use packages that can better dissipate heat, such as ceramic packages.
  - Place heat sinks and thermal planes around the device on the PCB.
  - Use better airflow techniques, such as mechanical airflow guides and fans (both system fans and device mounted fans).
- To achieve the lowest standby power:
  - All clocks and combinatorial logic should be held at a steady state
  - All inputs should be held at a rail; if not possible, toggling inputs should gated using Power Guard
  - All outputs should be tri-stated and the Bank Controller should turn off referenced and LVDS outputs
  - Internal oscillator should be turned off using the STDBY port
  - PLLs should be turned off using the STDBY port
  - Bandgap and POR should be turned off using the Power Controller

Power Calculator Assumptions

The following are the assumptions made by the Power Calculator.

- The Power Calculator tool uses equations with constants based on a room temperature of 25°C. The default temperature of 25°C can be changed.
- Users can define the ambient temperature (T_A) for device junction temperature (T_J) calculation based on the power estimation. T_J is calculated from the user-entered T_A and the power calculation of typical room temperature.
- I/O power consumption is based on an output loading of 5 pF. Designers have the ability to change this capacitive loading.
- Users can estimate power dissipation and current for each type of power supply (V_{CC}, V_{CCIO}).
- The nominal V_{CC} is used by default to calculate power consumption. A lower or higher V_{CC} can be chosen from a list of available values.
- Θ_{JA} can be changed to better estimate the operating system manually or by entering Airflow in Linear Feet per Minute (LFM) along with a Heat Sink options.
- The default value of the I/O types for MachXO2 devices is LVCMOS25, 8 mA.
- The activity factor (AF) is defined as the toggle rate of the registered output. For example, assuming that the input of a flip-flop is changing at every clock cycle, 100% AF of a flip-flop running at 100 MHz is 50 MHz. The default activity factor for logic is 10%.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
April 2011	01.1	Updated Reducing Power Consumption list.
February 2012	01.2	Updated document with new corporate logo.
		Document status changed from Advance to Final.
October 2012	01.3	IPexpress Power Controller Descriptions table – updated Values and Default Values columns for the Entry Signals and Wake Signals.
December 2012	01.4	Corrected formatting in the description of the Power Controller block diagram.

Using User Flash Memory and Hardened Control Functions in MachXO2 Devices

October 2012 Technical Note TN1205

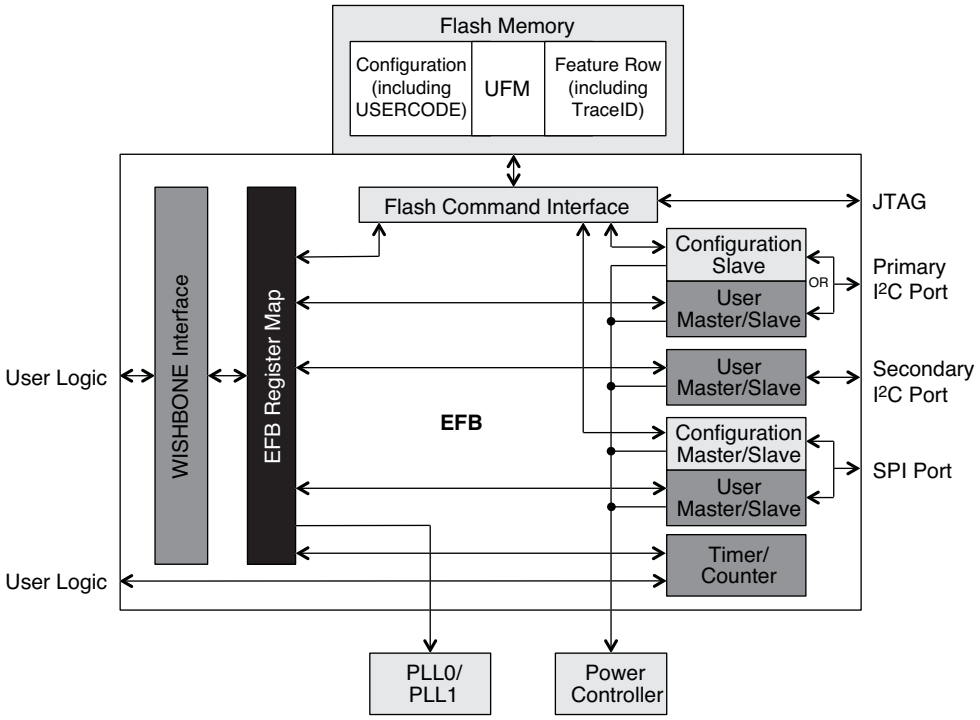
Introduction

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 I^2C 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 I^2C and SPI

Figure 9-1 shows the EFB architecture and the WISHBONE interface to the FPGA user logic.

Figure 9-1. Embedded Function Block (EFB)



The hard SPI, I^2C, Timer/Counter IPs contained in the EFB can save in excess of 500 LUTs when compared to implementing these same functions in FPGA logic using Lattice reference designs.

The EFB Register Map is used to access the EFB hardened functions through the Slave WISHBONE bus. Each hard IP has dedicated 8-bit Data and Control registers, with the exception of the Flash Memory (UFM/Configuration), which is accessed through the same set of registers. Ports having access to the EFB Register Map have access to all registers. As an example from the Primary I^2C Slave port you could access the Timer/Counter registers. The EFB Register Map is shown below:

Table 9-1. EFB Memory Map

Address Range (Hex)	8-bit Data/Control Registers Function
0x00-0x1F	PLL0 Dynamic Access[1]
0x20-0x3F	PLL1 Dynamic Access[1]
0x40-0x49	I^2C Primary
0x4A-0x53	I^2C Secondary
0x54-0x5D	SPI
0x5E-0x6F	Timer/Counter
0x70-0x75	Flash Memory (UFM/Configuration)
0x76-0x77	EFB Interrupt Source

1. 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 TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide, for details on PLL configuration registers and recommended usage.

The EFB module is represented in design software as a primitive and it is described in this document. Use IPexpress™ to configure the EFB, and to generate Verilog or VHDL source code. The source code is instantiated in your design.

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, as well as connectivity between the individual EFB functional blocks. The User Logic must include a WISHBONE Master interface to communicate with the WISHBONE Slave interface of the EFB. An example of a WISHBONE Master is the LatticeMico8™.

The block diagram in Figure 9-2 shows the WISHBONE bus signals between the FPGA core and the EFB. Table 9-2 provides a detailed definition of the signals.

Figure 9-2. WISHBONE Bus Interface Between the FPGA Core and the EFB Module

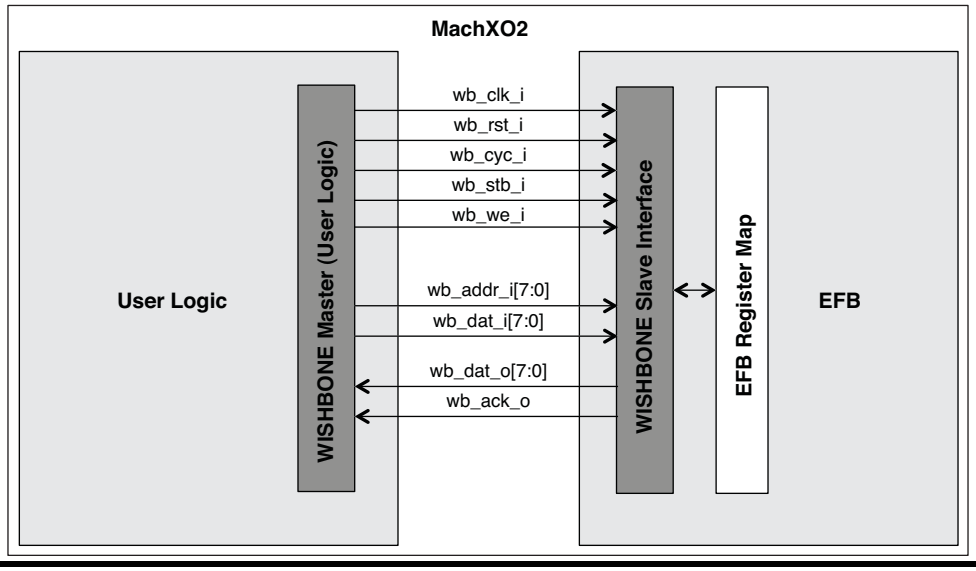


Table 9-2. WISHBONE Slave Interface Signals of the EFB Module

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.
wb_rst_i	Input	1	Synchronous reset signal that resets the WISHBONE interface logic. This signal does not affect the contents of any registers. It terminates an active bus cycle. Wait 1us after de-assertion before starting any subsequent WISHBONE transactions.
wb_cyc_i	Input	1	Asserted by the WISHBONE master, indicates a valid bus cycle is present on the bus.
wb_stb_i	Input	1	Strobe 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 selects an EFB specific register.
wb_dat_i	Input	8	A WISHBONE Master writes data to the addressed EFB register using the wb_dat_i bus during write cycles.
wb_dat_o	Output	8	A WISHBONE Mater receives data from the addressed EFB register using wb_dat_o during read memory cycles.
wb_ack_o	Output	1	Signals the WISHBONE Master the bus cycle is complete; data written to the EFB is accepted. Data read from the EFB is valid.

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.

WISHBONE Protocol

For information on the WISHBONE protocol and command sequences read the WISHBONE section of TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

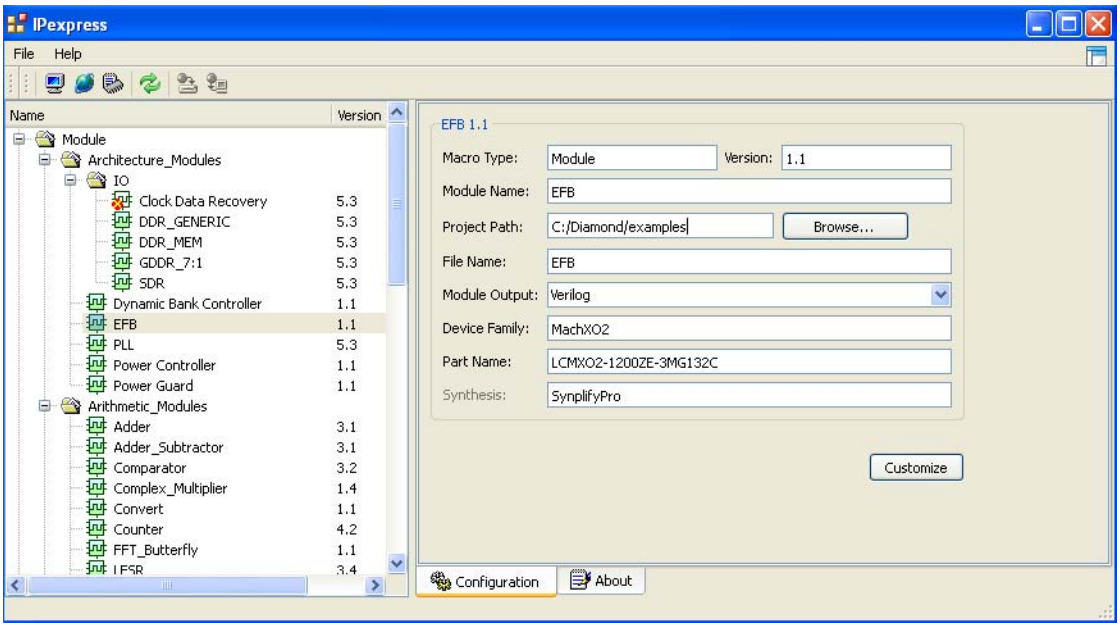
WISHBONE Design Tips

1. Take note when dynamically turning off components for power savings; the EFB requires the MachXO2 internal oscillator to be enabled even if it is not the source of the WISHBONE Clock.
2. If the EFB WISHBONE input signals are not used they should be connected to ‘0’.
3. For more information on the WISHBONE spec can be found on OpenCores website.
4. Many Lattice reference designs have a WISHBONE bus (www.latticesemi.com/products/intellectualproperty/aboutreferencedesigns.cfm)

Generating an EFB Module with IPexpress

IPexpress is used to configure the EFB hard IP functions and generate the EFB module. From the Lattice Diamond® top menu select **Tools > IPexpress**. With a MachXO2 device targeted for the Diamond project, the IPexpress window opens and the EFB module can be found under **Modules > Architecture Modules**.

Figure 9-3. EFB Module in IPexpress

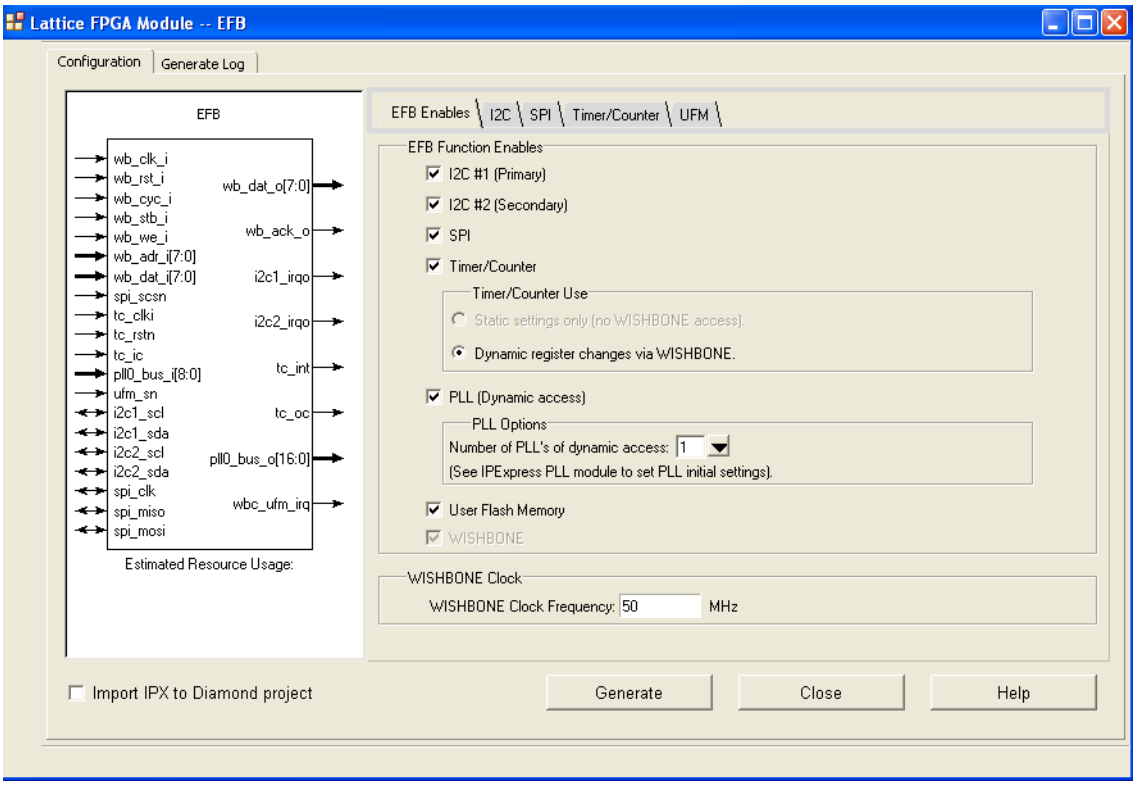


Fill in the Project Path, File Name, and Design Entry fields, and click **Customize**.

After clicking on the Customize button, the EFB configuration dialog appears. The left side of the EFB window displays a graphical representation of the I/O associated with each IP function. The I/O pins appear and disappear as each IP is enabled or disabled. The initial tab is used to enable the hardened functions, the dynamic access to the PLL configuration settings, the User Flash Memory (UFM) and enter the WISHBONE Clock Frequency. An example EFB with all features enabled is shown in Figure 9-4.

The hardened IP functions and the UFM have individual tabs in the EFB window for individual configuration settings. These tabs will be discussed later in the document, with the technical description of the specific functions. When all functions have been configured, click on the **Generate** button and the EFB module will be generated and ready to be instantiated in your design.

Figure 9-4. Generating an EFB Module with IPexpress



The number of available PLL modules depends on the device density and this will be reflected in the IPexpress EFB GUI. The MachXO2-256 and MachXO2-640 do not have PLL modules and so the PLL checkbox in the EFB window is not available for selection. The MachXO2-640U, MachXO2-1200, MachXO2-1200U, and MachXO2-2000 each have one PLL, and the MachXO2-2000U, MachXO2-4000 and MachXO2-7000 each have two PLLs available for dynamic access through the EFB WISHBONE Slave interface.

The default WISHBONE Clock Frequency is set to 50 MHz. Designers can enter a clock frequency up to 133 MHz. The WISHBONE clock is used by the EFB WISHBONE interface registers and also by the SPI and I^2C hardened IP cores. The EFB requires the MachXO2 internal oscillator to be enabled even if it is not the source of the WISHBONE Clock.

Like other modules, the EFB settings can be viewed in the Map Report as shown below:

```
Embedded Functional Block Connection Summary:
-----------------------------------------------

   Desired WISHBONE clock frequency: 2.0 MHz
   Clock source:                     clk
   Reset source:                     wb_rst
   Functions mode:
      I2C #1 (Primary) Function:     ENABLED
      I2C #2 (Secondary) Function:   DISABLED
      SPI Function:                  ENABLED
      Timer/Counter Function:        DISABLED
      Timer/Counter Mode:            WB
      UFM Connection:                DISABLED
      PLL0 Connection:               DISABLED
      PLL1 Connection:               DISABLED
```

```
I2C Function Summary:
--------------------
   I2C Component:          PRIMARY
   I2C Addressing:         7BIT
   I2C Performance:        100kHz
   Slave Address:          0b0001001

   General Call:           ENABLED
   I2C Wake Up:            DISABLED
   I2C Component:          UFM/Configuration
   I2C Addressing:         7BIT
   I2C Performance:        100kHz
   Slave Address:          0b0001000
SPI Function Summary:
--------------------
   SPI Mode:               BOTH
   SPI Data Order:         LSB to MSB
   SPI Clock Inversion:    DISABLED
   SPI Phase Adjust:       DISABLED
   SPI Wakeup:             DISABLED
Timer/Counter Function Summary:
------------------------------
   None
UFM Function Summary:
--------------------
   UFM Utilization:        EBR Initialization
   Available General
   Purpose Flash Memory:   511 Pages (511*128 Bits)

   EBR Blocks with Unique
   Initialization Data:    6

   WID            EBR Instance
   ---            ------------
   0b0000000011   LCDCharMap_inst/LCDCharMap_0_0_0
   0b0000000100   STRING_TABLE_INST/EXT_ROM_INST/pmi_romXhmenusdn8101024_0_0_0
   0b0000000101
  lm8_inst/u1_isp8/u1_isp8_prom/pmi_romXhprom_initadn18112048_1_1_0
   0b0000000110
  lm8_inst/u1_isp8/u1_isp8_prom/pmi_romXhprom_initadn18112048_0_0_3
   0b0000000111
  lm8_inst/u1_isp8/u1_isp8_prom/pmi_romXhprom_initadn18112048_0_1_2
   0b0000001000
  lm8_inst/u1_isp8/u1_isp8_prom/pmi_romXhprom_initadn18112048_1_0_1
```

Hardened I²C IP Cores

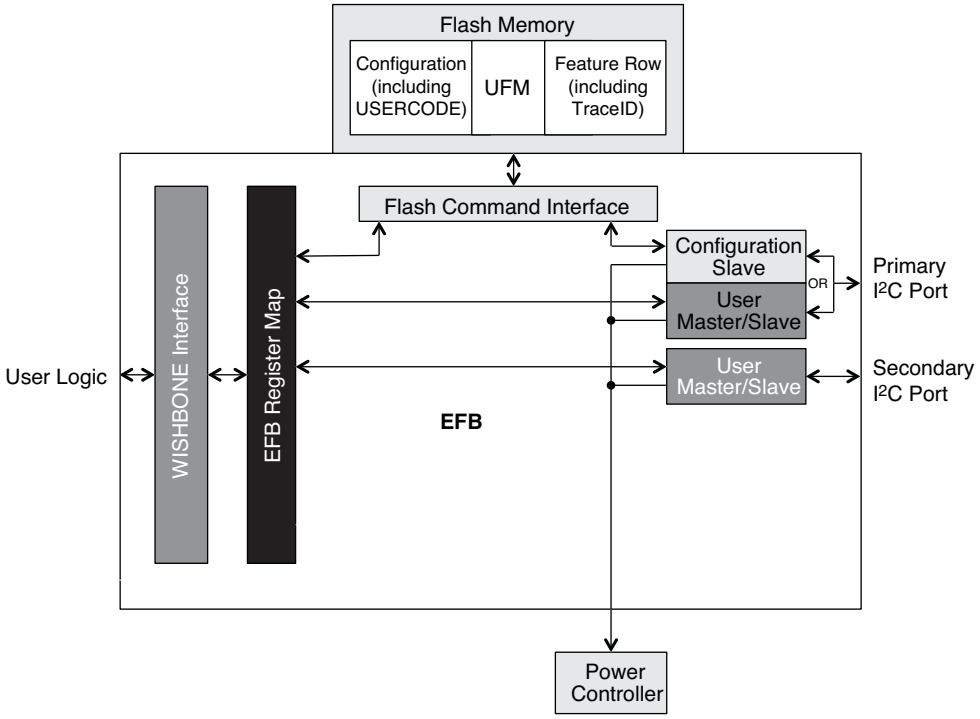
I²C is a widely used two-wire serial bus for communication between devices on the same board. Every MachXO2 device contains two hardened I²C IP cores designated as the “Primary” and “Secondary” I²C IP cores. The two cores in the MachXO2 can operate as an I²C Master or as an I²C Slave. The difference between the two cores is that the Primary core has pre-assigned I/O pins while the ports of the secondary core can be assigned to any general purpose I/O. In addition, the Primary core also has access to the Flash Memory (UFM/Configuration) through the Flash Command Interface. The hardened I²C IP core functionality and block diagram are shown below.

Table 9-3. Hardened I²C Functionality

	Primary I²C Configuration	Primary I²C User	Secondary I²C User
I²C Port as Master	No	Yes	Yes
I²C Port as Slave	Yes[1]	Yes[1]	Yes
Access the Flash Memory (UFM/Configuration)	Yes[1]	No	No
Access the User Logic	No	Yes	Yes
Must use dedicated I/Os	Yes	Yes	No
Wake Power Controller from Standby Mode	Yes	Yes	Yes
Enter Power Controller Standby Mode	Yes	No	No

1. Primary port can be used as Configuration/UFM port or as a User port, but not both.

Figure 9-5. I²C Block Diagram



When an EFB I²C core is a Master it can control other devices on the I²C bus through the physical interface. When an EFB I²C core is the Slave, the device can provide I/O expansion to an I²C Master. Both MachXO2 Primary and Secondary cores support the following I²C functionality:

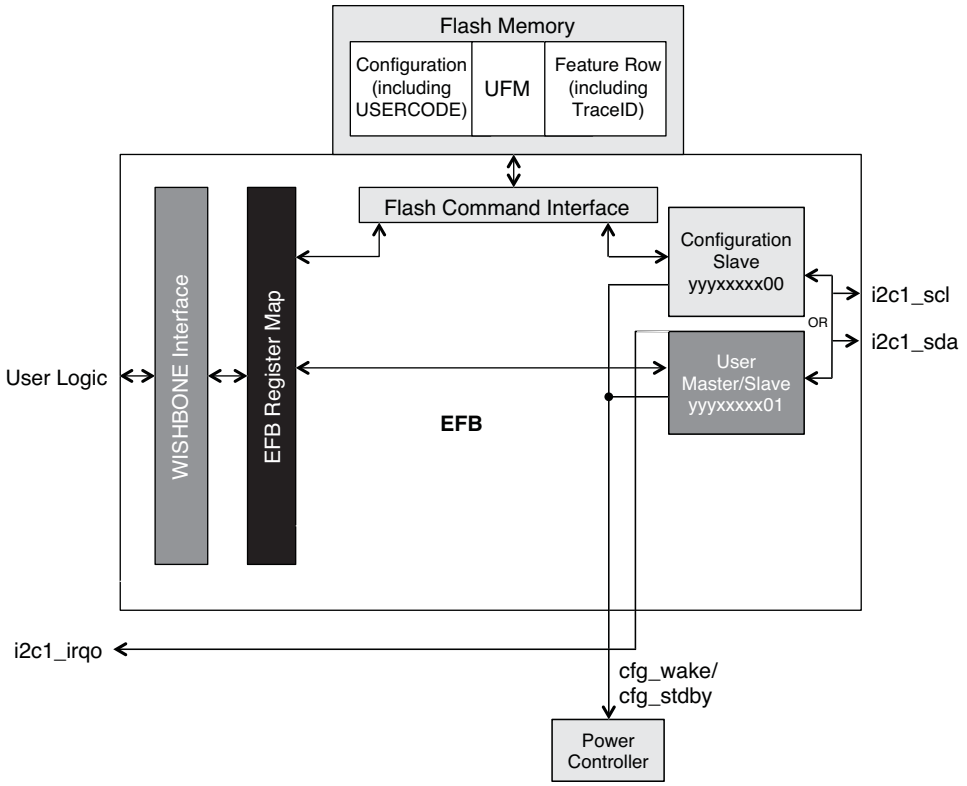
- Master/Slave mode support
- 7-bit and 10-bit addressing
- Clock stretching (I²C Slave is allowed to hold down or stretch the clock to reduce the bus speed)
- Supports 50KHz, 100KHz, and 400KHz data transfer speed
- General Call support (addresses all devices on the bus using the I²C address 0)
- Interface to User Logic through the EFB WISHBONE Slave interface

Primary I²C

The MachXO2 Primary I²C Controller is shown in Figure 9-6. The main functions of the Primary Controller are:

- Either:
  - I²C Configuration Slave provides access to the Flash Memory (UFM/Configuration); or
  - I²C User Slave provides access to the User Logic
- I²C Configuration or User Slave provides access to the MachXO2 Power Controller
- I²C User Master provides access to peripherals attached to the MachXO2

Figure 9-6. I²C Primary Block Diagram



The Primary I²C core can be used for accessing the User Flash Memory (UFM) and for programming the Configuration Flash. However, the Primary I²C port cannot be used for both UFM/Configuration access and user functions in the same design. The block diagram in Figure 9-6 shows an interface between the I²C block and the Flash Memory (UFM/Configuration). For information on Programming the MachXO2 through I²C port reference, the I²C section of TN1204, MachXO2 Programming and Configuration Usage Guide.

Slave I²C peripherals on a bus are accessed by the Master I²C calling the Slave's unique addresses. The Primary Configuration address is "yyyxxxxx00" and the Primary User address is "yyyxxxxx01" where "y" and "x" are user programmable from IPexpress.

The Primary Configuration I²C can be used to wake the Power Controller from Standby or enter Standby. The Primary User can only be used to wake the Power Controller from Standby mode. For more information on the Power Controller, reference TN1198, Power Estimation and Management for MachXO2 Devices. The I²C Power Controller features can be set up through IPexpress as documented later and the register settings are defined in TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

Table 9-4 documents the IP signals of the Primary I²C cores.

Table 9-4. I²C Primary – IP Signals

Signal Name	Pre-Assigned Pin Name	I/O	Width	Description
i2c1_scl	SCL	Bi-directional	1	**Open drain clock line of the I²C core** – The signal is an output if the I²C core is performing a Master operation. The signal is an input for Slave operations. This signal MUST be routed directly to the corresponding Pre-Assigned Pin. Please reference the MachXO2 Family Data Sheet pin tables for detailed pad and pin locations of I²C ports in each MachXO2 device.
i2c1_sda	SDA	Bi-directional	1	**Open drain data line of the I²C core** – The signal is an output when data is transmitted from the I²C core. The signal is an input when data is received into the I²C core. This signal MUST be routed directly to the corresponding Pre-Assigned Pin. Please reference the MachXO2 Family Data Sheet pin tables for detailed pad and pin locations of I²C ports in each MachXO2 device.
i2c1_irqo	—	Output	1	**Interrupt request output signal of the I²C core** – The intended use of this signal is for it to be connected to a WISHBONE Master controller (i.e. a microcontroller or state machine) and request an interrupt when a specific condition is met. These conditions are described with the I²C section of TN1246.
cfg_wake	—	Output	1	**Wake-up signal** – Hardwired signal to be connected ONLY to the Power Controller of the MachXO2 device for functional simulation support. The signal is enabled only if the Wakeup Enable feature has been set within the EFB GUI, I²C Tab.
cfg_stdby	—	Output	1	**Stand-by signal** – Hardwired signal to be connected ONLY to the Power Controller of the MachXO2 device for functional simulation support. The signal is enabled only if the Wakeup Enable feature has been set within the EFB GUI, I²C Tab.

Secondary I²C

The Secondary I²C controller in the MachXO2 provides the same functionality as the Primary I²C controller with the exception of access to the MachXO2 Configuration Logic. The i2c2_scl and i2c2_sda ports are routed through the general purpose routing of the FPGA fabric and designers can assign them to any General Purpose I/O (GPIO).

The Secondary I²C can be used to wake the Power Controller from Standby mode. For more information on the Power Controller, reference TN1198, Power Estimation and Management for MachXO2 Devices. The I²C Power Controller features can be setup through IPexpress as documented later and the register settings are defined in TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

Slave I²C peripherals on a bus are accessed by the User Master I²C calling the Slave's unique addresses. The Secondary User address is "yyyxxxxx10" where "y" and "x" are user-programmable from IPexpress.

Figure 9-7 shows the block diagram of the Secondary I²C core.

Figure 9-7. I²C Secondary Block Diagram

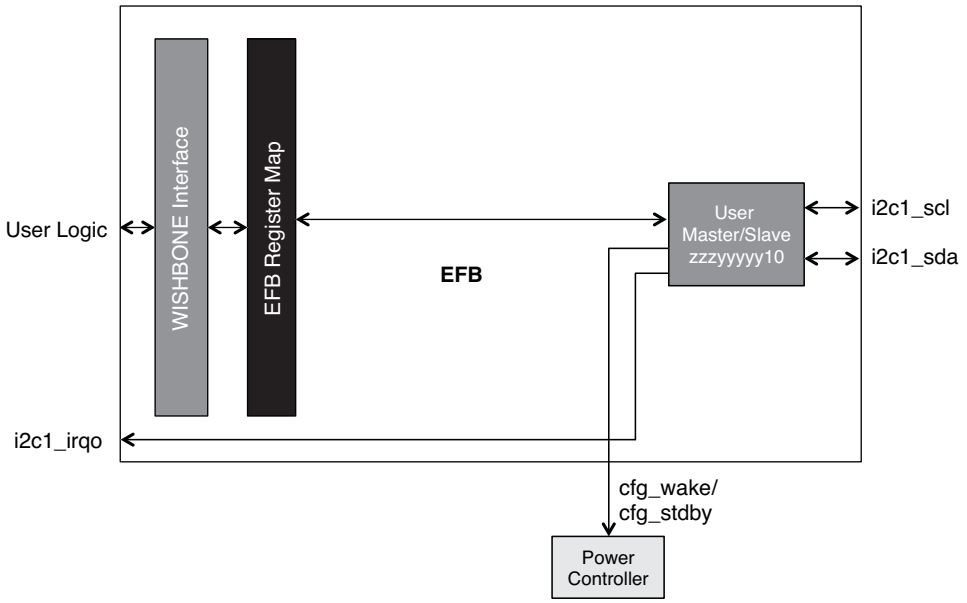


Table 9-5 documents the IP signals of the Secondary I²C cores. These signals can be routed to any GPIO of the MachXO2 devices.

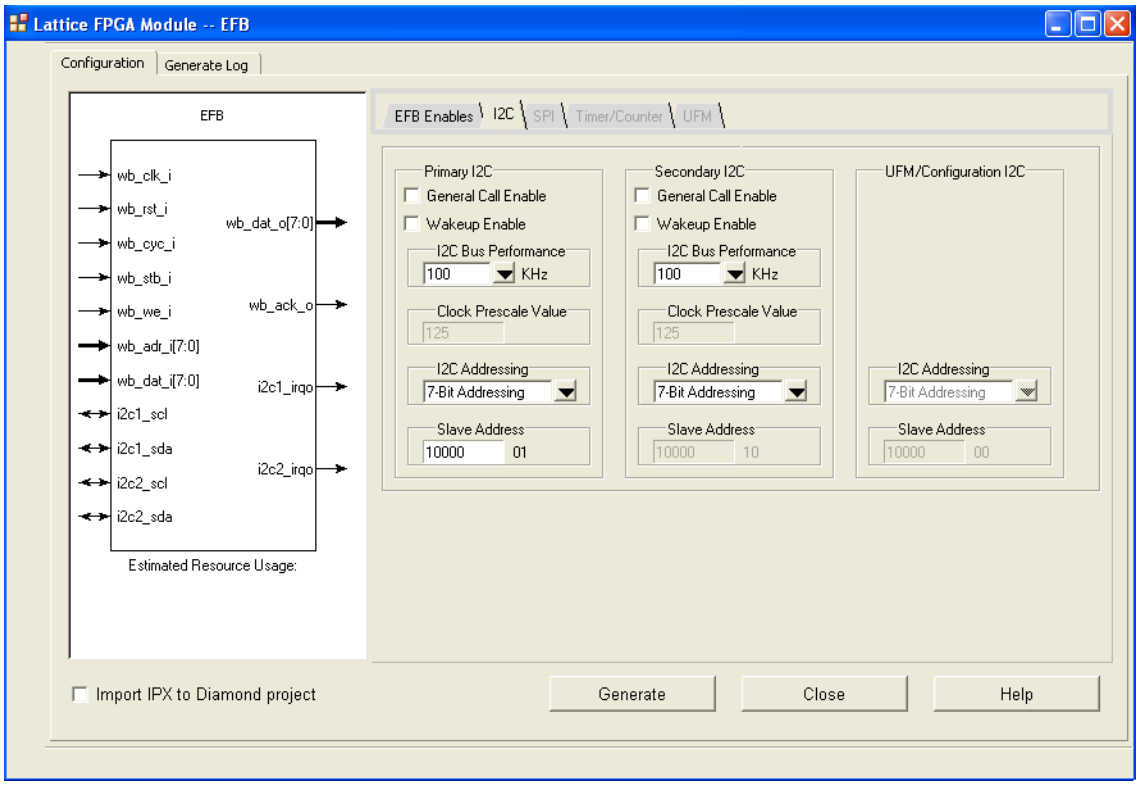
Table 9-5. I²C Secondary – IP Signals

Signal Name	Pre-assigned Pin Name	I/O	Width	Description
i2c2_scl	—	Bi-directional	1	**Open drain clock line of the I²C core**. The signal is an output if the I²C core is performing a Master operation. The signal is an input for Slave operations. The signal can be routed to any GPIO of the MachXO2.
i2c2_sda	—	Bi-directional	1	**Open drain data line of the I²C core**. The signal is an output when data is transmitted from the I²C core. The signal is an input when data is received into the I²C core. The signal can be routed to any GPIO of the MachXO2.
i2c2_irqo	—	Output	1	**Interrupt request output signal of the I²C core**. This signal is intended to be connected to a WISHBONE master controller (i.e. a microcontroller or state machine) and to request an interrupt when a specific condition is met. These conditions are described with the I²C register definitions.
cfg_wake	—	Output	1	**Wake-up signal** – Hardwired signal to be connected ONLY to the Power Controller of the MachXO2 device for functional simulation support. The signal is enabled only if the Wakeup Enable feature has been set within the EFB GUI, I²C Tab.
cfg_stdby	—	Output	1	**Stand-by signal** – Hardwired signal to be connected ONLY to the Power Controller of the MachXO2 device for functional simulation support. The signal is enabled only if the Wakeup Enable feature has been set within the EFB GUI, I²C Tab.

Configuring I²C Cores with IPexpress

Designers can configure the I²C cores and generate the EFB module with IPexpress. Selecting the I²C tab in the EFB GUI will display the configurable settings of the I²C cores. Figure 9-8 shows an example where the I²C cores are configured for an example design.

Figure 9-8. Configuring the I²C Functions of the EFB Module with IPexpress



General Call Enable

This setting enables the I²C General Call response (addresses all devices on the bus using the I²C address 0) in Slave mode. This setting can be modified dynamically by enabling the GCEN bit in the Primary register I2C_1_CR or the Secondary register I2C_2_CR.

Wake-up Enable

For more information on the Power Controller reference TN1198, Power Estimation and Management for MachXO2 Devices.

When the Wake-up Enable is selected an external I²C Master can cause the MachXO2 to leave the Standby Power state. There are two methods an external I²C master can use to wake the MachXO2 device:

- Primary or Secondary Slave I²C EFB address match
- Perform a General Call followed by the 0xF3 hex command opcode

The WKUPEN bit in the I2C_1_CR or the I2C_2_CR can be modified dynamically allowing the Wake Up function to be enabled or disabled.

I²C Bus Performance

Designers can select an I²C frequency of 50KHz, 100KHz, or 400KHz. This will be the frequency of the SCL clock on the I²C bus. This GUI value, together with the WISHBONE Clock Frequency attribute from the EFB Enables tab, allows the software to calculate the clock divider value for the 10-bit pre-scale registers using the equation (WB Clock)/(Clock Pre-scale Value). This pre-scale value is modified dynamically by accessing the Primary I²C Baud Rate register pair I2C_1_BR1, I2C_1_BR0 or the Secondary I²C Baud Rate register pair I2C_2_BR1, I2C_2_BR0.

I²C Addressing

Designers can select between a 7-bit or 10-bit I²C Slave addressing scheme. The last two bits of the 7-bit address and 10-bit address are hard-coded and select one of the I²C components. The programmable bits of the I²C address are shared between I²C modules and defined as:

yyyxxxxxww

ww bits are hard coded with the following definition

00 = Primary Configuration Flash Memory (UFM/Configuration) I²C

01 = Primary User I²C

10 = Secondary User I²C

11 = I²C core reset

xxxxx bits are programmable using the IPexpress GUI and have the default value of 10000

yyy bits are programmable when 10-bit addressing is selected and have the default value of 000

The Primary I²C address is the same as the length (seven or ten bits) as the Flash Memory (UFM/Configuration) I²C address. The Primary and Secondary I²C address sizes can be of differing lengths. For example the Primary I²C address could be ten bits and the Secondary I²C address could be seven bits.

MachXO2 I²C Usage Cases

The I²C usage cases described below refer to Figure 9-9.

1. Master MachXO2 I²C Accessing Slave External I²C Devices
   a. A WISHBONE bus Master is implemented in the MachXO2 logic
   b. I²C devices 1, 2, and 3 are all Slave devices
   c. The WISHBONE bus Master performs bus transactions to the Primary I²C controller in the EFB to access external Slave I²C Device 1 on Bus A
   d. The WISHBONE bus Mater performs bus transactions to the Secondary I²C controller in the EFB to access the external Slave I²C Devices number 2 or 3 on Bus B
   e. For information on the I²C register definitions and command sequences reference the I²C section of TN1246, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide
2. External Master I²C Device Accessing Slave MachXO2 I2C
   a. The I²C devices 1, 2, and 3 are I²C Master devices
   b. The external master I²C Device 1 on Bus A performs I²C memory cycles to access the EFB Primary I²C controller using address yyyxxxxx01.
   c. The external master I²C Device 2 or 3 on Bus B performs I²C memory cycles to access the EFB Secondary I²C User with the address yyyxxxxx10
   d. A WISHBONE bus master in the MachXO2 fabric must manage data reception and transmission. The WISHBONE master can use interrupts or polling techniques to manage data transfer, and to prevent data overrun conditions.
      i. For information on the I²C register definitions and command sequences reference I²C section of TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices
3. External Master I²C Device Accessing the MachXO2 Flash Memory Using the Primary I²C Interface
   a. The external Master I²C Device 1 on Bus A performs bus transactions using address yyyxxxxx00. The external master interacts with the MachXO2 Configuration Logic using this address. The Configuration Logic provides the controls necessary for performing Flash memory operations.

b. More details on the accessing the Flash Memory (UFM/Configuration) of the MachXO2 device through I²C will be found later in this document and in TN1246, MachXO2 EFB Reference Guide.

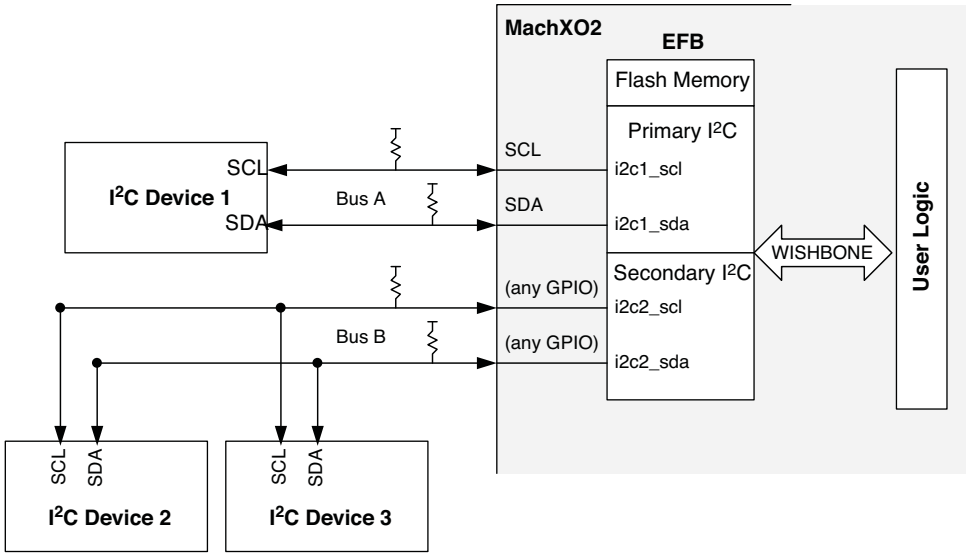
c. For information on Programming the MachXO2 through I²C port reference, the I²C section of TN1204, MachXO2 Programming and Configuration Usage Guide.

4. The above usage cases are not mutually exclusive. For example:

a. External Master Device 1 on Bus A can access the MachXO2 Configuration Logic at the same time a WISHBONE Master transfers data to the I²C slave devices on Bus B.

b. A WISHBONE master can transfer data to a microprocessor on Bus A (i.e. I²C Device 1), and at some future time the microprocessor can send data back to the WISHBONE Master. The microprocessor may also at some future time reprogram the MachXO2 Flash memory in order to update the MachXO2 device's functionality.

Figure 9-9. I²C Circuit



I²C Design Tips

1. For information on the I²C register definitions and command sequences, reference the I²C section of TN1246, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide.
2. Take note when dynamically turning off components for power savings; the EFB requires the MachXO2 internal oscillator to be enabled even if it is not the source of the WISHBONE Clock.
3. I²C has lower priority than JTAG Port and the Slave SPI Port when accessing the Flash Memory (UFM/Configuration). Reference "Flash Memory (UFM/Configuration) Access" on page 29 for details.
4. The Primary I²C port cannot be used for both UFM/Configuration access and user functions in the same design.
5. If the secondary I²C Secondary Port is enabled after issuing a Refresh command or toggling PROGRAMN it is recommended to reset the state machine with a I²C STOP. I²C STOP is performed with a single register write 0x40 to I2C_2_CMDR. This will cause a short low-pulse on SCK as the block signals the STOP. Normal I²C activity can be commenced without additional delay.
6. There are a number of I²C reference designs on the Lattice website (www.latticesemi.com/products/intellectualproperty/aboutreferencedesigns.cfm) including:
   a. RD1124: I²C Slave Peripheral Using Embedded Function Block
   b. UG55: MachXO2 Hardened I²C Master/Slave Demo
7. For information on the I²C Protocol reference www.i2c-bus.org/

Hardened SPI IP Core

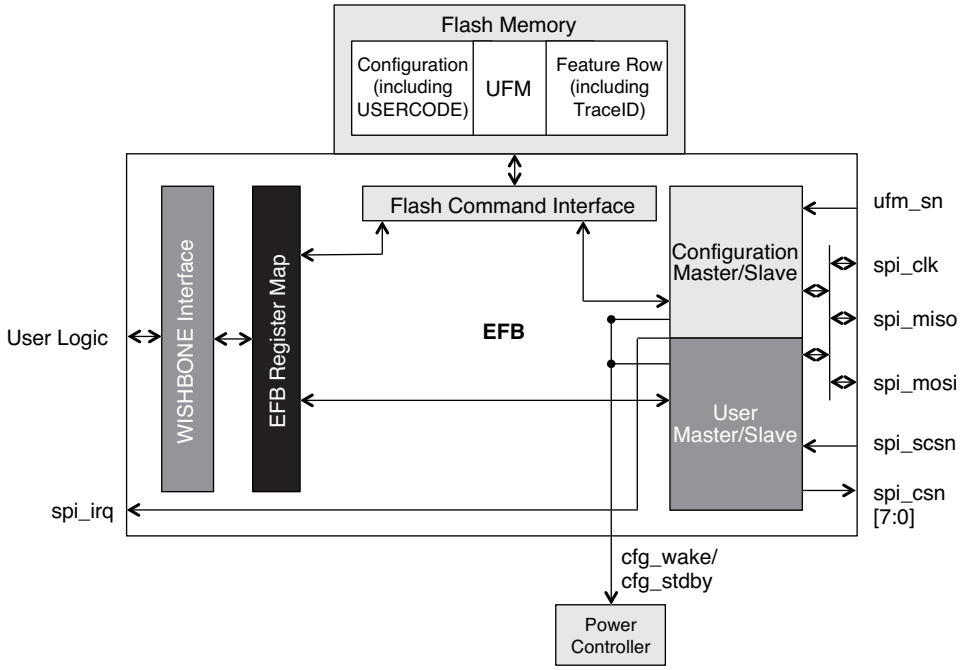
SPI is a widely used four-wire serial bus which operates in full duplex for communication between devices. The MachXO2 EFB contains a SPI Controller that can be configured as a SPI Master/Slave or a SPI Slave. When the IP core is configured as a Master/Slave it is able to control up to either other devices with Slave SPI interfaces. When the core is configured as a Slave, it is able to interface to an external SPI Master device. The SPI core interfaces with the MachXO2's Configuration Logic or the other User Logic. The hardened SPI IP core functionality and block diagram are shown below.

Table 9-6. Hardened SPI Functionality

	Configuration SPI	User SPI
Slave SPI Port	Yes	Yes
Master SPI Port	No[1]	Yes
Access the Flash Memory (UFM/Configuration)	Yes	No
Must use dedicated I/Os	Yes	Yes[2]
Wake Power Controller from Standby Mode	Yes	Yes
Enter Power Controller Standby Mode	No	Yes

1. Only for the configuring SRAM.
2. Any GPIO can be used for spi_csn[7:1] and spi_scsn.

Figure 9-10. SPI Block Diagram



The SPI IP core on MachXO2 devices supports the following functions:

- Configurable Master and Slave modes
- Mode Fault error flag with CPU interrupt capability
- Double-buffered data register for increased throughput
- Serial clock with programmable polarity and phase
- LSB First or MSB First Data Transfer
- Interface to custom logic through the EFB WISHBONE slave interface

In Master/Slave SPI mode:

- The User SPI controller has eight available Master Chip Selects (spi_csn[7:0]) ports. This allows the control of up to eight external devices with Slave SPI interface.
- The Configuration SPI upon power-up, if the SPI port has been enabled to boot the MachXO2 device from an external Slave SPI Flash memory, then the SPI port will act as a Master SPI controller and spi_csn[0] is used as a Master Chip Select for selecting a specific SPI Flash memory. For information on Programming the MachXO2 through the SPI port, reference the SPI section of TN1204, MachXO2 Programming and Configuration Usage Guide.

In Slave SPI mode:

- The User SPI core has one Slave Chip Select (spi_scsn) pin. This allows the User SPI core to be selected by an external device with a Master SPI interface. User Logic is access through the EFB WISHBONE interface by a WISHBONE Master in the FPGA logic.
- The Configuration SPI has one Slave Chip Select (ufm_sn) pin. An external SPI Master can access the MachXO2's Configuration Logic by asserting the chip select input. The external SPI Master can reprogram the MachXO2's Configuration Flash and User Flash Memory by performing bus transfers with SN asserted.

This usage guide is focused on the User SPI access. For more information on Programming the MachXO2 through SPI port reference, the SPI section of TN1204, MachXO2 Programming and Configuration Usage Guide.

The Slave Configuration SPI port can be used to wake the Power Controller from Standby or enter Standby. The Slave User SPI port can only be used to wake the Power Controller from Standby mode. For more information on the Power Controller, reference TN1198, Power Estimation and Management for MachXO2 Devices. The SPI Power Controller features can be set up through IPexpress as documented later and the register settings are defined in TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

SPI Interface Signals

The SPI interface uses a serial transmission protocol. Data is transmitted serially (shifted out from the transmitting device) and it is received serially, shifted into the receiving device. The master device selects a specific slave device by asserting a chip select, enabling the slave device to shift in the commands/data and to respond by shifting out data.

Table 9-7 documents the signals that are generated with the IP core. Each signal has a description of the usage and how it should be connected in a design project.

Table 9-7. SPI – IP Signals

Signal Name	Pre-assigned Pin Name	I/O	Width	Description
spi_clk	MCLK/CCLK	Bi-directional	1	The signal is an output if the SPI core is in Master mode (MCLK). The signal is an input if the SPI core is in Slave mode (CCLK). This signal MUST be routed directly to the corresponding Pre-Assigned Pin. Reference the MachXO2 Family Data Sheet pin tables for detailed pad and pin locations of SPI signals in each MachXO2 device.
spi_miso	SPISO/SO	Bi-directional	1	The signal is an input if the SPI core is in Master mode (SPISO). The signal is an output if the SPI core is in Slave mode (SO). This signal MUST be routed directly to the corresponding Pre-Assigned Pin. Reference the MachXO2 Family Data Sheet pin tables for detailed pad and pin locations of SPI signals in each MachXO2 device.

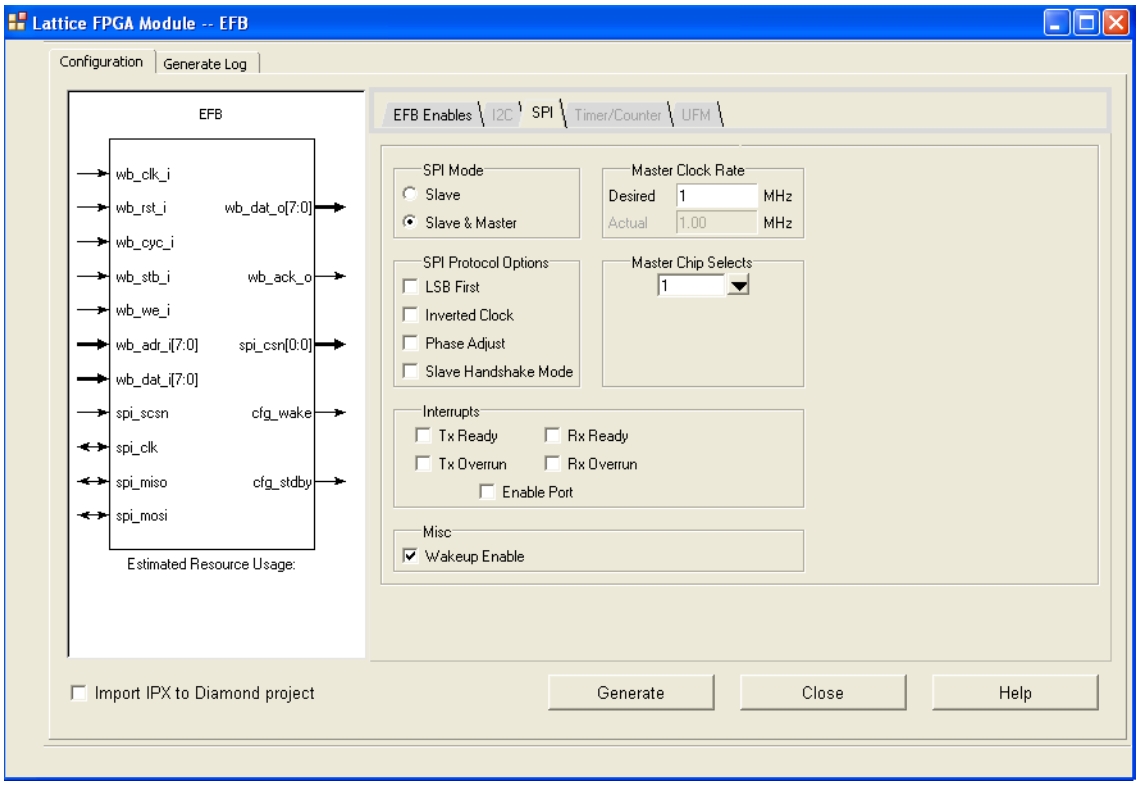
Table 9-7. SPI – IP Signals (Continued)

Signal Name	Pre-assigned Pin Name	I/O	Width	Description
spi_mosi	SISPI/SI	Bi-directional	1	The signal is an output if the SPI core is in Master mode (SISPI). The signal is an input if the SPI core is in Slave mode (SI). This signal MUST be routed directly to the corresponding Pre-Assigned Pin. Reference the MachXO2 Family Data Sheet pin tables for detailed pad and pin locations of SPI signals in each MachXO2 device.
spi_csn[7:0]	CSSPIN	Output	8	Master Chip Select (Active Low). Up to eight independent slave SPI devices can be accessed using the MachXO2 SPI Controller when it is in Master SPI mode. The signal spi_csn[0] MUST be routed directly to the corresponding Pre-Assigned Pin. Reference the MachXO2 Family Data Sheet tables for detailed pad and pin locations of SPI signals in each MachXO2 device.
spi_scsn	—	Input	1	User Slave Chip Select (Active Low). An external SPI Master controller asserts this signal to transfer data to/from the SPI Controllers Transmit Data/Receive Data registers. The signal can be routed to any GPIO of MachXO2 device.
ufm_sn	SN	Input	1	Configuration Logic Chip select (Active Low) is dedicated for selecting the Flash Memory UFM and Configuration Sectors. This signal MUST be routed directly to the corresponding Pre-Assigned Pin. Reference the MachXO2 Family Data Sheet pin tables for detailed pad and pin locations of SPI signals in each MachXO2 device. SN is an active pin whenever the SPI core is instantiated, whether 'ufm_sn' appears on the EFB primitive or not. Thus, SN cannot be recovered as user I/O. SN can be tied high externally to augment the weak internal pull-up if not connected to an external Master SPI bus. SN is also active in a blank or erased device.
spi_irq	—	Output	1	Interrupt request output signal of the SPI core. This signal is intended to be connected to a WISHBONE master controller (i.e. a microcontroller or state machine). It is asserted when specific conditions are met. These conditions controlled using the SPI register settings.
cfg_wake	—	Output	1	Wakeup signal – to be connected only to the Power Controller module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, SPI Tab.
cfg_stdby	—	Output	1	Stand-by signal – to be connected only to the Power Controller module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, SPI Tab.

Configuring the SPI Core with IPexpress

IPexpress is used to configure the SPI Controller and to generate Verilog or VHDL source code for inclusion in your design. Selecting the SPI tab, in the EFB GUI, displays the configurable settings for the SPI core. Figure 9-11 shows an example SPI Controller configuration.

Figure 9-11. Configuring the SPI Functions of the EFB Module with IPexpress



SPI Mode

This option allows the user to select between “Slave”, or “Slave & Master” modes for the initial mode of the SPI core. Selecting “Slave & Master” enables SPI Master settings which include Master Clock Rate and Master Chip Selects. This option can be updated dynamically by modifying the MSTR bit of the register SPICR2.

SPI Master Clock Rate

Desired Frequency: The EFB SPI Controller, when it is configured as a SPI Master, provides an output clock to the SPI Slave devices on the bus. The output frequency uses a “power of two” value to divide the WISHBONE Clock Frequency. The SPI Master uses the Master Clock Rate to time all SPI bus transactions and internal operations. The MachXO2 SPI Master interface can operate at speeds up to 45 MHz. You input the WISHBONE Clock Frequency on the EFB Enables tab of the dialog.

The divisor can be changed while the FPGA is in user mode. Updating the divider value in the SPIBR register causes the SPI Controller to reset and use a new output clock frequency.

Actual Frequency: It is not always possible to divide the input WISHBONE clock exactly to the frequency requested by the user. The actual frequency value will be returned in this read-only field. When both the desired SPI clock and WISHBONE clock fields have valid data and either is updated, this field will return the value rounded to two decimal places.

SPI Protocol Options

LSB First: This setting specifies the order of the serial shift of a byte of data. The data order (MSB or LSB first) is programmable within the SPI core. This option can be updated dynamically by modifying the LSBF bit in the register SPICR2.

Inverted Clock: The inverts the clock polarity used to sample and output data is programmable for the SPI core. When selected the edge changes from the rising to the falling clock edge. This option can be updated dynamically by accessing the CPOL bit of register SPICR2.

Phase Adjust: An alternate clock-data relationship is available for SPI devices with particular requirements. This option allows the user to specify a phase change to match the application. This option can be updated dynamically by accessing the CPHA bit in the register SPICR2.

Slave Handshake Mode: Enables Lattice proprietary extension to the SPI protocol. For use when the internal support circuit (e.g. 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. This option can be updated dynamically by accessing the SDBRE bit in the register SPICR2.

Master Chip Selects

The SPI Controller provides the ability to provide up to eight individual chip select outputs for master operation. Each slave SPI device accessed by the master must have their own dedicated chip select. This option can be updated dynamically by modifying the register SPICSR.

SPI Controller Interrupts

TX Ready: An interrupt which indicates the SPI transmit data register (SPITXDR) is empty. The interrupt bit is IRQTRDY of the register SPIIRQ. When enabled, indicates TRDY was asserted. Write a '1' to this bit to clear the interrupt. This option can be change dynamically by modifying the bit IRQTRDYEN in the register SPICSR.

RX Ready: An interrupt which indicates the receive data register (SPIRXDR) contains valid receive data. The interrupt is bit IRQRRDY of the register SPIIRQ. When enabled, indicates RRDY was asserted. Write a '1' to this bit to clear the interrupt. This option can be change dynamically by modifying the bit IRQRRDYEN in the register SPICSR.

TX Overrun: An interrupt which indicates the Slave SPI chip select (SPI_SCSN) was driven low while a SPI Master. The interrupt is bit IRQMDF of the register SPIIRQ. When enabled, indicates MDF (Mode Fault) was asserted. Write a '1' to this bit to clear the interrupt. This option can be change dynamically by modifying the bit IRQMDFEN in the register SPICSR.

RX Overrun: An interrupt which indicates SPIRXDR received new data before the previous data. The interrupt is bit IRQROE of the register SPIIRQ. When enabled, indicates ROE was asserted. Write a '1' to this bit to clear the interrupt. This option can be change dynamically by modifying the bit IRQROEEN in the register SPICSR.

Enable Port (Interrupts): This enables the interrupt request output signal (spi_irq_ of the SPI core signal. This signal is intended to be connected to a WISHBONE master controller (i.e. a microcontroller or state machine) and to request an interrupt when a specific condition is met.

Wake-up Enable

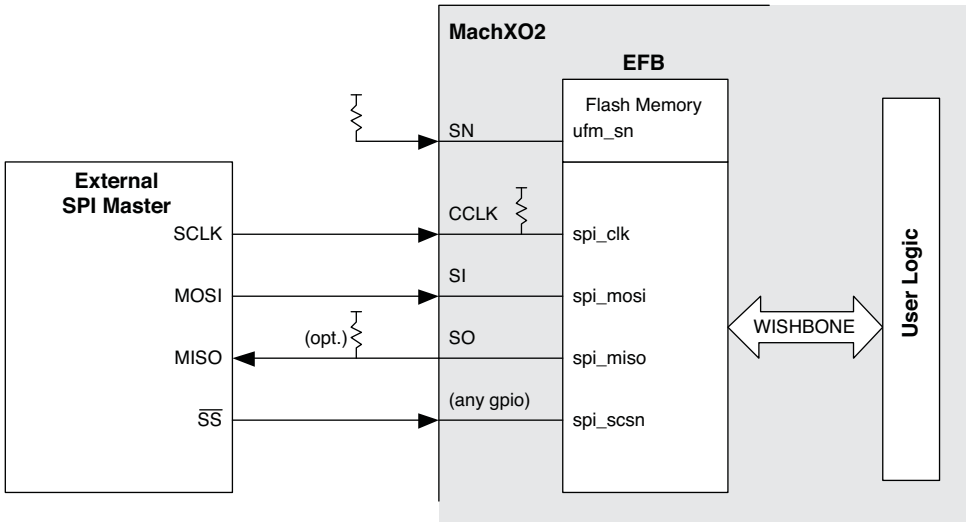
Enables the SPI core to send a wake-up signal to the Power Controller to wake the part from standby mode when the User Slave SPI chip select (spi_csn[0]) is driven low. This option can be updated dynamically by modifying the bit WKUPEN_USER in the register SPICR1.

MachXO2 SPI Usage Cases

The SPI usage cases described below refer to the figures below:

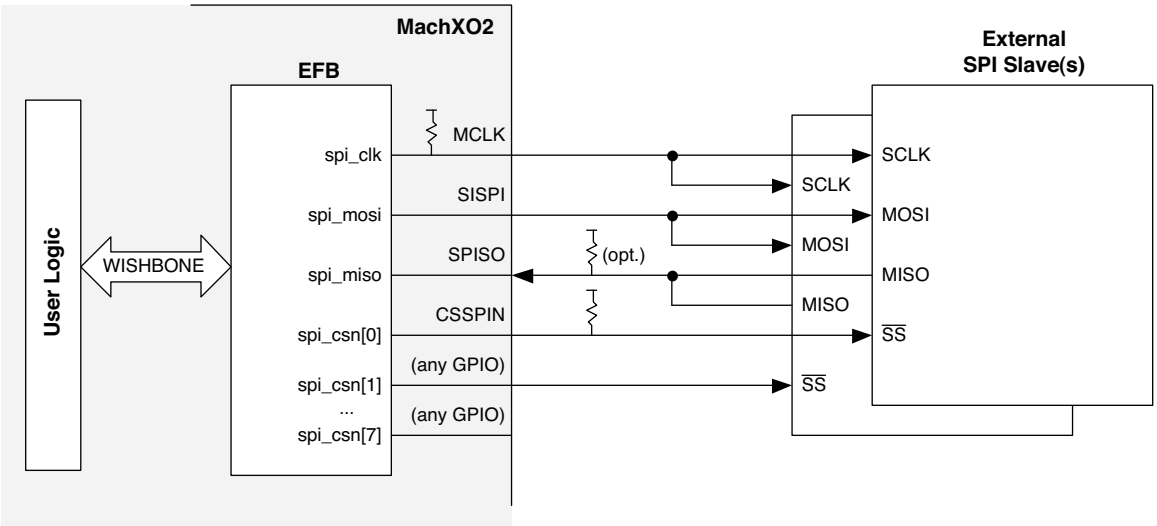
1. External Master SPI Device Accessing the Slave MachXO2 User SPI
   a. The External Master SPI is connected to the MachXO2 using the dedicated SI, SO, CCLK pins. The spi_scsn is placed on any Generic I/O. The EFB SPI Mode is set to Slave only.
   b. A WISHBONE Master controller is implemented in the MachXO2 general purpose logic array. The master controller monitors the availability to transmit or receive data by polling the SPI status registers, or by responding to interrupts generated by the SPI Controller.
      i. For information on the SPI register definitions and command sequences reference SPI section of TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.
   c. The external SPI Master does not have access to the MachXO2 Configuration Logic because the SN that selects the Configuration Logic is pulled high.

Figure 9-12. External Master SPI Device Accessing the Slave MachXO2 User SPI



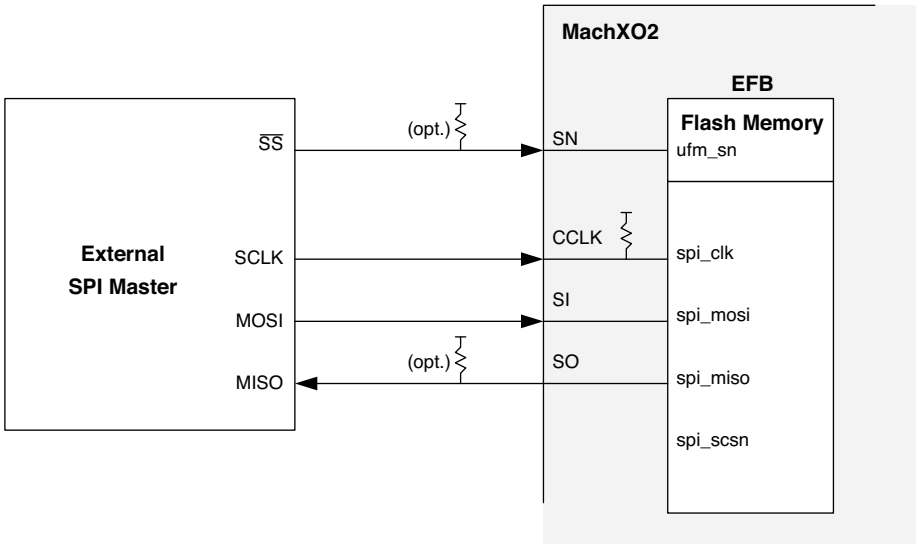
2. MachXO2 User SPI Master accessing one or multiple External Slave SPI devices
   a. The MachXO2 SPI Master is connected to External SPI Slave devices using the dedicated SPI port pins. The Chip Selects are configured as follows:
      i. The MachXO2 SPI Master Chip Select spi_scn[0] is placed on the dedicated CSSPIN and connected to the External Slave Chip Select.
      ii. The MachXO2 SPI Master Chip Select spi_scn[1] is placed on any I/O and connected to another External Slave Chip Select.
      iii. Up to eight External Slave SPIs can be connected using spi_scn[7:0]
   b. A WISHBONE Master controller is implemented in the MachXO2 general logic. It controls transfers to the slave SPI devices. It can use a polling method, or it can use SPI Controller interrupts to manage transfer and reception of data.
      i. For information on the SPI register definitions and command sequences reference SPI section of TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

Figure 9-13. MachXO2 User SPI Master Accessing One or Multiple External Slave SPI Devices



3. External Master SPI Device accessing the MachXO2 Configuration Logic
   a. The External SPI Master is connected to the MachXO2 dedicated slave Configuration SPI port pins. The external SPI Master's chip select controls the SN input that enables the MachXO2's Configuration Logic block. The external master sends commands to the Configuration Logic block permitting it to interface to the Configuration Flash and the UFM.
      i. For information on the accessing the Flash Memory (UFM/Configuration) of the MachXO2 device through SPI can be found later in this document.
      ii. For more information on Programming the MachXO2 through SPI port reference, the SPI section of TN1204, MachXO2 Programming and Configuration Usage Guide.

Figure 9-14. External Master SPI Device Accessing the MachXO2 Configuration Logic



SPI Design Tips

1. For information on the SPI register definitions and command sequences reference SPI section of TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.
2. Take note when dynamically turning off components for power savings; the EFB requires the MachXO2 internal oscillator to be enabled even if it is not the source of the WISHBONE Clock.
3. The SPI bus is bidirectional. A byte is received for every byte transmitted. Always discard RX data to avoid Receiver Overrun Error (ROE).
4. SN is an active pin whenever the SPI core is instantiated, whether 'ufm_sn' appears on the EFB primitive or not. Thus, SN cannot be recovered as user I/O. SN should be tied high externally to augment the weak internal pull-up if not connected to an external Master SPI bus.
5. There are a number of SPI reference designs on the Lattice website (www.latticesemi.com/products/intellectualproperty/aboutreferencedesigns.cfm) including:
   a. RD1125: SPI Slave Peripheral using Embedded Function Block
   b. UG56: MachXO2 Hardened SPI Master/Slave Demo

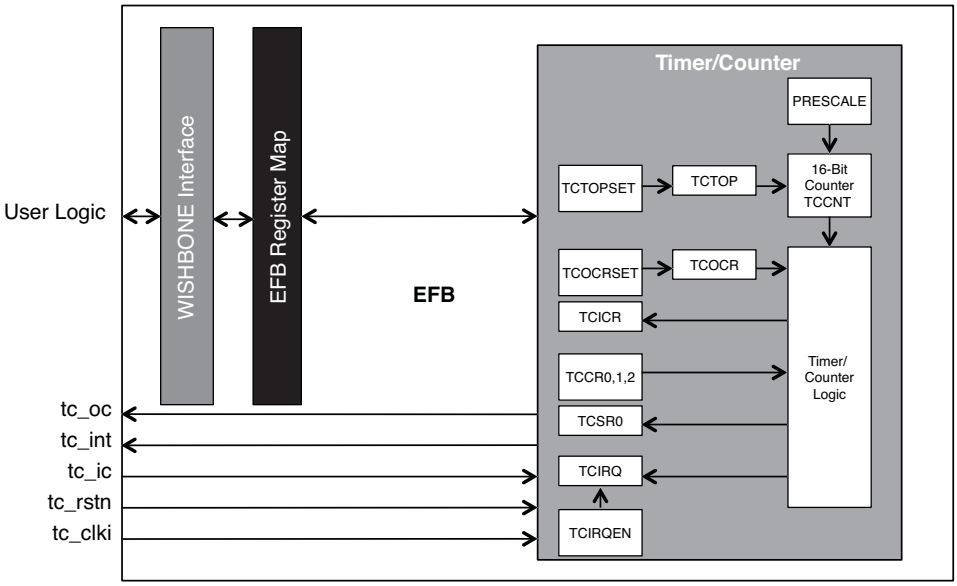
Timer/Counter

The MachXO2 EFB contains a Timer/Counter function. This Timer/Counter is a general purpose 16-bit Timer/Up Down Counter module with independent output compare units and Pulse Width Modulation (PWM) support. The Timer/Counter supports the following functions:

- Four unique modes of operation:
  – Watchdog timer

  - Clear Timer on Compare Match
  - Fast PWM
  - Phase and Frequency Correct PWM
- Programmable clock input source
- Programmable input clock pre-scale
- Interrupt output to FPGA fabric
- Three independent interrupt sources: overflow, output compare match, and input capture
- Auto reload
- Time-stamping support on the input capture unit
- Waveform generation on the output
- Glitch-free PWM waveform generation with variable PWM period
- Internal WISHBONE bus access to the control and status registers
- Stand-alone mode with preloaded control registers and direct reset input

Figure 9-15. Timer/Counter Block Diagram



The Timer/Counter communicates with the FPGA core logic through the WISHBONE interface and specific signals such as clock, reset, interrupt, timer output, and event trigger. The Timer/Counter can be included in a design with or without the WISHBONE interface.

Timer/Counter Modes of Operation

The Timer/Counter is a flexible function, which has four modes of operation. These modes are designed to meet different system needs related to sequencing of events and PWM (Pulse Width Modulation). The four Timer/Counter modes are:

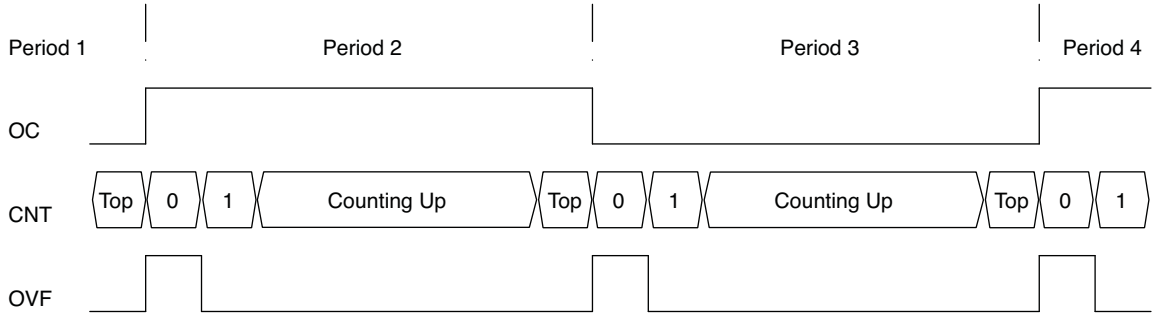
- Clear Timer on Compare Match
- Watchdog Timer
- Fast PWM
- Phase and Frequency Correct PWM

Clear Timer on Compare Match Mode

CTCM (Clear Timer on Compare Match) is a basic counter with interrupts. The counter is automatically cleared to 0x0000 when the counter value, TCCNT register, matches the value loaded in the TCTOP register. The value of the TCTOP register can by dynamically updated through the WISHBONE register, or it can hold a static value that is assigned with IPexpress at the time of IP generation. The default value of the TCTOP register is 0xFFFF.

The data loaded into the timer counter to define the top counter value is double-registered. The WISHBONE host writes the data to TCTOPSET register, which is then automatically loaded onto the TCTOP register at the moment of auto-clear. Therefore, a new top value can be written to the TCTOPSET register after the overflow flag and during the counting-up to the top value. Updating the value of the TCTOP register will change the frequency of the output signal of the Timer/Counter.

Figure 9-16. Timer/Counter Output Waveform



Watchdog Timer Mode

Watchdog timers are used to monitor a system's operating behavior and provide a reset or interrupt when the system's microcontroller or embedded state machine is no longer operational. One scenario is for a microcontroller to reset the Watchdog Timer to 0x0000 before it begins a process. The microcontroller must complete the process and reset the Watchdog Timer before the timer reaches its terminal count. In the event that the microprocessor does not clear the timer quickly enough the Watchdog Timer asserts a strobe signaling time expired. The system uses the "time exceeded" strobe to gracefully recover the system.

Another way to use the Watchdog Timer is to periodically turn OFF system modules in order to save power. It can also be used to interact with the on-chip power controller of MachXO2.

The most commonly used ports of the Timer/Counter in Watchdog Timer Mode are the clock, reset and interrupt. Optionally, the WISHBONE interface can be used to read time stamps from the TCICR register and update the top value of the counter.

Fast PWM Mode

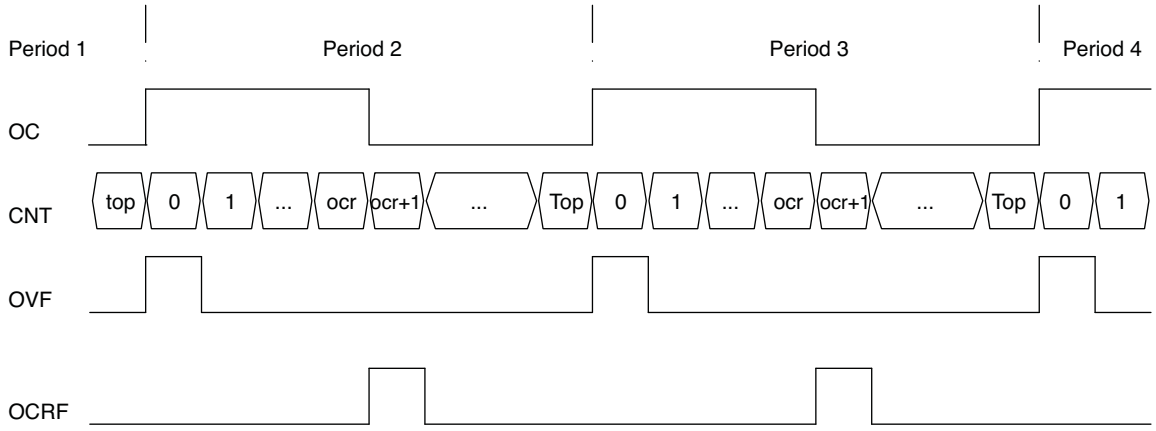
Pulse-Width Modulation (PWM) is a popular technique to digitally control analog circuits. PWM uses a rectangular pulse wave whose pulse width is modulated resulting in the variation of the average value of the waveform. Designers can vary the period and the duty cycle of the waveform by loading 16-bit digital values on the TCTOP register to define the top value of the counter, and the TCOCR register to provide a compare value for the output of the counter.

The output of the Timer/Counter is cleared when the counter value matches the top value that is loaded in the TCTOP register. The output is set when the value of the counter matches the compare value that is loaded into the TCOCR register. The clear/set functions can be inverted. This means that the output of the Timer/Counter is set when the counter value matches the top value and it is cleared when the value of the counter matches the compare value.

The interrupt line can be used for Overflow Flag (OVF) and Output Compare Flag (OCRF).

Figure 9-8 shows the PWM waveform generation. The output of the Timer/Counter is configured to be set when the counter matches the top value and clear when the counter matches the compare value.

Table 9-8. PWM Waveform Generation



Phase and Frequency Correct PWM Mode

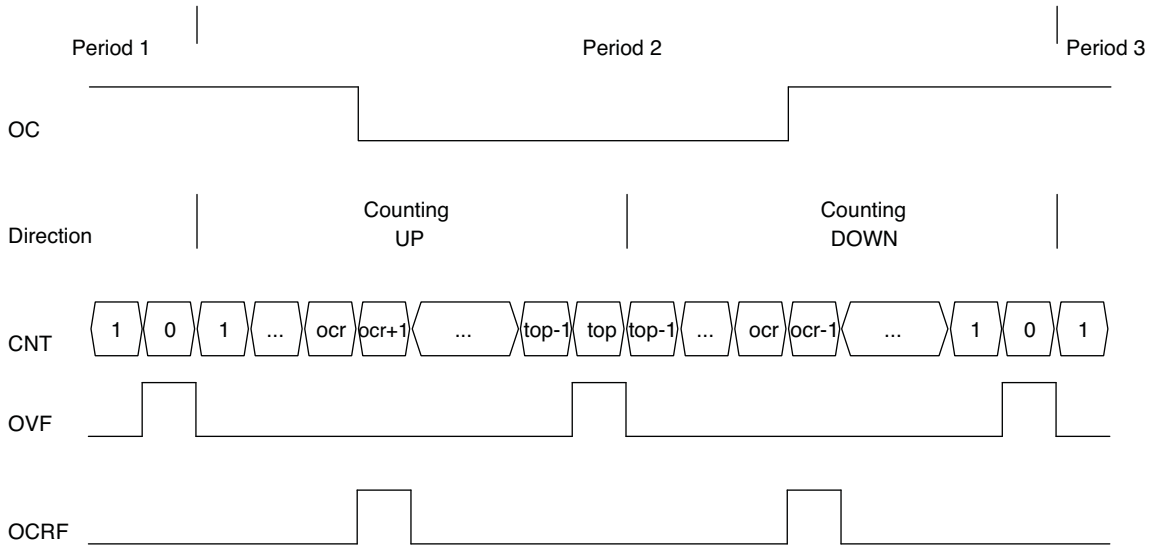
In phase and frequency correct PWM mode, the counting direction will change from up to down at the moment the counter is incrementing to the top value (top value minus 1). The moment that the counter is decrementing from 0x0001 to 0x0000, the following will occur:

1. TCTOP is updated with the value loaded in the TCTOPSET register
2. TCOCR is updated with the value loaded in the TCOCRSET register
3. Overflow TCSR[OVF] is asserted for one clock cycle

The output of the Timer/Counter is updated only when the counter value matches the compare value in TCOCR register. This occurs twice within one period. The first match occurs when the counter is counting up and the second match occurs when the counter is counting down. The Output Compare Flag TCSR[OCRF] is asserted when both matches occur. The output of the Timer/Counter is set on the first compare match and cleared on the second compare match. The order of set and clear can be inverted.

The mode allows users to adjust the frequency (based on the top value) and phase (based on the compare value) of the generated waveform.

Figure 9-17. Phase and Frequency Correct PWM Waveform Generation



Timer/Counter IP Signals

Table 9-9 documents the signals that are generated with the IP. Each signal has a description of the usage and how it should be connected in a design project.

Table 9-9. Timer/Counter – IP Signals

Signal Name	I/O	Width	Description
tc_clki	Input	1	Timer/Counter input clock signal. Can be connected to the on-chip oscillator. The clock signal is limited to 133 MHz.
tc_rstn	Input	1	This is an active-low reset signal, which resets the 16-bit counter.
tc_ic	Input	1	This is an active-high input capture trigger event, applicable for non-PWM modes with WISHBONE interface. If enabled, a rising edge of this signal will be detected and synchronized to capture the counter value (TCCNT Register) and make the value accessible to the WISHBONE interface by loading it into TCICR register. The common usage is to perform a time-stamp operation with the counter.
tc_int	Output	1	This is an interrupt signal, indicating the occurrence of a specific event such as Overflow, Output Compare Match, or Input Capture.
tc_oc	Output	1	Timer/Counter output signal

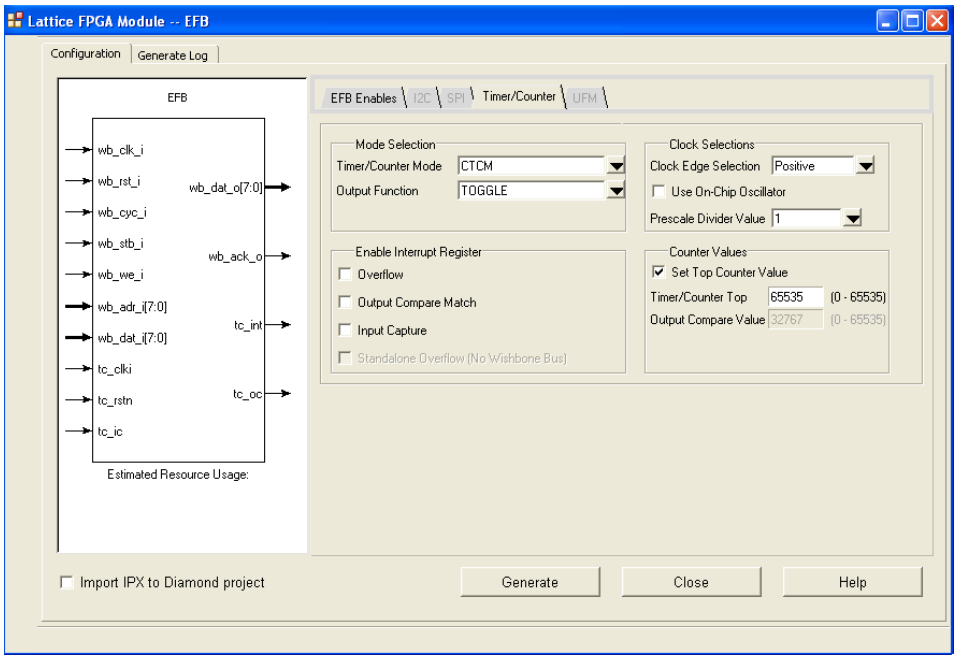
Configuring the Timer/Counter

IPexpress is used to configure the Timer/Counter. Selecting the Timer/Counter tab in the EFB GUI displays the configurable settings of the Timer/Counter core.

The Timer/Counter can be used with our without the WISHBONE interface. For usage without the WISHBONE interface, designers can select/enter values in the IPexpress EFB GUI. The values are programmed in the Timer/Counter registers with the programmable bitstream. Using the Timer/Counter with a WISHBONE interface enables users to dynamically update the register content through the WISHBONE interface. The main EFB GUI, EFB Enables tab, allows the user to make a selection between using the Timer/Counter with or without a WISHBONE screen shot.

Figure 9-18 shows an example of the Timer/Counter is configured for a specific design.

Figure 9-18. Configuring the Timer/Counter



Timer/Counter Mode
This option allows the user to select one of the four operating modes.

- CTCM – Clear Timer on Compare Match
- WATCHDOG – Watchdog timer
- FASTPWM – Fast PWM
- PFCPWM – Phase and Frequency Correct PWM

This option can be updated dynamically by modifying the bits TCM[1:0] of the register TCCR1.

Output Function
Designers can select the function of the output signal (tc_oc) of the Timer/Counter IP. The available functions are:

- STATIC – The output of the Timer/Counter is static low
- TOGGLE – The output of the Timer/Counter toggles based on the conditions defined by the Timer/Counter Mode
- WAV_GENERATE – The waveform is generated by the Set/Clear on the conditions defined by the Timer/Counter Mode
- INV_WAV_GENERATE – The waveform is inverted

This option can be updated dynamically by modifying bits OCM[1:0] of the register TCCR1.

Clock Edge Selection
Designers can select the edge (positive or negative) of the input clock source as well as enable the usage of the on-chip oscillator. The selections are:

- PCLOCK – Positive edge of the clock from user logic
- POSC – Positive edge of the clock from the internal oscillator
- NCLOCK – Negative edge of the clock from user logic
- NOSC – Negative edge of the clock from the internal oscillator

This option can be updated dynamically by modifying the bits CLKSEL and CLKEDGE of the register TCCR0.

Pre-scale Divider Value
Pre-scale divider values (0, 1, 8, 64, 256, 1024) are provided to divide the input clock prior to reaching the 16-bit counter. This option can be updated dynamically by modifying the bits PRESCALE[2:0] of the register TCCR1.

Timer/Counter Top
Designers can select the top value of the counter. This option can be updated dynamically by modifying the bits TCTOPSET of the registers TCTOPSET1 and TCTOPSET0.

Output Compare Value
Designers can select the output compare value of the counter. This option can be updated dynamically by modifying the bits TCOCRSET the registers TCOCRSET1 and TCOCRSET0

Enable Interrupt Registers

- Overflow – An interrupt which indicates the counter matches the TCTOP0/1 register value. The interrupt is bit IRQOVF of the register TCIRQ. When enabled, indicates OVF was asserted. Write a '1' to this bit to clear the interrupt. This option can be updated dynamically by modifying the bit IRQOVFEN of the register TCIRQEN.
- Output Compare Match – An interrupt which indicates when counter matches the TCOCR0/1 register value. The interrupt is bit IRQOCRF of the register TCIRQ. When enabled, indicates OCRF was asserted. Write a '1' to this bit to clear the interrupt. This option can be updated dynamically by modifying the IRQOCRFEN bit of register TCIRQEN.
- Input Capture – An interrupt which indicates when the user asserts the TC_IC input signal. The interrupt is bit IRQICRF of the register TCIRQ. When enabled, indicates ICRF was asserted. Write a '1' to this bit to clear the interrupt. This option can be updated dynamically by modifying IRQICRFEN bit of the register TCIRQEN.
- Standalone Overflow – Used without the WISHBONE interface and serves as the only available interrupt request.

MachXO2 Timer/Counter Usage Cases

1. Basic counter with interrupts
   a. Configure the Timer/Counter for Static or Dynamic operation. Dynamic operation enables the WISHBONE bus interface allowing a WISBONE Master to control the Timer/Counter.
   b. In the Timer Counter tab
      i. Configure the Timer/Counter Mode using the Mode Selection controls
         1. Select CTCM (Clear Timer on Compare Match) Mode
         2. Select the Output Function
            a. Hold static 0 (STATIC)
            b. Toggle (TOGGLE)
      ii. Update the Clock Selections
         1. Select the clock edge to which the Timer/Counter responds
            a. Positive (PCLOCK) or Negative (NCLOCK) edge
         2. Select the Clock Pre-scale Divider
      iii. Configure the Counter Values
         1. Enable the Top Counter Value
         2. Select a Timer/Counter Top value
      iv. Enable optional interrupts
2. Watchdog Timer
   a. Configure the Timer/Counter for Static or Dynamic operation. Dynamic operation enables the WISHBONE bus interface allowing a WISBONE Master to control the Timer/Counter.
   b. In the Timer Counter tab

i. Configure the Timer/Counter Mode using the Mode Selection controls
   1. Select WATCHDOG mode
   2. Select Output Function
      a. Hold static 0 (STATIC)

ii. Update the Clock Selections
   1. Select the clock edge to which the Timer/Counter responds
      a. Positive (PCLOCK) or Negative (NCLOCK) edge
   2. Select the Clock Pre-scale Divider

iii. Configure the Counter Values
   1. Enable the Top Counter Value
   2. Select a Timer/Counter Top value
   3. Select an Output Compare Value

iv. Enable interrupts (TC_INT)
   1. Select Output Compare Match
   2. Select Input Capture

3. PWM Output with variable duty cycle and period
   a. Configure the Timer/Counter for Static or Dynamic operation. Dynamic operation enables the WISHBONE bus interface allowing a WISBONE Master to control the Timer/Counter.
   b. In the Timer Counter tab
      i. Configure the Timer/Counter Mode using the Mode Selection controls
         1. Select the FASTPWM mode
         2. Select Output Function
            a. Hold static 0 (STATIC)
            b. PWM (WAVE_GENERATOR)
            c. Complement PWM (INV_WAVE_GENERATOR)
      ii. Update the Clock Selections
         1. Select the clock edge to which the Timer/Counter responds
            a. Positive (PCLOCK) or Negative (NCLOCK) edge
         2. Select the Clock Pre-scale Divider
      iii. Configure the PWM Values
         1. Select the PWM period (Timer Counter Top)
         2. Select the PWM duty cycle (Output Compare Value)
            a. Where the duty cycle is (Output Compare Value)/(Timer Counter Top): [(Timer Counter Top) – (Output Compare Value)]/(Timer Counter Top)
4. PWM output with 50:50 duty variable phase and period
   a. Configure the Timer/Counter for Static or Dynamic operation. Dynamic operation enables the WISHBONE bus interface allowing a WISBONE Master to control the Timer/Counter.
   b. In the Timer Counter tab
      i. Configure the Timer/Counter Mode using the Mode Selection controls
         1. Select FASTPWM
         2. Select Output Function
            a. Hold static 0 (STATIC)
            b. PWM (WAVE_GENERATOR)
            c. Complement PWM (INV_WAVE_GENERATOR)
      ii. Update the Clock Selections
         1. Select the clock edge to which the Timer/Counter responds
            a. Positive (PCLOCK) or Negative (NCLOCK) edge

2. Select the Clock Pre-scale Divider

iii. Configure the PWM Values

1. Select the PWM period (Timer Counter Top)
2. Select the Phase Adjustment (Output Compare Value)
    a. Where Phase Adjustment is (360 degrees)*(Output Compare Value)/ (Timer Counter Top)

Timer/Counter Design Tips

1. For information on the Timer/Counter register definitions and command sequences reference Timer/Counter section of TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.
2. Take note when dynamically turning off components for power savings; the EFB requires the MachXO2 internal oscillator to be enabled even if it is not the source of the WISHBONE Clock.

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 three sectors:

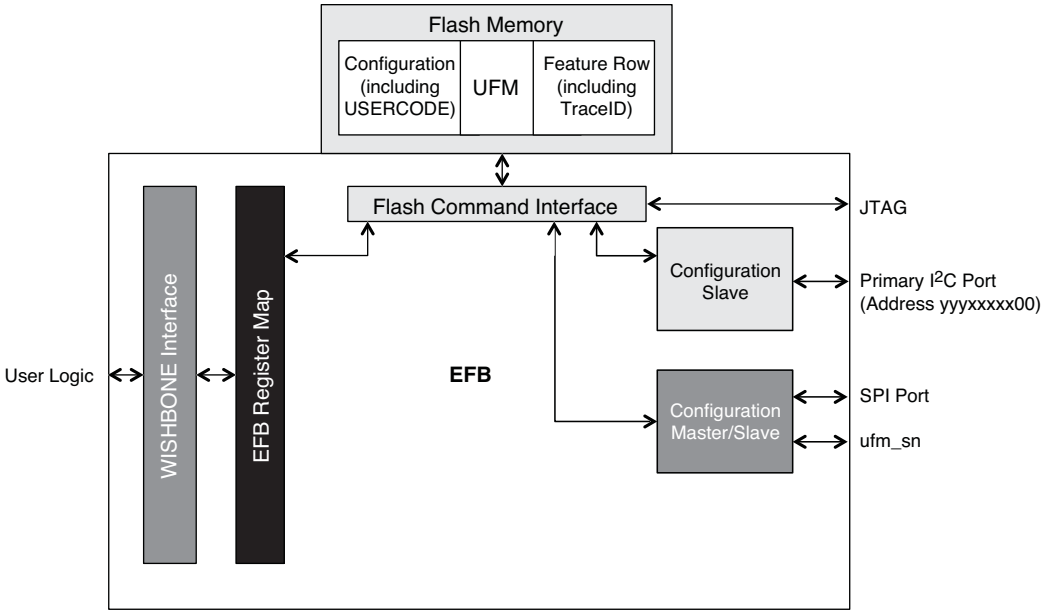
- User Flash Memory (UFM)
- Configuration
- Feature Row

The ports to access the Flash Memory and the block diagram are shown in Table 9-10.

Table 9-10. Flash Memory (UFM/Configuration) Access

	UFM	Configuration Flash
JTAG	Yes	Yes
Slave SPI Port with Chip Select (ufm_sn)	Yes	Yes
Slave SPI Port with Chip Select (spi_scsn)	No	No
Primary Slave I2C Port address (yyyxxxxx00)	Yes	Yes
Primary Slave I2C Port address (yyyxxxxx01)	No	No
Secondary Slave I2C Port address (yyyxxxxx10)	No	No
WISHBONE	Yes	Yes

Figure 9-19. Flash Memory (UFM/Configuration) Block Diagram



Flash Memory (UFM/Configuration) Access Ports

The Configuration Logic arbitrates access to the Flash Memory 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 – All MachXO2 devices have a JTAG port, which can be used to perform Read/Write operations to the Flash Memory (UFM/Configuration). The JTAG port is compliant with the IEEE 1149.1 and IEEE 1532 specifications. JTAG has the highest priority to the Flash Memory (UFM/Configuration).

2. Slave SPI Port – All MachXO2 devices have a Slave SPI port, which can be used to perform Read/Write operations to the Flash Memory (UFM/Configuration). Asserting the Flash Memory (UFM/Configuration) Slave Chip Select (ufm_sn) will enable access.
3. I^2C Primary Port – All MachXO2 devices have an I^2C port, which can be used to perform Read/Write operations to the Flash Memory (UFM/Configuration). The Primary I2C Port address is yyyxxxxx00 (Where 'y' and 'x' are user programmable).
4. WISHBONE Slave Interface – The WISHBONE Slave interface of the EFB module enables designers to access the Flash Memory (UFM/Configuration) from the FPGA user logic by creating a WISHBONE Master. In addition to the WISHBONE bus signals, an interrupt request output signal is brought to the FPGA fabric. An IRQ (wbc_ufm_irq) is provided to assist the WB Master to perform UFM/CF programming operations. It is configurable and provides information about the R/W FIFO, and arbitration errors.

   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, reference AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

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

- *Power Controller*
- *GSR*
- *Hardened User SPI Port*
- *Hardened Primary User I^2C Port*

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

Interface to UFM

MachXO2-640 and higher density devices provide one sector of User Flash Memory (UFM). Designers can access the UFM sector through the Configuration Logic interface using the JTAG, Configuration SPI, Primary Configuration I^2C or WISHBONE interfaces. The UFM is a separate sector within the on-chip Flash Memory and is organized by pages. Each page is 128 bits (16 bytes). Table 9-11 shows the UFM resources in each device, represented in bits, bytes and pages.

Table 9-11. UFM Resources in MachXO2 Devices

	MachXO2-256	MachXO2-640	MachXO2-640U	MachXO2-1200	MachXO2-1200U	MachXO2-2000	MachXO2-2000U	MachXO2-4000	MachXO2-7000
UFM Bits	0	24,576	65,536	65,536	81,920	81,920	98,304	98,304	262,144
UFM Bytes	0	3,072	8,192	8,192	10,240	10,240	12,288	12,288	32,768
UFM Pages	0	192	512	512	640	640	768	768	2048

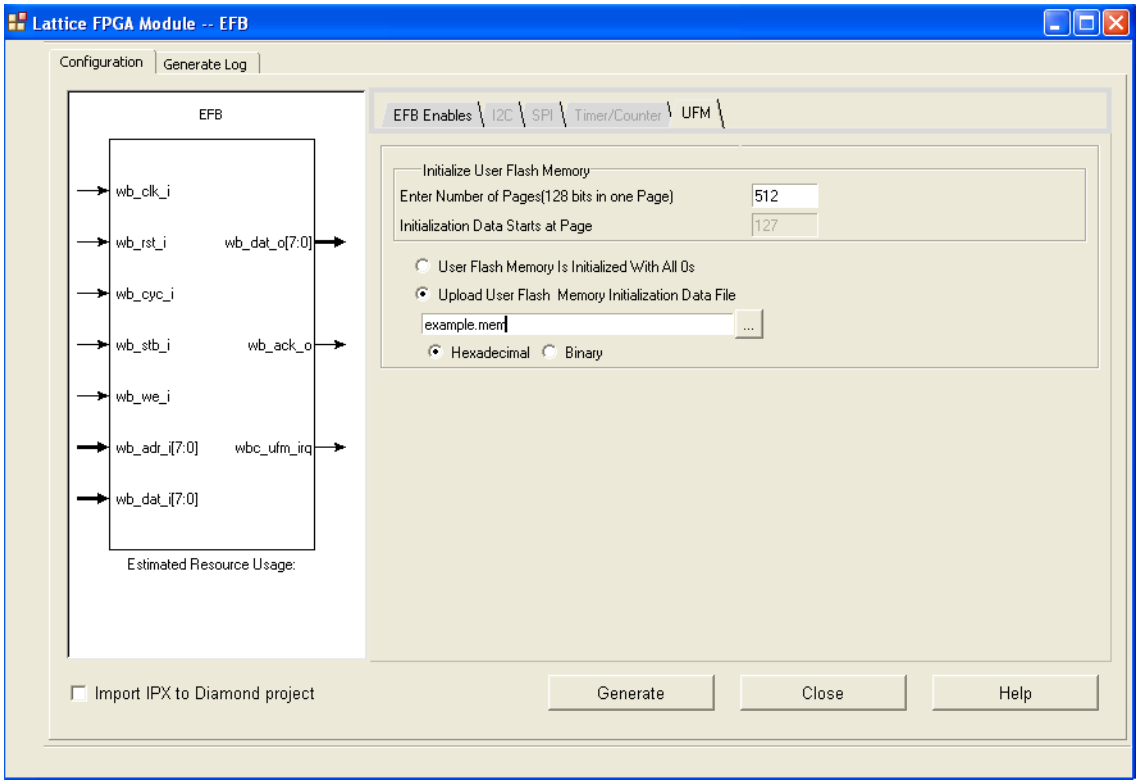
The UFM is a general purpose Flash Memory. Refer to the MachXO2 Family Data Sheet for the number of Programming/Erase Specifications. The UFM is typically used to store system-level data, Embedded Block RAM initialization data, or executable code for microprocessors and state-machines. Use the following tools to partition the UFM resource:

- IPexpress: Use IPexpress to enable the UFM and to initialize the memory
- Spreadsheet View (Diamond): The global sysConfig configuration options control the use of the UFM. Read TN1204, MachXO2 Programming and Configuration Usage Guide for a complete description of available options.

Initializing the UFM with IPexpress

Designers can initialize the UFM sector with system level non-volatile data and generate the EFB module via IPexpress. Selecting the UFM tab in the EFB GUI will display the configurable settings of the UFM.

Figure 9-20. Initializing the UFM Sector with IPexpress



Designers enter the number of pages to be pre-loaded with initialization data. Because initialization data is 'bottom justified,' the read-only field “Initialization Data Starts at Page” informs the designer with the address of the first page that is initialized. In the example used for this document, 512 pages of the UFM sector of MachXO2-2000 are to be initialized with the content of the memory file “example.mem”.

	UFM Utilization Example (MachXO2-2000) Initializing 512 Pages
Uninitialized Pages	Page 0
	Page 1
	...
First Page of Initial Data →	Page 128
	Page 129
	...
Last Page of Initial Data →	Page 639

The MachXO2-2000 has 640 pages in the UFM sector. The initialization data occupies the bottom (highest) addressable pages of the UFM, while the top (lowest) addressable pages remain uninitialized (all zeros). The format of the memory file is page-based and can be expressed in binary or hexadecimal format.

UFM Initialization Memory File

The initialization data file has the following properties and format:

Extension:	.mem
Format:	Binary, Hexadecimal
Data Width:	1 page (128 bits in one row of the file)
No. of Rows:	Less than or equal to the number of available pages

Example 1. Binary

1010…1010 (Placed at the starting page of the UFM initialization data, address = N)
1010…1010 (page address = N + 1)
1010…1010 (page address = N + 2)
…
1010…1010 (Placed at the highest UFM page address)

Example 2. Hexadecimal

A…B (Placed at the starting page of the UFM initialization data, address = N)
C…D (page address = N + 1)
E…F (page address = N + 2)
…
A…F (Placed at the highest UFM page address)

The most significant byte (byte 15) of the page is on the left side of the row. The least significant byte of the page (byte 0) is on the right side of the rowbit in a row is the left-most (i.e. bit 127), the least significant is right-most (bit 0)

EBR Initialization

For designers that choose to share the UFM sector with EBR initialization data, the sector map below should be referenced as an example when planning the design. Note at this time the EBR Mapping is not supported Diamond.

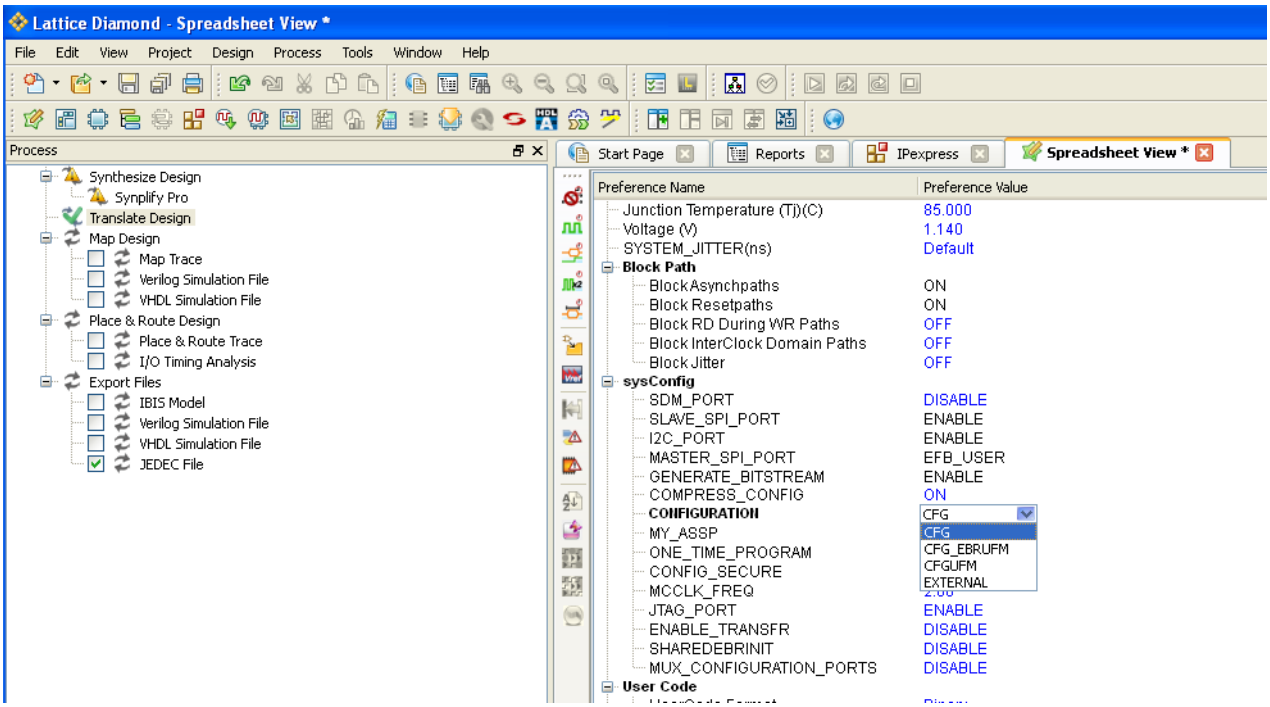
	UFM Sector (MachXO2-2000)
Block N Location and Initiliazation Data	Page 0 (EBR Block N Location)
	Page 1 (EBR Block N Initialization Data)
	...
	Page 72 (EBR Block N Initialization Data)
Block M Location and Initialization Data	Page 73 (EBR Block M Location)
	Page 74 (EBR Block M Initialization Data)
	...
	Page 145 (EBR Block M Initialization Data)
Pages Available for General Purpose Non-volatile Data	Page 146
	...
	...
	Page 639

Care must be taken when performing a Bulk-Erase operation to prevent the loss of EBR initialization data. Prior to erasing the UFM sector, designers should make a copy of the pages holding the EBR block location and EBR initialization data in a secondary memory resource. After the UFM sector is erased, the data can be written back into the UFM. Upon power-up or a reconfiguration command, the EBR initialization data is automatically loaded in their respective EBR blocks. The EBR Block Location data guides the configuration logic on the location of the EBR block.

UFM in Lattice Diamond Software

The UFM sector is one of the Flash Memory resources of the MachXO2. The CONFIGURATION option in the Diamond Spreadsheet view controls how UFM behaves. Each options impact to the UFM is described below.

Figure 9-21. UFM in Diamond Spreadsheet View



Configuration Parameter Options

- CFG_EBRUFM – The SRAM configuration (not including EBR initialization data) is stored in the Configuration Flash. EBR initialization data, if any, is stored in the lowest page addresses (starting from Page 0) of the UFM sector. Any unoccupied UFM pages after the mapping of EBR initialization data is available as general purpose memory.
- CFG – The SRAM configuration (including EBR initialization data, if any) is stored in the Configuration Flash array and does not overflow into UFM. The full UFM sector is available as general purpose Flash memory.
- CFGUFM – The SRAM configuration (including EBR initialization data, if any) is stored in the Configuration Flash and is allowed to overflow into UFM. The UFM cannot be used as general purpose memory.
- EXTERNAL – The SRAM configuration (including EBR initialization data, if any) is stored in an external memory SPI memory. The full UFM sector is available as general purpose Flash memory.

Configuration Flash Memory

The WISHBONE interface of the EFB module allows a WISHBONE master to access the Configuration resources of MachXO2 devices. This is particularly useful for reading data from Configuration resources such as USERCODE and TraceID. A WISHBONE master can update the Configuration Flash memory using the Configuration Logic's transparent mode. The new design is active after power-up or a Configuration Refresh operation.

For more information on Programming the MachXO2, reference TN1204, MachXO2 Programming and Configuration Usage Guide and TN1246 Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide.

For more information on the TraceID, reference TN1207, Using TraceID in MachXO2 Devices.

Table 9-12. Configuration Flash Resources in MachXO2 Devices

	MachXO2-256	MachXO2-640	MachXO2-640U	MachXO2-1200	MachXO2-1200U	MachXO2-2000	MachXO2-2000U	MachXO2-4000	MachXO2-7000
CFG Bits	73,600	147,328	278,400	278,400	409,344	409,344	737,024	737,024	1,179,136
CFG Bytes	9,200	18,416	34,800	34,800	51,168	51,168	92,128	92,128	147,392
CFG Pages	575	1,151	2,175	2,175	3,198	3,198	5,758	5,758	9,212

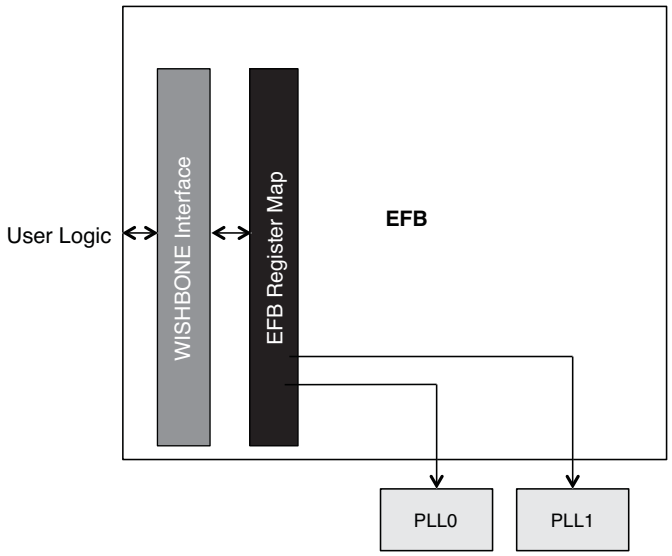
Flash Memory (UFM/Configuration) Design Tips

1. For information on the Flash Memory (UFM/Configuration) register definitions and command sequences reference Flash Memory (UFM/Configuration) section of TN1246, Reference Guide for Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.
2. For more information Programming the MachXO2, reference TN1204, MachXO2 Programming and Configuration Usage Guide.
3. For more information on the TraceID, reference TN1207, Using TraceID in MachXO2 Devices.
4. Take note when dynamically turning off components for power savings; The EFB requires the MachXO2 internal oscillator to be enabled even if it is not the source of the WISHBONE Clock.
5. It is recommended when accessing the Configuration Flash the Feature Row be Read Only after initial programming as it contains the persistent settings for the Configuration ports
6. The Configuration Flash Chip Select (ufm_sn) is enabled when the UFM and SPI functions are enabled. Tie ufm_sn inactive (i.e. high) if you are not using the Slave SPI interface to access the Configuration Flash/UFM.
7. The buses used to access the Flash Memory and UFM have a priority. The bus priority is, from highest to lowest, the JTAG Port, Slave SPI Port, I^2C Primary Port, and WISHBONE Slave Interface. When higher priority ports are enabled Flash Memory access by lower priority ports is blocked. You must define a process that prevents simultaneous access to the Configuration Flash/UFM by masters on each configuration port.
8. The Primary I^2C port cannot be used for both UFM/Configuration access and user functions in the same design.
9. Enabling Flash Memory (UFM/Configuration) Interface using Enable Configuration Interface command 0x74 Transparent Mode will temporarily disable the Power Controller, GSR, Hardened User SPI port, Functionality is restored after the Flash Memory (UFM/Configuration) Interface is disabled using Disable Configuration Interface command 0x26 followed by Bypass command 0xFF.
10. In Flash memory, ‘0’ defines erased, ‘1’ defines written
11. Smallest unit for a Write operation (bits => 1) is 1 page (16 bytes)
12. Smallest unit for an Erase operation (bits => 0) is one sector (There are only two available Flash sectors: Configuration and UFM)
13. The UFM has a limited number of erase/programming cycles. The number of cycles is described in DS1035. Lattice recommends storing static, or data that varies infrequently in the UFM.
14. There are a number of Flash Memory (UFM/Configuration) Reference designs on the Lattice website (www.latticesemi.com/products/intellectualproperty/aboutreferencedesigns.cfm) including:
    a. RD1126: RAM-Type Interface for Embedded User Flash Memory
    b. UG57: MachXO2 Programming via WISHBONE Demo

Interface to Dynamic PLL Configuration Settings

The WISHBONE interface of the EFB module can be used to dynamically update configurable settings of the Phase Locked Loops (PLLs) in MachXO2 devices.

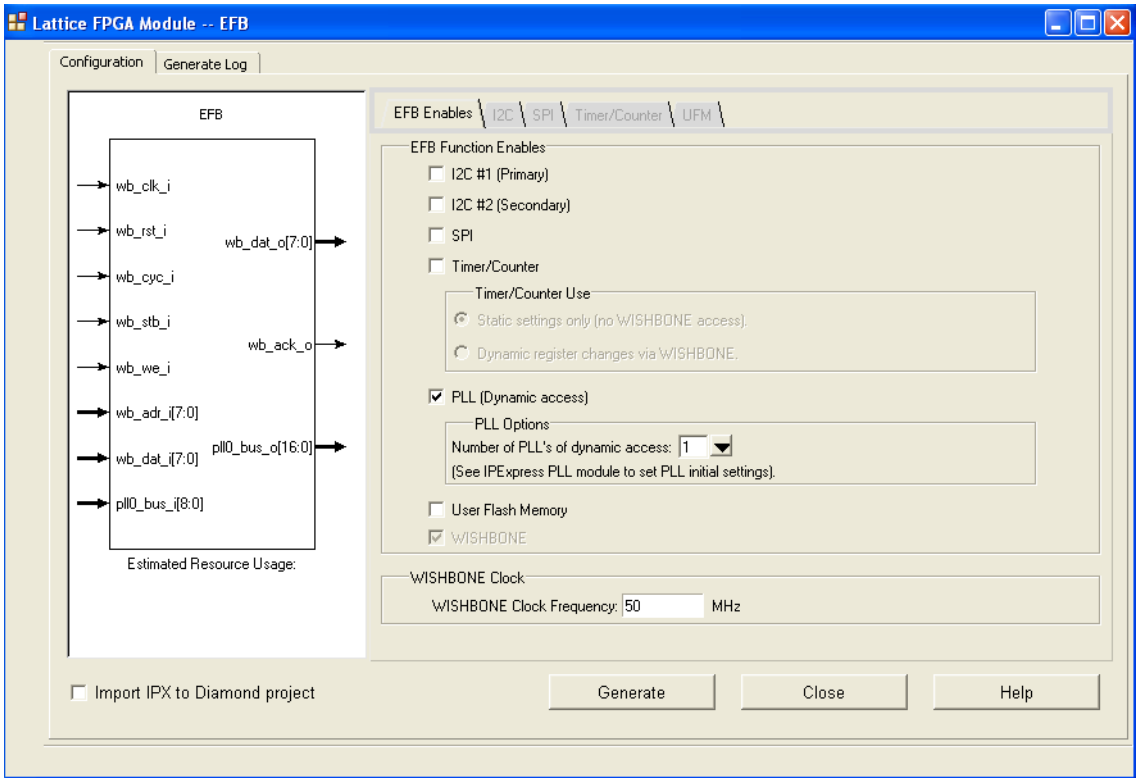
Figure 9-22. EFB Interface to Dynamic PLL



There can be up to two PLLs in a MachXO2 device. PLL0 has an address range from 0x00 to 0x1F in the EFB register map. PLL1 (if present) has an address range from 0x20 to 0x3F in the EFB register map. Reference TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide, for details on PLL configuration bits and recommended usage.

Users can enable the WISHBONE interface to the PLL components in IPexpress, EFB GUI as shown in Figure 9-23.

Figure 9-23. Interface to Dynamic PLL Configuration Settings



Enabling the interface to dynamically control PLL settings through the WISHBONE interface will generate an IP with the following ports, which are used for dedicated connections to the PLL(s).

Table 9-13 documents the signals that are generated with the IP. Each signal has a description of the usage and how it should be connected in a design project.

Table 9-13. PLL Interface – IP Signals

Signal Name	I/O	Width	Description
pll0_bus_i	Input	9	Input data and control bus. Users must connect this bus only to a PLL component that is instantiated in the design.
pll0_bus_i	Input	9	Input data and control bus. Users must connect this bus only to a PLL component that is instantiated in the design.
pll0_bus_o	Output	17	Output data and control bus. Users must connect this bus only to a PLL component that is instantiated in the design.
pll0_bus_1	Output	17	Output data and control bus. Users must connect this bus only to a PLL component that is instantiated in the design.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
January 2011	01.1	Updated for ultra-high I/O (“U”) devices.
June 2011	02.0	Expanded WISHBONE register definitions for I^2C, SPI, TC and UFM/Configuration blocks.
		Added Master I^2C and Master SPI WISHBONE flow diagrams.
		Expanded UFM/Configuration Command tables.
		Added UFM/Configuration Command descriptions
		Added I^2C and SPI timing diagrams.
		Added Configuration Flash resources table.
		Added UFM/Configuration performance table.
		Added Appendices A, B and C.
		Various minor corrections, additions and refinements to the text.
July 2011	02.1	Added diagram: I^2C Slave Read/Write Example (via WISHBONE).
		Added references to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (R-1) Devices, throughout the document.
August 2011	02.2	Added diagram: Basic Configuration Flash Update Example.
		Removed 'Erase_DONE' command.
		Clarified SPITXDR and WBRESET timing.
September 2011	02.3	Clarified WISHBONE timing for wait states.
		Corrected Flash Check Status checking.
		Modified I^2C Master Read/Write Example (Figure 9-10).
		Clarified R1 UFM/Config Access.
		Clarified Flash address auto-increment functionality.
		Added notation for non-simulation commands (Table 9-57).
October 2011	02.4	Corrected SPI Clock Prescale equation.
		Clarified user SPI port restoration command sequence following Configuration Enable.
		Added 0xC6 (Enable Offline Configuration) and 0xFF (Bypass).
December 2011	02.5	Noted operational delays which must be observed following wb_rst_i and UFM/Config Enable, Erase and Programming commands.
		Added new figure: SPI Master Read/Write Example (via WISHBONE) – Production Silicon.
		GSR and Power Controller functionality is suspended during UFM/Configuration Interface Enable (0x74)
		BYPASS command (0xFF) must now follow all Configuration Disable (0x26) commands.
		Added Feature Row capability to Erase Flash (0x0E) command.
		Corrected examples for Read Config Flash (0x73)
January 2012	02.6	Check Busy Flag (0xF0) table, updated example
		Added operational delays following Enable Configuration Interface command in Appendix B examples.
January 2012	02.7	Corrected Data Format (Binary) for Check Busy Flag (0xF0).
January 2012	02.8	Check Busy Flag (0xF0) Data Format changed from: B: bit 0: Busy Flag (1= busy) to B: bit 7: Busy Flag (1= busy)

Date	Version	Change Summary
January 2012	02.9	Updated UFM and Configuration Resource tables.
		Changed BYPASS operands to FF FF FF.
February 2012	03.0	Updated document with new corporate logo.
February 2012	03.1	Fixed Busy flag read example.
June 2012	04.0	Split into two documents: TN1205 Usage Guide and TN1246 Reference Guide.
October 2012	04.1	Added restriction: Primary port can be used as Configuration/UFM port or as a user port, but not both.
		Added restriction: Primary I^2C port is unavailable while in ISC_ENABLE_X (transparent) configuration access mode.

MachXO2 sysIO Usage Guide

March 2013 Technical Note TN1202

Introduction

The MachXO2™ PLD family sysIO™ buffers are designed to meet the needs of flexible I/O standards in today's fast-paced design world. The supported I/O standards range from single-ended I/O standards to differential I/O standards so that users can easily interface their designs to standard buses, memory devices, video applications and emerging standards. This technical note provides a description of the supported I/O standards and the banking scheme for the MachXO2 PLD family. The sysIO architecture and the software usage are also discussed to provide a better understanding of the I/O functionality and placement rules.

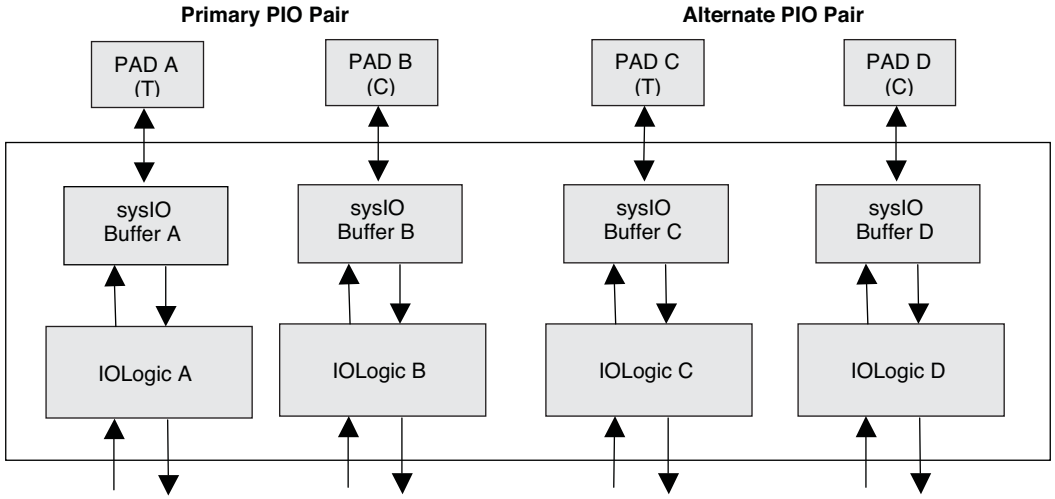
sysIO Buffer Overview

The basic building block of the MachXO2 sysIO is the Programmable I/O Cell (PIC) block. There are four types of PIC blocks in the MachXO2 device architecture. These include the basic PIC block, the memory PIC block for DDR memory support, the receiving PIC block with gearing, and the transmitting PIC block with gearing. The PIC blocks with gearing are used for video and high-speed applications. The PIC blocks with gearing have a built-in control module for word alignment. The memory PIC block has additional logic to manage DQS strobe signals and clock phase shift. The details of the memory PIC block and the gearing PIC block can be found in TN1203, MachXO2 High-Speed Source Synchronous and Memory Interfaces.

A common feature of all four types of PIC blocks is that each PIC block consists of four programmable I/Os (PIOs). Each PIO includes a sysIO buffer and an I/O logic block. A simplified sysIO block diagram is shown in Figure 10-1. The I/O logic block consists of an input block, an output block, and a tri-state block. These blocks have registers, input delay cells, and the necessary control logic to support various operational modes. The sysIO buffer determines the compliance to the supported I/O standards. It also supports features like hysteresis to meet common design needs. The I/O logic block and the sysIO buffer are designed with a minimal use of die area; providing easy bus interfacing, and pin out efficiency.

Two adjacent PIOs can form a pair of complementary output drivers. In addition, PIOA and PIOB of the PIC block form the primary pair of the buffer, while PIOC and PIOD form the alternate pair of the buffer. The primary pairs have additional capability that is not available on the alternate pair. The sysIO buffers of the PIC block are equivalent when implemented as the single-ended I/O standards.

Figure 10-1. PIC Block Diagram



Supported sysIO Standards

The Lattice MachXO2 sysIO buffer supports both single-ended and differential standards. The single-ended standard can be further divided into internally ratioed standards such as LVCMOS, and externally referenced standards such as SSTL. The internally ratioed standards support individually configurable drive strength and bus maintenance circuits (weak pull-up, weak pull-down, or bus keeper).

There are two types of ratioed input buffers. One is connected to V_{CCIO} and the other is connected to V_{CC} (1.2V). Each sysIO buffer supports both buffers in parallel, and therefore provides an option to program any input buffer to be a 1.2V ratioed input buffer regardless of the V_{CCIO} voltage.

All banks of the MachXO2 devices support true differential inputs, and emulated differential outputs using external resistors and the complementary LVCMOS outputs. The true-LVDS differential outputs and LVDS input termination are supported in specific banks as described in the sysIO Banking Scheme section of this document.

Table 10-1. Supported Input Standards

Input Standard	V_{REF} (Nominal)	V_{CCIO}[1] (Nominal)
Single-Ended Interfaces		
LVTTL33	—	—
LVCMOS33	—	—
LVCMOS25	—	—
LVCMOS18	—	—
LVCMOS15	—	—
LVCMOS12	—	—
SSTL25 Class I, II	1.25	—
SSTL18 Class I, II	0.9	—
HSTL18 Class I, II	0.9	—
PCI33	-	3.3
Differential Interfaces		
LVDS25	—	—
LVPECL33	—	—
MLVDS25	—	—
BLVDS25	—	—
RSDS25	—	—
SSTL25 Differential	—	—
SSTL18D Differential	—	—
HSTL18D Differential	—	—
LVTTL / LVCMOS Differential	—	—

1. If not specified, refer to mixed voltage support in the VCCIO Requirement section.

Table 10-2. Supported Output Standards

Output Standards	Drive (mA)	V_{CCIO} (Nominal)
Single-Ended Interfaces		
LVTTL33	4, 8, 12, 16, 24	3.3
LVCMOS33	4, 8, 12, 16, 24	3.3
LVCMOS25	4, 8, 12, 16	2.5
LVCMOS18	4, 8, 12	1.8
LVCMOS15	4, 8	1.5
LVCMOS12	2, 6	1.2
SSTL25 Class I	8	2.5
SSTL18 Class I	8	1.8
HSTL18 Class I	8	1.8
PCI33	24	3.3
Differential Interfaces		
LVDS25	3.5, 2.5, 2.0, 1.25	2.5, 3,3
LVPECL33	16	3.3
MLVDS25	16	2.5
BLVDS25	16	2.5
RSDS25	8	2.5
SSTL25 Differential	8	2.5
SSTL18D Differential	8	1.8
HSTL18D Differential	8	1.8
LVTTL33 Differential	4, 8, 12, 16, 24	3.3
LVCMOS33 Differential	4, 8, 12, 16, 24	3.3
LVCMOS25 Differential	4, 8, 12, 16	2.5
LVCMOS18 Differential	4, 8, 12	1.8
LVCMOS15 Differential	4, 8	1.5
LVCMOS12 Differential	2, 6	1.2

sysIO Banking Scheme

The MachXO2 family has a non-homogeneous I/O banking structure. MachXO2-256, MachXO2-640/U and MachXO2-1200 have four I/O banks each with one I/O bank per side. MachXO2-1200U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices have six I/O banks each, with one I/O bank on each of the top, bottom and right sides, and three banks on the left side.

The MachXO-640U, MachXO-1200/U and higher density devices support true LVDS differential outputs through the primary pairs on the top bank (bank 0). These devices also support 100 ohm differential input termination on every I/O pair on the bottom I/O bank. There is also a programmable PCI clamp available on the bottom I/O bank for these devices. For the “R1” version of the MachXO2 devices, the 100 ohm differential input termination is approximately 200 ohms. The “R1” versions of the MachXO2 devices have an “R1” suffix at the end of the part number (e.g., LCMXO2-1200ZE-1TG144CR1). For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

MachXO2-256 and MachXO2-640 do not support true LVDS differential outputs, differential input termination, and PCI clamps in any banks (MachXO2-640U I/O architecture is similar to the larger devices and supports the aforementioned features). Each of the I/O pins on all MachXO2 PLDs has a clamp feature which can be disabled or enabled. This clamp is similar to the PCI clamp but it is not PCI compliant except in the bottom bank of the MachXO2-640U, MachXO2-1200/U and higher density devices. The arrangements of the I/O banks are shown in Figures 10-2, 10-3, and 10-4. DDR memory support in bank 1 is not available for devices in wafer level chip scale

packages (WLCSP).

Figure 10-2. MachXO2-256 and MachXO2-640 I/O Banking Arrangement

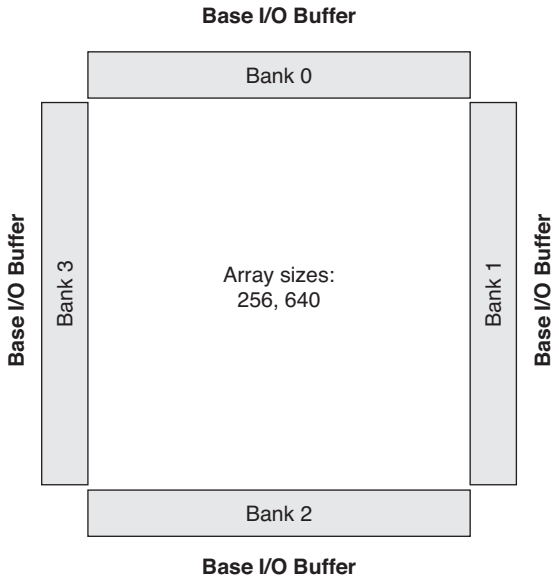


Figure 10-3. MachXO2-640U and MachXO2-1200 I/O Banking Arrangement

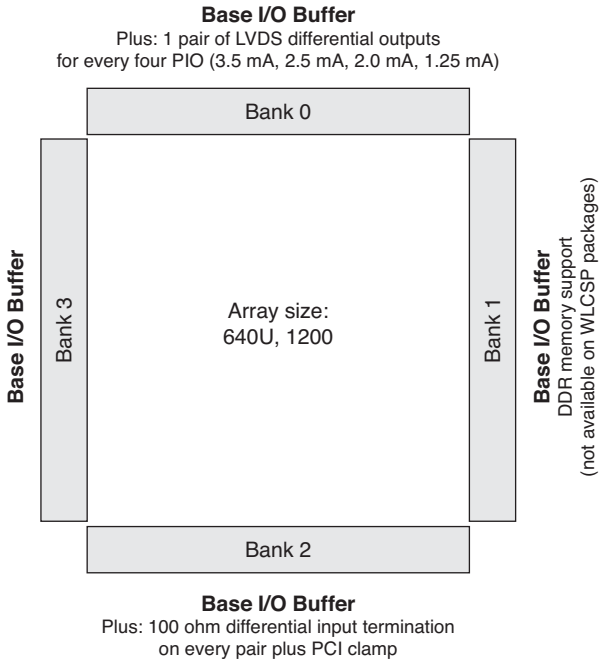
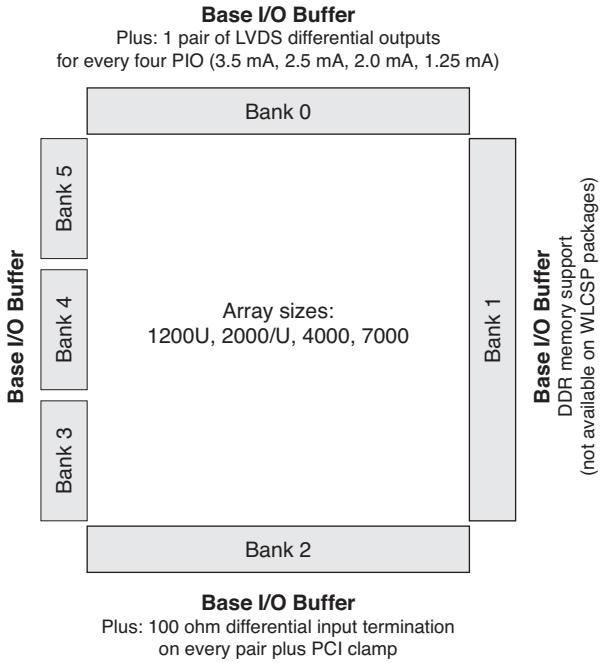


Figure 10-4. MachXO2-1200U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 I/O Banking Arrangement



sysIO Standards Supported by I/O Banks

All banks can support multiple I/O standards under the V_{CCIO} rules discussed above. Tables 10-3 and 10-4 summarize the I/O standards supported on various sides of the MachXO2 device.

Table 10-3. Single-Ended I/O Standards Supported on Various Sides

Standard	Top	Bottom	Left	Right
PCI33	—	Yes[1]	—	—
LVTTL33	Yes	Yes	Yes	Yes
LVCMOS33	Yes	Yes	Yes	Yes
LVCMOS25	Yes	Yes	Yes	Yes
LVCMOS18	Yes	Yes	Yes	Yes
LVCMOS15	Yes	Yes	Yes	Yes
LVCMOS12	Yes	Yes	Yes	Yes
SSTL25[2]	Yes	Yes	Yes	Yes
SSTL18[2]	Yes	Yes	Yes	Yes
HSTL18[2]	Yes	Yes	Yes	Yes

1. PCI33 is supported at the bottom bank of MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices.
2. SSTL Class II and HSTL Class II are supported as input only.

Table 10-4. Differential I/O Standards Supported on Various Sides

Standard	Top	Bottom	Left	Right
LVDS output	Yes[1]	—	—	—
LVPECL33E[2]	Yes	Yes	Yes	Yes
MLVDS25E[2]	Yes	Yes	Yes	Yes
BLVDS25E[2]	Yes	Yes	Yes	Yes
RSDS25E[2]	Yes	Yes	Yes	Yes
LVDS25E[2]	Yes	Yes	Yes	Yes
SSTL25D output	Yes	Yes	Yes	Yes
SSTL18D output	Yes	Yes	Yes	Yes
HSTL18D output	Yes	Yes	Yes	Yes
LVTTL33D output	Yes	Yes	Yes	Yes
LVCMOS33D output	Yes	Yes	Yes	Yes
LVCMOS25D output	Yes	Yes	Yes	Yes
LVCMOS18D output	Yes	Yes	Yes	Yes
LVCMOS15D output	Yes	Yes	Yes	Yes
LVCMOS12D output	Yes	Yes	Yes	Yes
LVDS input	Yes	Yes	Yes	Yes
LVPECL33 input	Yes	Yes	Yes	Yes
MLVDS25 input	Yes	Yes	Yes	Yes
BLVDS25 input	Yes	Yes	Yes	Yes
RSDS25 input	Yes	Yes	Yes	Yes
SSTL25D input	Yes	Yes	Yes	Yes
SSTL18D input	Yes	Yes	Yes	Yes
HSTL18D input	Yes	Yes	Yes	Yes
LVTTL33D input	Yes	Yes	Yes	Yes
LVCMOS33D input	Yes	Yes	Yes	Yes
LVCMOS25D input	Yes	Yes	Yes	Yes
LVCMOS18D input	Yes	Yes	Yes	Yes
LVCMOS15D input	Yes	Yes	Yes	Yes
LVCMOS12D input	Yes	Yes	Yes	Yes

1. True LVDS output is supported at the top bank of MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices.
2. Emulated output standards are denoted with a trailing “E” in the name of the standard.

Power Supply Requirements

The MachXO2 device family has a simplified power supply scheme for sysIO buffers. The core power V_{CC} and the bank power V_{CCIO} are the two main power supplies. A MachXO2 device can be powered and operated with a single power supply by connecting V_{CC} and V_{CCIO} to nominal voltages of 1.2V. The JTAG programming pins are powered by V_{CCIO} in bank 0 where the JTAG pins reside. All the user sysIOs have a weak pull-down after power-up is complete and before the device configuration is done.

V_{CCIO} Requirement for I/O Standards

Each I/O bank of a MachXO2 device has a separate V_{CCIO} supply pin that can be connected to 1.2V, 1.5V, 1.8V, 2.5V or 3.3V. This voltage is used to power the output I/O standard and source the drive strength for the output. In addition to this, V_{CCIO} also powers the ratioed input buffers such as LVTTL, LVCMOS and PCI. This ensures that the threshold of the input buffers tracking the V_{CCIO} voltage level.

For LVCMOS I/O types, mixed input voltage support is allowed in each I/O bank as long as the V_{CCIO} requirement for the input or output I/O standard is the same, or when all inputs in the bank are within the over-drive or under-drive range as specified in Tables 10-5 and 10-6. Two other options exist to further increase the input receiver flexibility. One is to configure an I/O to be a 1.2V ratioed input buffer, regardless of the bank V_{CCIO} voltage. This is possible because the MachXO2 sysIO buffer has two ratioed input buffers connected to V_{CCIO} and V_{CC} in parallel. The other option is to use the input reference voltage pin to set the input threshold for LVCMOS standards that are not covered by the V_{CCIO} of the bank.

Table 10-5. V_{CCIO} for Same Bank LVCMOS/LVTTL Input/Output Requirements[1]

I/O Type	Bank Restrictions
LVCMOS12	Outputs require V_{CCIO} = 1.2V Inputs available in all V_{CCIO} levels
LVCMOS15	Outputs require V_{CCIO} = 1.5V Inputs available in all V_{CCIO} levels.
LVCMOS15R33[2, 3]	Inputs only, require V_{CCIO} = 3.3V and V_{REF} = 0.75V
LVCMOS15R25[2, 3]	Inputs only, require V_{CCIO} = 2.5V and V_{REF} = 0.75V
LVCMOS18	Outputs require V_{CCIO}= 1.8V Inputs require V_{CCIO} = 1.5V, 1.8V, 2.5V, or 3.3V
LVCMOS18R33[2, 3]	Inputs only, require V_{CCIO} = 3.3V and V_{REF} = 0.9V
LVCMOS18R25[2, 3]	Inputs only, require V_{CCIO} = 2.5V and V_{REF} = 0.9V
LVCMOS25	Outputs require V_{CCIO} = 2.5V Inputs require V_{CCIO} = 1.5V, 1.8V, 2.5V, or 3.3V.
LVCMOS25R33[2, 3]	Inputs only, require V_{CCIO} = 3.3V and V_{REF} = 1.25V
LVCMOS33	Outputs require V_{CCIO} = 3.3V Inputs require V_{CCIO} = 1.5V, 1.8V, 2.5V, or 3.3V
LVTTL33	Outputs require V_{CCIO} = 3.3V Inputs require V_{CCIO} = 1.5V, 1.8V, 2.5V, or 3.3V
PCI33	Inputs and outputs both require V_{CCIO}= 3.3V

1. Certain I/O type and bank V_{CCIO} combinations may cause higher DC current. For more details refer to Table 10-6. Use Power Calculator to get power estimation for I/O types.
2. The HYSTERESIS option and BUS KEEPER option are not available for these I/O types.
3. Since only one V_{REF} signal can be supported in each I/O bank, only one of these I/O standards can be used in each I/O bank.

Table 10-6. Mixed Voltage Support for LVCMOS and LVTTL I/O Types

V_{CCIO}	Inputs					Outputs				
	1.2V	1.5V	1.8V	2.5V	3.3V	1.2V	1.5V	1.8V	2.5V	3.3V
1.2V	YES	YES[6]				YES				
1.5V	YES[1]	YES	YES[6]	YES[6]	YES[6]		YES			
1.8V	YES[1]	YES[5]	YES	YES[6]	YES[6]			YES		
2.5V	YES[1]	YES[2, 5, 7]	YES[3, 5, 7]	YES	YES[6]				YES	
3.3V	YES[1]	YES[2, 5, 7]	YES[3, 5, 7]	YES[4, 5, 7]	YES					YES

1. Leakage will occur if bus hold or weak pull-up is turned on.
2. This input standard can be supported using the ratioed input buffer in under-drive conditions or using the I/O types LVCMOS15R25 or LVCMOS15R33 with the referenced input buffer.
3. This input standard can be supported using the ratioed input buffer in under-drive conditions or using the I/O type LVCMOS18R25 or LVCMOS18R33 with the referenced input buffer.
4. This input standard can be supported using the ratioed input buffer in under-drive conditions or using the I/O type LVCMOS25R33 with the referenced input buffer.
5. Under-drive condition when using the ratioed input buffer and the input standard voltage is below Vccio
   a. Under-drive causes higher DC current when the IO is at logic high. It is recommended to use Power Calculator to estimate the power consumption under such condition.
   b. Hysteresis is not supported. In the Diamond software, HYSTERESIS must be set to NA.
   c. CLAMP is not supported. In the Diamond software, CLAMP must be set to OFF.
   d. IO termination is not supported. In the Diamond software, PULLMODE must be set to NONE.
6. Over-drive condition when using the ratioed input buffer and the input standard voltage is above Vccio
   a. Hysteresis is not supported. In the Diamond software, HYSTERESIS must be set to NA.
   b. CLAMP is not supported. In the Diamond software, CLAMP must be set to OFF.
   c. IO termination is not supported. In the Diamond software, PULLMODE must be set to NONE.
7. Ratioed input buffer in under-drive conditions is preferred over referenced input buffer due to lower power requirement for the ratioed input buffer.
8. When using the ratioed input buffers in under-drive or over-drive conditions, the HYSTERESIS setting shall be NA, the CLAMP setting shall be OFF, and the UP and KEEPER PULLMODE settings are not supported.

For differential input standards, certain mixed voltage support is allowed in the architecture as shown in Table 10-7.

Table 10-7. Mixed Voltage Support for Differential Input Standards

V_{CCIO}	Differential Inputs					
	LVDS, LVPECL33, MLVDS25, BLVDS25, RSDS25	SSTL25D	SSTD18D, HSTL18D	LVTTL33D, LVCMOS33D	LVCMOS25D, LVCMOS15D, LVCMOS12D	LVCMOS18D
1.2V						
1.5V						
1.8V			YES			YES
2.5V	YES	YES	YES		YES	YES
3.3V	YES	YES	YES	YES	YES	YES

Input Reference Voltage

Each I/O bank supports one reference voltage (V_{REF}). Any I/O in the bank can be configured as the input reference voltage pin. This pin is a regular I/O if it is not used as reference voltage input. To support SSTL and HSTL inputs, the reference voltage is set to half of the V_{CCIO} level. The input reference voltage can also be generated internally from the V_{REF} generator. Again, there is one V_{REF} generator per bank and its programmable settings include OFF, 45% of V_{CCIO}, 50% of V_{CCIO}, and 55% of V_{CCIO}. Programming of the internal V_{REF} generator and the external V_{REF} pin cannot be set at the same time for a particular bank because there is only one V_{REF} bus per bank.

sysIO Buffer Configuration

MachXO2 devices have three types of general-purpose sysIO buffer pairs to support a variety of single-ended and differential standards. Each sysIO buffer pair is made of two PIO buffers. PIO A and B pads form the primary pair, and PIO C and D pads form the alternate pair. Pads A and C of the pair are considered the “true” pad, while pads B and D are considered the “comp” pad. The “true” pad is associated with the positive side of the differential signal, while the “comp” pad is associated with the negative side of the differential signal.

All the PIOs support programmable clamp and bus maintenance circuitry to allow a weak pull-up, or a weak pull-down, or a weak bus keeper. The base sysIO buffer pair is used on all sides of the smaller devices, and on the left and right sides of MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices. The LVDS sysIO buffer pairs have additional LVDS output drivers in the primary PIO pairs. They are used on the top bank of the MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices. The bottom sysIO buffer pairs have additional 100ohm termination resistors between the “true” and “comp” pads. The bottom sysIO buffer pairs also support PCI clamp. They are supported on the bottom I/O bank of the MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices.

LVCMOS Buffer Configurations

The LVCMOS buffers are built on the base sysIO buffer pairs. These LVCMOS buffers can be configured in a variety of modes to support common circuit design needs.

Bus Maintenance circuit

Each pad has a weak pull-up, weak pull-down, and weak bus-keeper capability. These are selected with ON and OFF programmability. The pull-up and pull-down settings offer a fixed characteristic, which is useful in creating wired logic such as wired ORs. The bus-keeper option latches the signal in the last driven state, holding it at a valid level with minimal power dissipation. Input leakage can be minimized by turning off the bus maintenance circuitry. However, it is important to ensure that inputs are driven to a known state to avoid unnecessary power dissipation in the input buffer. The bus maintenance circuit is available for single-ended ratioed I/O standards.

Programmable Drive Strength

All single-ended drivers have programmable drive strength. This option can be set for each I/O independently. The drive strengths available for each I/O standard can be found in Table10-9. The MachXO2 programmable drive architecture is guaranteed with minimum drive strength for each drive setting. The V/I curves in the data sheet provide details of output driving capability versus the output load. This information, together with the current per bank and the package thermal limit current, should be taken into consideration when selecting the drive strength.

Input Hysteresis

All ratioed input receivers, except LVCMOS12, support input hysteresis. The input hysteresis for the LVCMOS33, LVCMOS25, LVCMOS18 and LVCMOS15 have two settings for flexibility. The ratioed input receivers have no input hysteresis when they are operated in under-drive or over-drive input conditions as shown in Tables 10-5 and 10-6.

Programmable Slew Rate

The single-ended output buffer for each device I/O pin has programmable output slew rate control that can be configured for either low noise (SLEWRATE=SLOW) or high speed (SLEWRATE=FAST) performance. Each I/O pin has an individual slew rate control. This slew rate control affects both the rising edge and the falling edges. The rise and fall ramp rates for each I/O standard can be found in the in the device IBIS file for a given I/O configuration.

Tri-state and Open Drain Control

Each single-ended output driver has a separate tri-state control in addition to the global tri-state control for the device. The single-ended output drivers also support open drain operation on each I/O independently. The open drain output is typically pulled up externally and only the sink current specification is maintained.

PCI Support with PCI Clamp

The bottom sysIO buffer pair supports an optional PCI clamp diode that may be programmed individually.

This is only supported at the bottom edge of MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices. The PCI clamp supports a larger clamping current than the programmable clamp available on all other sides of the devices.

Complementary Outputs

Each sysIO buffer pair has built-in complementary circuit that can optionally be driven by the complement of the data that drives the single-ended driver associated with the true pad. This allows a pair of single-ended drivers to be used to drive complementary outputs with the lowest possible skew between the signals.

Differential Buffer Configurations

The base sysIO buffer pair supports differential input standards. Its complementary outputs support SSTL and HSTL differential output standards. The top and bottom edges of MachXO2-640U, MachXO2-1200/U and higher density devices support some additional functions over those supported by the base sysIO buffer pairs.

Differential Receivers

All the sysIO buffer pairs support differential input on all edges of the device. When a sysIO buffer pair is configured as differential receiver, the input hysteresis and the bus maintenance capabilities will be disabled for the buffer.

On-Chip Input Termination

The MachXO2 device supports on-chip 100 ohm (nominal) input differential termination on the bottom edge of MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices. The termination is available on all input PIO pairs of the bottom edge and is programmable.

Emulated Differential Outputs

All sysIO buffer pairs support complementary outputs as described above. This feature can be used to drive complementary SSTL or HSTL signals as required for differential SSTL and HSTL standards. It can also be used together with off-chip resistor networks for emulating the differential output standards such as LVPECL, MLVDS, BLVDS, and RSDS differential standards. When a sysIO buffer pair is configured as differential transmitter, the bus maintenance and open drain capabilities will be disabled. All single-ended sysIO buffers pairs in the MachXO2 family can support emulated differential output standards.

True Differential Output And Output Drive

MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices support true differential output drivers on the top edge of these devices. These true differential outputs are only available on the primary PIO pairs. The output driver has a fixed common mode of 1.2V and a programmable drive current of 1.25 mA, 2.5 mA, 2.0 mA or 3.5 mA. Only one true LVDS differential drive setting is available at a time. All true LVDS differential drivers on the top edge must be programmed to have the same drive strength. The bank V_{CCIO} for true differential output can be 2.5V or 3.3V.

Software sysIO Attributes

The sysIO attributes or primitives must be used in the Lattice development software to control the functions and capabilities of the sysIO buffers. sysIO attributes or primitives can be specified in the HDL source code, in the Lattice Diamond™ Spreadsheet View GUI, or in the ASCII preference file (.lpf) file directly. Appendices A, B and C list examples of using such attributes in different environments. This section describes each of these attributes in detail.

HDL Attributes

All the attributes discussed in this section, except two, can be used in the HDL source code to direct the sysIO buffer functionality.

I/O_TYPE

This attribute is used to set the sysIO standard for an I/O. The V_{CCIO} required to set these I/O standards are embedded in the attribute names. The BANK VCCIO attribute is used to specify allowed V_{CCIO} combinations for each I/O type. Table shows the valid I/O types for the MachXO2 family.

Table 10-8. Supported I/O Types

sysIO Signaling Standard	I/O_TYPE
LVDS 2.5V	LVDS25
Emulated LVDS 2.5V[1]	LVDS25E
RSDS	RSDS25
Emulated RSDS[1]	RSDS25E
Bus LVDS 2.5V	BLVDS25
Emulated Bus LVDS 2.5V[1]	BLVDS25E
MLVDS 2.5V	MLVDS25
Emulated MLVDS 2.5V[1]	MLVDS25E
LVPECL 3.3V	LVPECL33
Emulated LVPECL 3.3V[1]	LVPECL33E
SSTL 25 Class I	SSTL25_I
SSTL 25 Class II[2]	SSTL25_II
SSTL 25 Class I differential[3]	SSTL25D_I
SSTL 25 Class II differential[2, 3]	SSTL25D_II
SSTL 18 Class I	SSTL18_I
SSTL 18 Class II[2]	SSTL18_II
SSTL 18 Class I differential[3]	SSTL18D_I
SSTL 18 Class II differential[2, 3]	SSTL18D_II
HSTL 18 Class I	HSTL18_I
HSTL 18 Class II[2]	HSTL18_II
HSTL 18 Class I differential[3]	HSTL18D_I
HSTL 18 Class II differential[2,3]	HSTL18D_II
PCI 3.3V	PCI33
LVTTL 3.3V	LVTTL33
LVTTL 3.3V differential[3]	LVTTL33D
LVCMOS 3.3V	LVCMOS33
LVCMOS 3.3V differential[3]	LVCMOS33D
LVCMOS 2.5V (default)	LVCMOS25
LVCMOS 2.5V differential[3]	LVCMOS25D
LVCMOS 2.5V in a 3.3V VCCIO bank[4]	LVCMOS25R33
LVCMOS 1.8V	LVCMOS18
LVCMOS 1.8V differential[3]	LVCMOS18D
LVCMOS 1.8V in 3.3V VCCIO bank[4]	LVCMOS18R33
LVCMOS 1.8V in 2.5V VCCIO bank[4]	LVCMOS18R25
LVCMOS 1.5V	LVCMOS15
LVCMOS 1.5V differential[3]	LVCMOS15D
LVCMOS 1.5V in 3.3V VCCIO bank[4]	LVCMOS15R33
LVCMOS 1.5V in 2.5V VCCIO bank[4]	LVCMOS15R25
LVCMOS 1.2V	LVCMOS12
LVCMOS 1.2V differential[3]	LVCMOS12D

1. These differential output standards are emulated by using a complementary LVCMOS driver pair together with an external resistor pack.
2. Only input mode is supported. Output or bidirectional modes are not supported for these I/O types.
3. These differential standards are implemented by using a complementary LVCMOS driver pair.
4. These are input only and require V_{REF} to be set to certain value to allow the specified I/O types to be used.

DRIVE
The DRIVE strength attribute is available for the output and bidirectional I/O standards. The default drive value depends on the I/O standard used. Table 10-9 shows the supported drive strength for the single-ended I/O types under designated V_{CCIO} conditions.

Table 10-9. Output Drive Capability for Ratioed sysIO Standards

Drive Strength (mA)	I/O Type					
	LVCMOS12	LVCMOS15	LVCMOS18	LVCMOS25	LVCMOS33	LVTTL33
2	YES					
4		YES	YES	YES	YES	YES
6	YES					
8		YES	YES	YES	YES	YES
12			YES	YES	YES	YES
16				YES	YES	YES
24					YES	YES

DIFFDRIVE
The DIFFDRIVE strength attribute is available for the true LVDS output standard. All true LVDS differential drivers on the top edge must be programmed to have the same drive strength. The DIFFDRIVE value will be listed in the DRIVE column of Design Planner since this value is only valid for LVDS25 outputs.

Values: 1.25, 2.0, 2.5, 3.5, NA
Default: 3.5

PULLMODE
The PULLMODE option can be enabled or disabled independently for each I/O. When the user selects OPEND-RAIN=ON, the PULLMODE for the output standard is default to NONE. If using LVCMOS I/O type in an under-drive or over-drive mode, the UP and KEEPER settings are not supported The FAILSAFE option is available for MLVDS25E bi-directional mode only.

Values: UP, DOWN, NONE, KEEPER, FAILSAFE
Default: DOWN for LVTTL, LVCMOS, and PCI; all others NONE

CLAMP
The CLAMP option can be enabled or disabled independently for each I/O. The available settings on the bottom edge of MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices is PCI or OFF. All other I/O have ON or OFF settings for this attribute.

Values: OFF, ON, PCI
Default: OFF

HYSTERESIS
The ratioed input buffers have two input hysteresis settings. The HYSTERESIS option can be used to change the amount of hysteresis for the LVTTL and LVCMOS input and bidirectional I/O standards, except for the LVCMOS12 inputs. The LVCMOS12 inputs do not support HYSTERESIS.

The LVCMOS25R33, LVCMOS18R25, LVCMOS18R33, LVCMOS15R25, and LVCMOS15R33 input types do not support HYSTERESIS. The HYSTERESIS option for each of the input pins can be set independently when it is supported for the I/O type.

Values: SMALL, LARGE, NA
Default: SMALL

VREF
The VREF option is enabled for single-ended SSTL and HSTL inputs and the referenced LVMCOS input buffers. The referenced LVMCOS input buffers are specified by choosing the I/O type as LVCMOS25R33, LVCMOS18R25, LVCMOS18R33, LVCMOS15R25, or LVCMOS15R33. The default value of NA will apply for all I/O types that do not use a VREF signal.

The VREF will default to external VREF pin for the single-ended SSTL/HSTL inputs, LVCMOS25R33, LVCMOS18R25, LVCMOS18R33, LVCMOS15R25, or LVCMOS15R33 inputs. The user may enter a VREF_NAME value in the "VREF Location(s)" pop-up window of the Spreadsheet View of the Diamond software. In doing so, the software will present the VREF_NAME as an available value in additional to the I45, I50 and I55 values in the VREF column of the Port Assignments tab of the Diamond Spreadsheet View. A pin location specified by the VREF_NAME value will be used as the VREF driver for that I/O bank. VREF_NAME is only necessary if the user wants to specify a pin to be used as an external VREF pin. Otherwise, the software will automatically assign a pin for the VREF signal.

There is only one VREF pin or internal VREF driver per I/O bank. Only one of the VREF driver settings chosen from I45, I50, I55 or VREF1_LOAD can be used in each I/O bank. This attribute can be set in the software GUI or in the ASCII preference file.

Values: OFF, I45, I50, I55, VREF_NAME
Default: NA

OPENDRAIN
The OPENDRAIN option is available for all LVTTL and LVCMOS output and bidirectional I/O standards. Each sysIO can be assigned independently to be open drain. When the OPENDRAIN attribute is used, the PULLMODE must be NONE and the CLAMP must be OFF.

Values: OFF, ON
Default: OFF

SLEWRATE
Each I/O pin has an individual slew rate control. This allows the designer to specify slew rate control on a pin-by-pin basis for outputs and bidirectional I/O pins. This is not a valid attribute for inputs or true differential outputs.

Values: FAST, SLOW, NA
Default: SLOW

DIFFRESISTOR
The bottom side I/O pins support on-chip differential input termination resistors on the MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices. The termination resistor is available for both the primary pair and the alternate pair of a sysIO. The values supported are zero (OFF) or 100 ohms.

Values: OFF, 100
Default: OFF

DIN/DOUT
The DIN/DOUT option is available for each I/O and can be configured independently. An input register is used for the input if the DIN attribute is assigned. Similarly, the software assigns an output register when the DOUT attribute is specified. By default, the software automatically assigns DIN or DOUT to input or output registers if possible.

LOC
This attribute specifies the site location for the component after the mapping process. When attached to multiple components, it indicates that these blocks are to be mapped together in the specified site. It specifies the PIC site for the pad when it is assigned to a pad. The LOC attribute can be attached to components that will end up on an I/O cell, clocks, and internal flip-flops, but it should not be attached to combinational logic that will end up on a logic cell; doing so could fail to generate a locate preference. The LOC attribute overrides register ordering.

Bank VCCIO

This attribute is necessary to verify the valid I/O types for a bank, to determine which input buffer to use, and to set the correct drive strength for the applicable I/O types. Since the I/O bank information is not required at the HDL level, this attribute is available through either the Diamond software's Spreadsheet View or in the ASCII preference file. Values: AUTO, 3.3, 2.5, 1.8, 1.5, 1.2. Default: AUTO.

sysIO Primitives

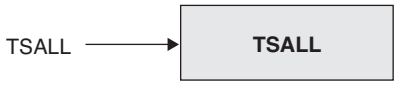
There are many sysIO primitives in the software library. A few are selected to be discussed in this section because some sysIO capabilities can only be utilized through instantiating the primitives in the HDL source code.

Tri-State All (TSALL)

The MachXO2 device supports the TSALL function that is used to enable or disable the tristate control to all the output buffers. The user can choose to assign any general purpose I/O pin to control the TSALL function since there is no dedicated TSALL pin. The TSALL primitive must be instantiated in the source code in order to enable the TSALL function. The input of the primitive can be assigned to an input pin or to an internal signal.

A value of TSALL=1 will tri-state all outputs but the outputs will be under individual OE control when TSALL=0.

Figure 10-5. TSALL Primitive



Fixed Data Delay (DELAYE)

This primitive supports up to 32 steps of static delay for all sysIO buffers in all banks of a MachXO2 device. Refer to the MachXO2 Family Data Sheet for delay step values. Although users can choose USER_DEFINED mode to set input delay, this primitive is primarily used by pre-defined source synchronous interfaces as described in TN1203, MachXO2 High-Speed Source Synchronous and Memory Interfaces.

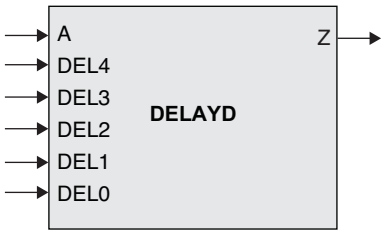
Figure 10-6. DELAYE Primitive and Associated Attributes

Attribute	Description	Value	Software Default
DEL_MODE	Fixed delay value depending on interface or user-defined delay values	SCLK_ZEROHOLD ECLK_ALIGNED ECLK_CENTERED SCLK_ALIGNED SCLK_CENTERED USER_DEFINED	USER_DEFINED
DEL_VALUE	User-defined value	DELAY0…DELAY31	DELAY0

Dynamic Data Delay (DELAYD)

This primitive supports dynamic delay for the sysIO buffers in the bottom bank (Bank 2) of MachXO2-640U, MachXO2-1200/U and larger devices. The 5-bit inputs can be controlled by user logic to modify the delay during the device operation.

Figure 10-7. DELAYD Primitive



Design Consideration and Usage

This section summarizes the MachXO2 designs rules and considerations that have been discussed in detail in previous sections. Table 10-6 lists the miscellaneous I/O features on each side of a MachXO2 device.

sysIO Buffer Features Common to All MachXO2 Devices

1. All banks support true differential inputs.
2. All banks support emulated differential outputs using external resistors and complementary LVCMOS outputs. Emulated differential output buffers are supported on both primary and alternate pairs.
3. All banks have programmable I/O clamps but they are not PCI compliant clamps.
4. All banks support weak pull-up, pull-down, and bus-keeper (bus hold latch) settings on each I/O independently.
5. V_{CCIO} voltage levels, together with the selected I/O types, determine the characteristics of an I/O, such as the pull mode, hysteresis, clamp behavior, and drive strength, supported in a bank. Multiple input standards can be supported in a bank through under-drive or over-drive conditions. Only one alternative input standard can be supported through the bank VREF setting (for example, LVCMOS25R33 requires V_{REF} to be 1.25V in a 3.3V V_{CCIO} bank). Each bank also supports 1.2V inputs regardless of the V_{CCIO} setting of the bank.
6. Each bank supports one V_{CCIO} signal.
7. Each bank supports one V_{REF} signal, whether it is from an external pin or from the internal V_{REF} generator.

sysIO Buffer Rules Specific to MachXO2-256 and MachXO2-640

1. Does not support true differential output buffers.
2. Does not support internal 100 ohm differential input terminations.
3. Does not support PCI clamps.

sysIO Buffer Rules Specific to MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000

1. Only Bank 0 (top side) supports true differential output buffers with programmable drive strengths. Only the primary pair supports true differential output buffers.
2. Only Bank 2 (bottom side) supports internal 100 ohm differential input terminations.
3. Only Bank 2 (bottom side) supports PCI compliant clamps.

Table 10-10. Miscellaneous I/O Features on Each Device Edge

Feature	Top	Bottom	Left	Right
100 Ohm Differential Resister	—	Yes[1]	—	—
Hot Socket	Yes	Yes	Yes	Yes
Clamp[3]	Yes	Yes	Yes	Yes
PCI Compliant Clamp	—	Yes[1]	—	—
Weak Pull-up[3]	Yes	Yes	Yes	Yes
Weak Pull-down[2]	Yes	Yes	Yes	Yes
Bus Keeper[3]	Yes	Yes	Yes	Yes
Input Hysteresis[3]	Yes	Yes	Yes	Yes
Slew Rate Control	Yes	Yes	Yes	Yes
Open Drain	Yes	Yes	Yes	Yes

1. Supported by MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000, and MachXO2-7000 devices.
2. Software default setting
3. I/O characteristic under special conditions
   a. HYSTERESIS option is not available for LVCMOS12.
   b. HYSTERESIS option and BUS KEEPER option are not available for referenced input standards.
   c. When using the ratioed input buffers in under-drive or over-drive conditions, the HYSTERESIS setting shall be NA, the CLAMP setting shall be OFF, and the UP and KEEPER PULLMODE settings are not supported.
   d. HYSTERESIS and the bus maintenance capabilities are disabled for differential receivers.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
January 2011	01.1	Updated for ultra-high I/O (“U”) devices.
April 2011	01.2	Updated for Lattice Diamond design software.
July 2011	01.3	Updated sysIO Banking Scheme text section with information on migrating from MachXO2-1200-R1 to Standard (non-R1) devices.
February 2012	01.4	Updated document with new corporate logo.
		Document status changed from Preliminary to Final.
March 2013	01.5	Updated footnotes in the Mixed Voltage Support for LVCMOS and LVTTL I/O Types table.
		Added information on LVCMOS Buffer Configurations - Programmable Slew Rate.

Appendix A. sysIO HDL Attributes

The sysIO attributes can be used directly in the HDL source codes. This section provides a list of sysIO attributes supported by the MachXO2 PLD family. The correct syntax and examples for the Synplify® synthesis tool are provided here for reference.

Attributes in VHDL Language

Syntax

Table 10-11. VHDL Attribute Syntax

Attribute	Syntax
I/O_TYPE	attribute I/O_TYPE: string; attribute I/O_TYPE of Pinname: signal is “I/O_TYPE Value”;
DRIVE	attribute DRIVE: string; attribute DRIVE of Pinname: signal is “Drive Value”;
DIFFDRIVE	attribute DRIVE: string; attribute DRIVE of Pinname: signal is “Diffdrive Value”;
DIFFRESISTOR	attribute DIFFRESISTOR: string; attribute DIFFRESISTOR of Pinname: signal is “DIFFRESISTOR Value”;
CLAMP	attribute CLAMP: string; attribute CLAMP of Pinname: signal is “Clamp Value”;
HYSTERESIS	attribute HYSTERESIS: string; attribute HYSTERESIS OF Pinname: signal is “Hysteresis Value”;
VREF	NA
PULLMODE	attribute PULLMODE: string; attribute PULLMODE of Pinname: signal is “Pullmode Value”;
OPENDRAIN	attribute OPENDRAIN: string; attribute OPENDRAIN of Pinname: signal is “OpenDrain Value”;
SLOWSLEW	attribute PULLMODE: string; attribute PULLMODE of Pinname: signal is “Slewrate Value”;
DIN	attribute DIN: string; attribute DIN of Pinname: signal is “value “;
DOUT	attribute DOUT: string; attribute DOUT of Pinname: signal is “value “;
LOC	attribute LOC: string; attribute LOC of Pinname: signal is “Pin locations”;
BANK VCCIO	NA

Examples

I/O_TYPE

```
 --***Attribute Declaration***
ATTRIBUTE I/O_TYPE: string;
 --***I/O_TYPE assignment for I/O Pin***
ATTRIBUTE I/O_TYPE OF portA: SIGNAL IS "PCI33";
ATTRIBUTE I/O_TYPE OF portB: SIGNAL IS "LVCMOS33";
ATTRIBUTE I/O_TYPE OF portC: SIGNAL IS "SSTL18_I";
ATTRIBUTE I/O_TYPE OF portD: SIGNAL IS "LVDS25";
```

DRIVE

```
--***Attribute Declaration***
ATTRIBUTE DRIVE: string;
--***DRIVE assignment for I/O Pin***
ATTRIBUTE DRIVE OF portB: SIGNAL IS “8”;
```

DIFFDRIVE

```
 --***Attribute Declaration***
ATTRIBUTE DIFFDRIVE: string;
--*** DIFFDRIVE assignment for I/O Pin***
ATTRIBUTE DIFFDRIVE OF portD: SIGNAL IS “2.0”;
```

DIFFRESISTOR

```
 --***Attribute Declaration***
ATTRIBUTE DIFFRESISTOR: string;
--*** DIFFRESISTOR assignment for I/O Pin***
ATTRIBUTE DIFFRESISTOR OF portD: SIGNAL IS "100";
```

CLAMP

```
--***Attribute Declaration***
ATTRIBUTE CLAMP: string;
--*** CLAMP assignment for I/O Pin***
ATTRIBUTE CLAMP OF portA: SIGNAL IS “PCI33”;
```

HYSTERESIS

```
--***Attribute Declaration***
ATTRIBUTE HYSTERESIS: string;
--*** HYSTERESIS assignment for Input Pin***
ATTRIBUTE HYSTERESIS OF portA: SIGNAL IS " LARGE ";
```

PULLMODE

```
--***Attribute Declaration***
ATTRIBUTE PULLMODE : string;
--***PULLMODE assignment for I/O Pin***
ATTRIBUTE PULLMODE OF portA: SIGNAL IS “DOWN”;
ATTRIBUTE PULLMODE OF portB: SIGNAL IS “UP”;
```

OPENDRAIN

```
--***Attribute Declaration***
ATTRIBUTE OPENDRAIN: string;
--***Open Drain assignment for I/O Pin***
ATTRIBUTE OPENDRAIN OF portB: SIGNAL IS “ON”;
```

SLEWRATE

```
 --***Attribute Declaration***
ATTRIBUTE SLEWRATE : string;
--*** SLEWRATE assignment for I/O Pin***
ATTRIBUTE SLEWRATE OF portB: SIGNAL IS “FAST”;
```

DIN/DOUT

```
--***Attribute Declaration***
ATTRIBUTE din : string; ATTRIBUTE dout : string;
--*** din/dout assignment for I/O Pin***
ATTRIBUTE din OF input_vector: SIGNAL IS “TRUE “;
ATTRIBUTE dout OF output_vector: SIGNAL IS “TRUE “;
```

LOC

```
--***Attribute Declaration***
ATTRIBUTE LOC : string;
 --*** LOC assignment for I/O Pin***
ATTRIBUTE LOC OF input_vector: SIGNAL IS “E3,B3,C3 “;
```

Attributes in Verilog Language

Syntax

Table 10-12. Verilog Attribute Syntax

Attribute	Syntax
I/O_TYPE	PinType PinName /* synthesis I/O_TYPE=”I/O_Type Value”*/;
DRIVE	PinType PinName /* synthesis DRIVE=”Drive Value”*/;
DIFFDRIVE	PinType PinName /* synthesis DIFFDRIVE =” DIFFDRIVE Value”*/;
DIFFRESISTOR	PinType PinName /* synthesis DIFFRESISTOR =” DIFFRESISTOR Value”*/;
CLAMP	PinType PinName /* synthesis CLAMP =” Clamp Value”*/;
HYSTERESIS	PinType PinName /*synthesis HYSTERESIS = “Hysteresis Value” */;
VREF	N/A
PULLMODE	PinType PinName /* synthesis PULLMODE=”Pullmode Value”*/;
OPENDRAIN	PinType PinName /* synthesis OPENDRAIN =”OpenDrain Value”*/;
SLOWSLEW	PinType PinName /* synthesis SLEWRATE=”Slewrate Value”*/;
DIN	PinType PinName /* synthesis DIN= “value” */;
DOUT	PinType PinName /* synthesis DOUT= “value” */;
LOC	PinType PinName /* synthesis LOC=”pin_locations “*/;
Bank VCCIO	N/A

Examples

//I/O_TYPE, PULLMODE, SLEWRATE and DRIVE assignment

```
output portB /*synthesis I/O_TYPE=”LVCMOS33”
PULLMODE =”UP” SLEWRATE =”FAST” DRIVE =”20”*/;
output portC /*synthesis I/O_TYPE=”LVDS25” */;
```

//DIFFDRIVE

```
output portD /* synthesis I/O_TYPE="LVDS25" DIFFDRIVE="2.0"*/;
```

//DIFFRESISTOR

```
output [4:0] portA /* synthesis I/O_TYPE="LVDS25" DIFFRESISTOR ="100"*/;
```

//CLAMP

```
output portA /*synthesis I/O_TYPE=”PCI33” CLAMP =”PCI” */;
```

//HYSTERESIS

```
input mypin /* synthesis HYSTERESIS = “LARGE” */;
```

//OPENDRAIN

```
output portA /*synthesis OPENDRAIN =”ON”*/;
```

// DIN Place the flip-flops near the load input

```
 input load /* synthesis din=”” TRUE */;
```

// DOUT Place the flip-flops near the outload output

```
 output outload /* synthesis dout=”TRUE” */;
```

//LOC pin location

```
input [3:0] DATA0 /* synthesis loc=”E3,B1,F3”*/;
```

//LOC Register pin location

```
reg data_in_ch1_buf_reg3 /* synthesis loc=”R10C16” */;
```

//LOC Vectored internal bus

```
reg [3:0] data_in_ch1_reg /*synthesis loc =”R10C16,R10C15,R10C14,R10C9” */;
```

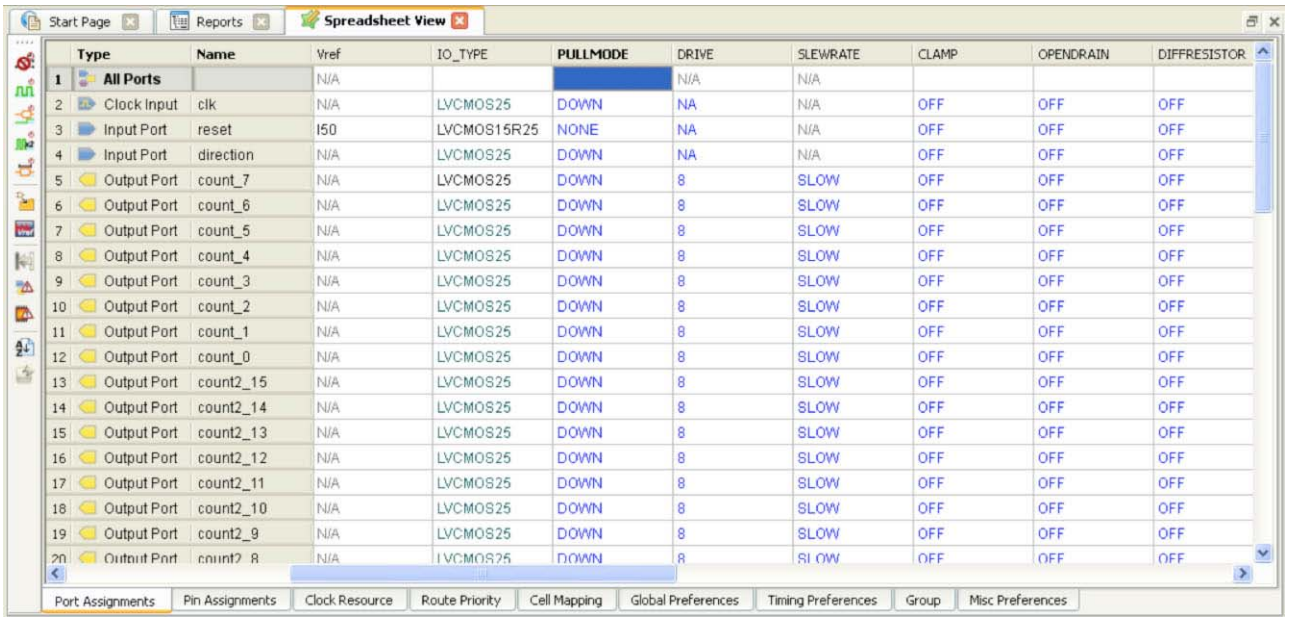
Appendix B. sysIO Attributes Using the Spreadsheet View

The sysIO buffer attributes can be assigned using the Spreadsheet View available in the Diamond design tool. The attributes that are not available as HDL attributes, such as VREF and Bank VCCIO, are available in the Spreadsheet View GUI.

The Port Assignment tab lists all the ports in a design and all the available sysIO attributes as preferences. Click on each of these cells for a list of all the valid I/O preferences for that port. Each column takes precedence over the next. Therefore, when a particular I/O_TYPE is chosen, the columns for the DRIVE, PULL-MODE, SLEW-RATE and other attributes will list the valid combinations for that I/O_TYPE. Pin locations can be locked using the Pin column of the Port Assignment tab. Right-clicking on a cell will list all the available pin locations. The Spreadsheet View can run a DRC check to check for incorrect sysIO attribute assignments.

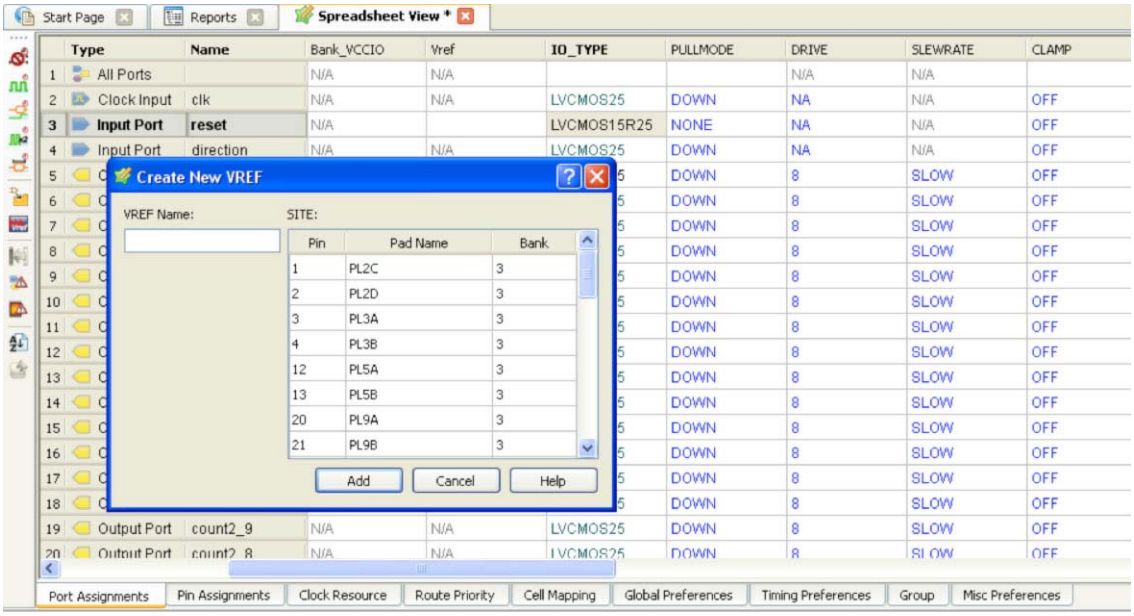
All the preferences assigned using the Spreadsheet View are written into the logical preference file (.lpf).

Figure 10-8. Port Assignment Tab of Spreadsheet View



	Type	Name	Vref	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	CLAMP	OPENDRAIN	DIFFRESISTOR
1	All Ports		N/A			N/A	N/A			
2	Clock Input	clk	N/A	LVCMOS25	DOWN	NA	N/A	OFF	OFF	OFF
3	Input Port	reset	I50	LVCMOS15R25	NONE	NA	N/A	OFF	OFF	OFF
4	Input Port	direction	N/A	LVCMOS25	DOWN	NA	N/A	OFF	OFF	OFF
5	Output Port	count_7	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
6	Output Port	count_6	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
7	Output Port	count_5	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
8	Output Port	count_4	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
9	Output Port	count_3	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
10	Output Port	count_2	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
11	Output Port	count_1	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
12	Output Port	count_0	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
13	Output Port	count2_15	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
14	Output Port	count2_14	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
15	Output Port	count2_13	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
16	Output Port	count2_12	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
17	Output Port	count2_11	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
18	Output Port	count2_10	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
19	Output Port	count2_9	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF
20	Output Port	count2_8	N/A	LVCMOS25	DOWN	8	SLOW	OFF	OFF	OFF

Figure 10-9. VREF Name and Location Pop-up Window of the Spreadsheet View



V_{REF} Assignment in the Spreadsheet View

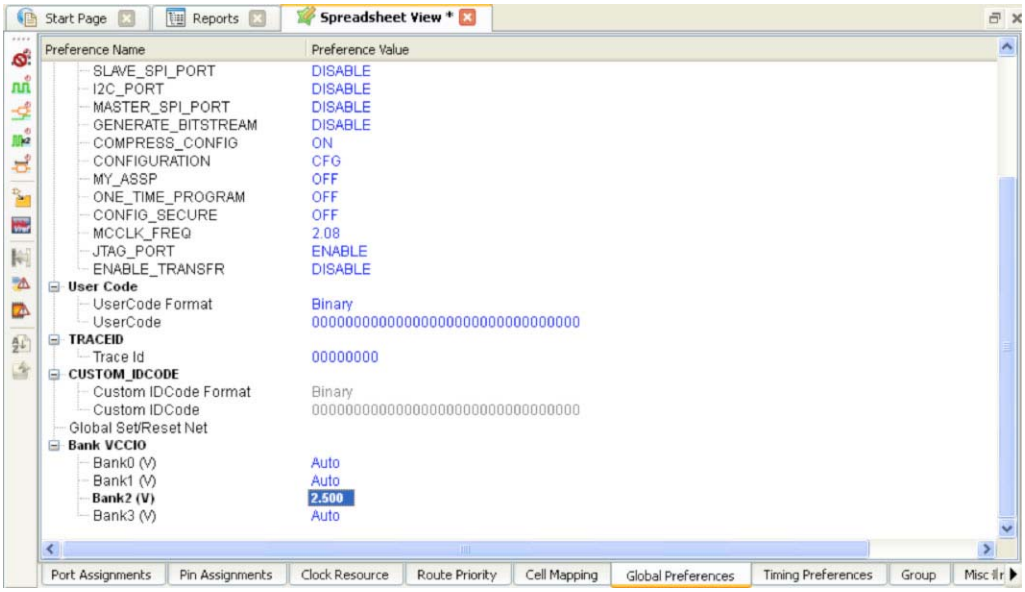
The VREF attribute can be assigned in the Spreadsheet View by clicking on the Vref Locations(s) button on the left hand side. It is required to use this button only if a specific location for the VREF driver is desired. Otherwise the software will assign the VREF driver signal to any location that does not violate the sysIO bank rules.

When the VREF_NAME is assigned to a specific pin, the software will list VREF_NAME in the VREF column of the Port Assignments tab. Both VREF_NAME and pin location will be reflected in the VREF column of the Pin Attribute sheet.

Bank V_{CCIO} Setting in the Spreadsheet View

Bank VCCIO is editable in the Global tab of the Spreadsheet View. The value of the Bank VCCIO can be chosen by the users to determine the value of VCCIO of a specific bank.

Figure 10-10. Bank VCCIO in Global Preference Tab



Appendix C. sysIO Attributes Using Preference File (ASCII File)

Designers can enter sysIO attributes directly in the preference (.lpf) file as sysIO buffer preferences. The LPF file is a post-synthesis FPGA constraint file that stores logical preferences that have been created or modified in the Spreasheet View or directly in a text editor. It also contains logical preferences originating in the HDL source. Modifying the Speadsheet View in the Diamond software will automatically update the content of the LPF file and vice versa. The settings in the Spreadsheet View are reflected in the preference file once they are saved. Details of the supported preferences and their corresponding syntax can be found in the Diamond Help System.

Implementing High-Speed Interfaces with MachXO2 Devices

April 2013 Technical Note TN1203

Introduction

In response to the increasing need for higher data bandwidth, the industry has migrated from the traditional Single Data Rate (SDR) to the Double Data Rate (DDR) architecture. SDR uses either the rising edge or the falling edge of the clock signal to transfer data, while DDR uses both edges of the clock signal for data transfer. This essentially doubles the data transmission rate using the same clock frequency because the data is transferred twice per clock cycle. The DDR clocking technique has largely been used in memory interfaces such as DDR SDRAM. As a result, DDR SDRAM memories achieve twice the bandwidth as SDR SDRAM memories without increasing the signal integrity requirements in the system.

The Lattice MachXO2™ PLD family supports high-speed interfaces for both DDR and SDR applications through built-in Programmable I/O (PIO) logic. The MachXO2 devices also have dedicated circuitry to support DDR, DDR2, and LPDDR SDRAM memory interfaces. This document focuses on the implementation of high-speed generic DDR interfaces, and memory DDR/DDR2 and LPDDR interfaces in the MachXO2 devices. It also provides guidelines for making use of the built-in capabilities of the MachXO2 devices to achieve the best performance for high-speed interfaces.

Architecture for High-Speed Interfaces

Gearing Logic Distribution

The high-speed generic DDR (GDDR) interfaces are supported through the built-in gearing logic in the Programmable I/O (PIO) cells. This gearing is necessary to support high-speed I/O while reducing the performance requirement on the FPGA fabric.

There are four gearing ratio settings available in the MachXO2 devices depending on the I/O bank locations and the logic density. The x1 gearing ratio is available in all banks for all the device densities. The x2, x4, and the 7:1 gearing ratio are available in the top and bottom banks of the MachXO2-640U, MachXO2-1200/U and higher density devices. The 7:1 gearing ratio is mainly used for video display applications. The x2/x4 gearing circuit is shared with the 7:1 circuit on both receive and the transmit sides. The right bank of the MachXO2-640U, MachXO2-1200/U and higher density devices support the memory DDR interface. The memory DDR uses x1 gearing logic in the dedicated memory PIO cells. Table 11-1 gives a breakdown of gearing logic support in the different I/O banks. Details of PIO cells can be found in the MachXO2 Family Data Sheet.

Table 11-1. Gearing Logic Distribution for MachXO2-640U, MachXO2-1200/U and Higher Density Devices

Gearing Logic	Definition	Gearing Ratio	Left	Right	Bottom	Top
DDR x1[1]	GDDR	1:2 or 2:1	Yes	Yes	Yes	Yes
Input DDR x2	GDDR	1:4	—	—	Yes	—
Input DDR x4	GDDR	1:8	—	—	Yes	—
Input DDR 7:1	GDDR	1:7	—	—	Yes	—
Output DDR x2	GDDR	4:1	—	—	—	Yes
Output DDR x4	GDDR	8:1	—	—	—	Yes
Output DDR 7:1	GDDR	7:1	—	—	—	Yes
mem DDR x1	Memory DDR	1:2 or 2:1	—	Yes	—	—

1. DDRx1 is available for all MachXO2 device densities.

Different Types of I/O Logic Cells

In order to support various gearing ratios, the MachXO2 devices support three types of PIO logic cells. These include a basic PIO cell, a memory PIO cell, and a video PIO cell.

The basic PIO cell supports traditional SDR registers and DDR x1 registers. It is available on all sides of all MachXO2 devices. The memory PIO cell supports DDR memory applications and is available on the right side of the MachXO2-640U, MachXO2-1200/U and higher density devices. The video PIO cell supports the x2/x4 and 7:1 gearing applications. They are available on MachXO2-640U, MachXO2-1200/U and larger devices on the bottom side for the receive interfaces, and on the top side for the transmit interfaces. The input and output structures of each type of PIO cell are discussed in detail in the MachXO2 Family Data Sheet. The block diagrams of the PIO cells are shown here again in this document for reference.

Figure 11-1. Basic PIO Cell Supports x1 Gearing Ratio

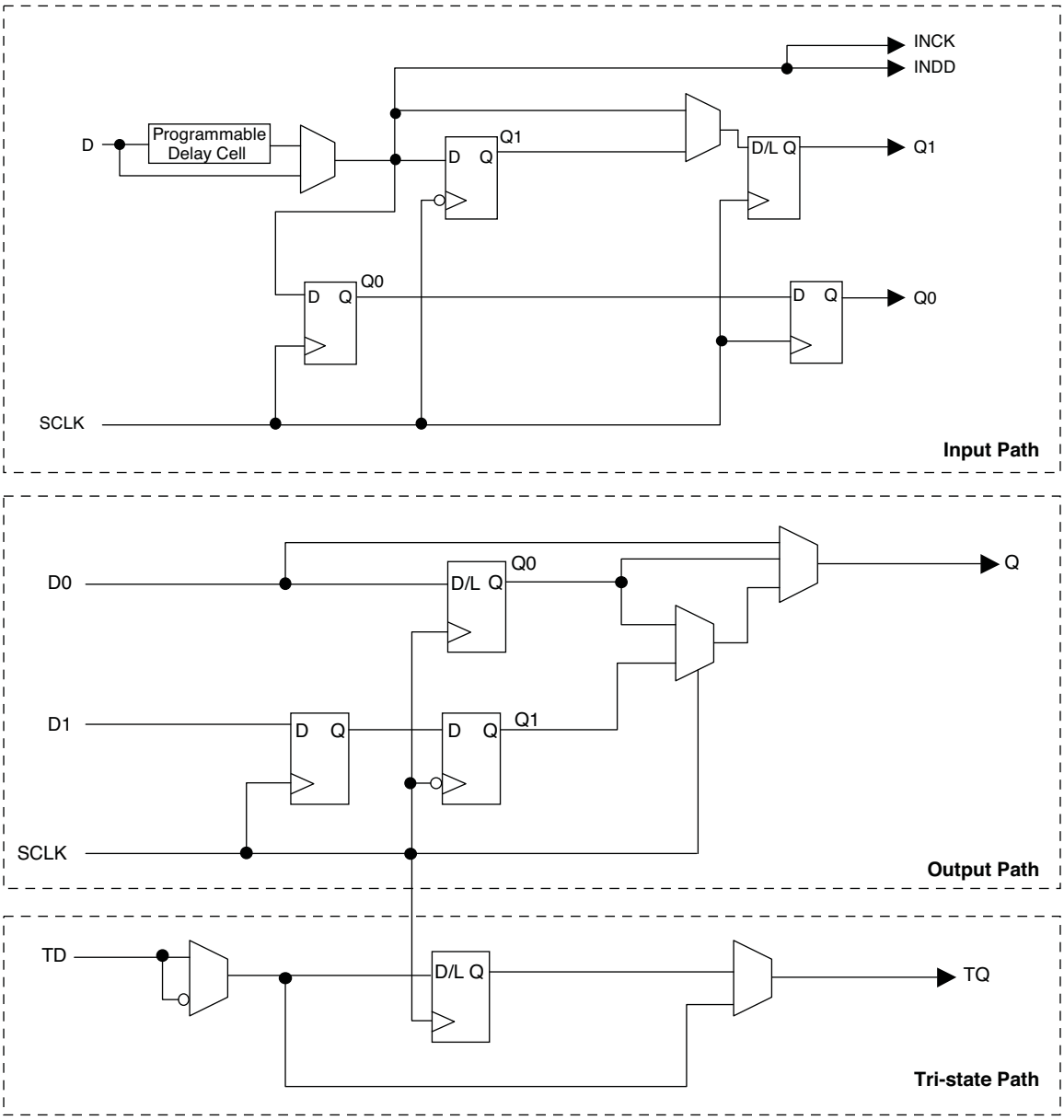


Figure 11-2. Memory PIO Cell to Support DDR Memory Applications

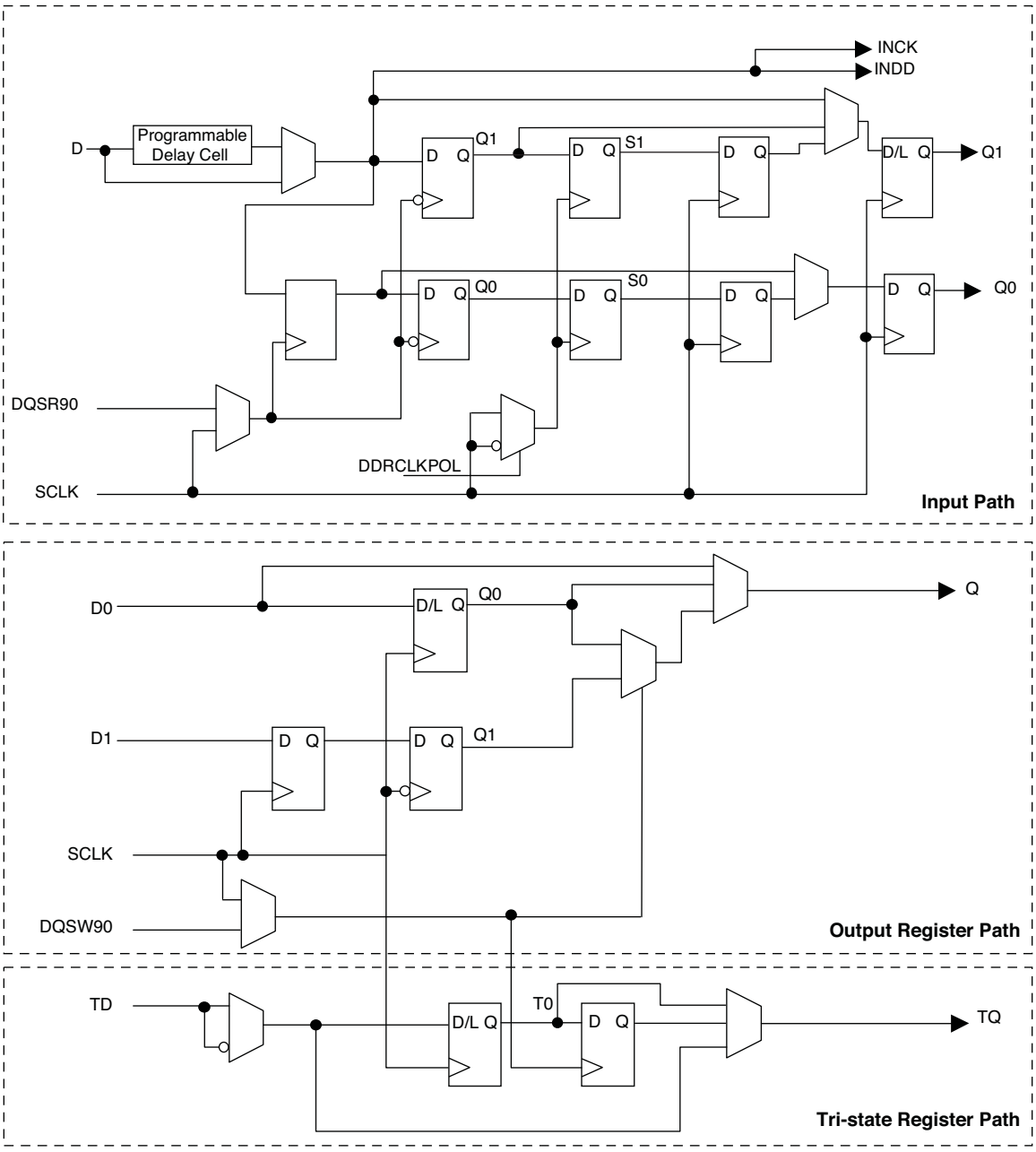
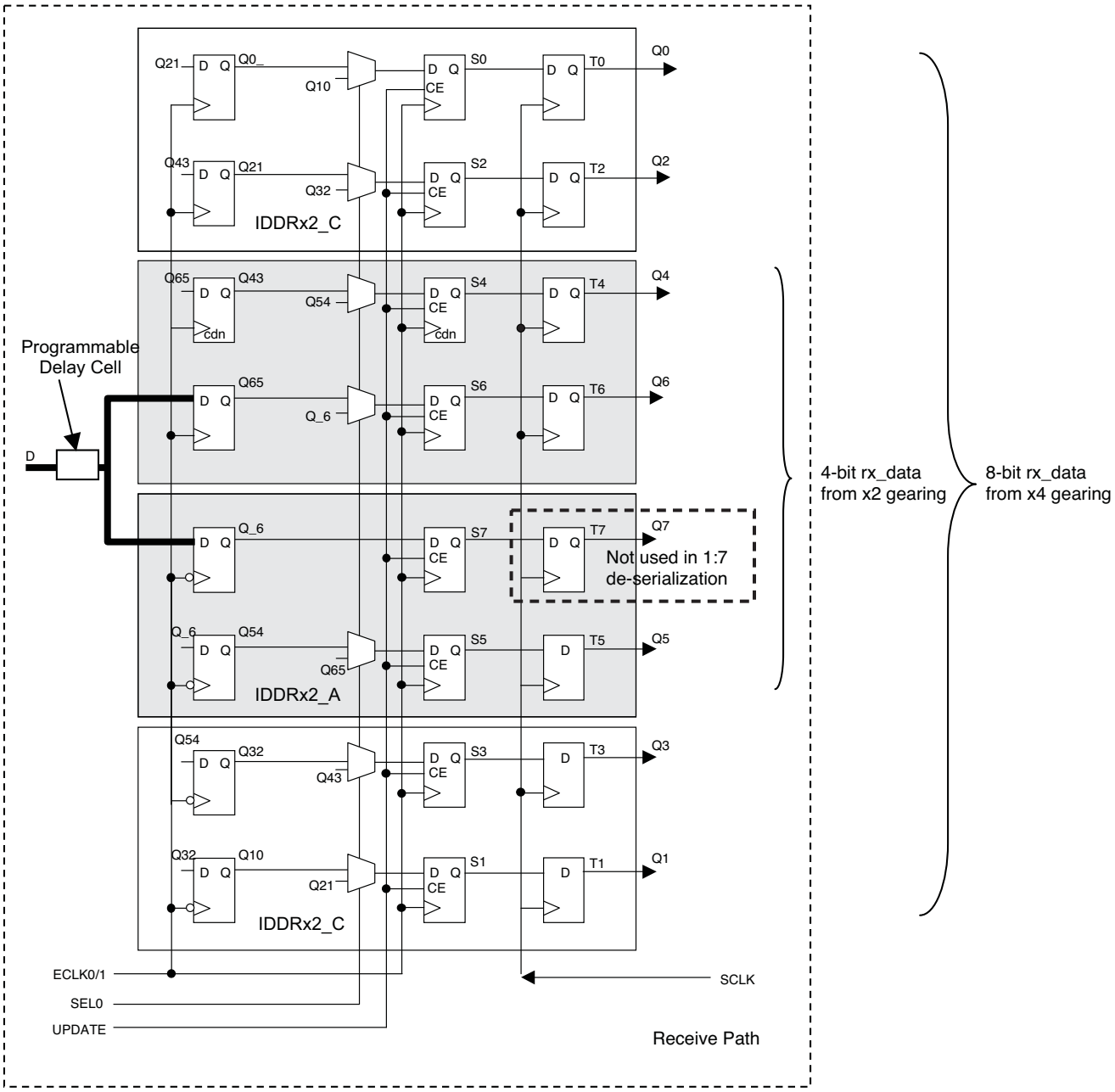
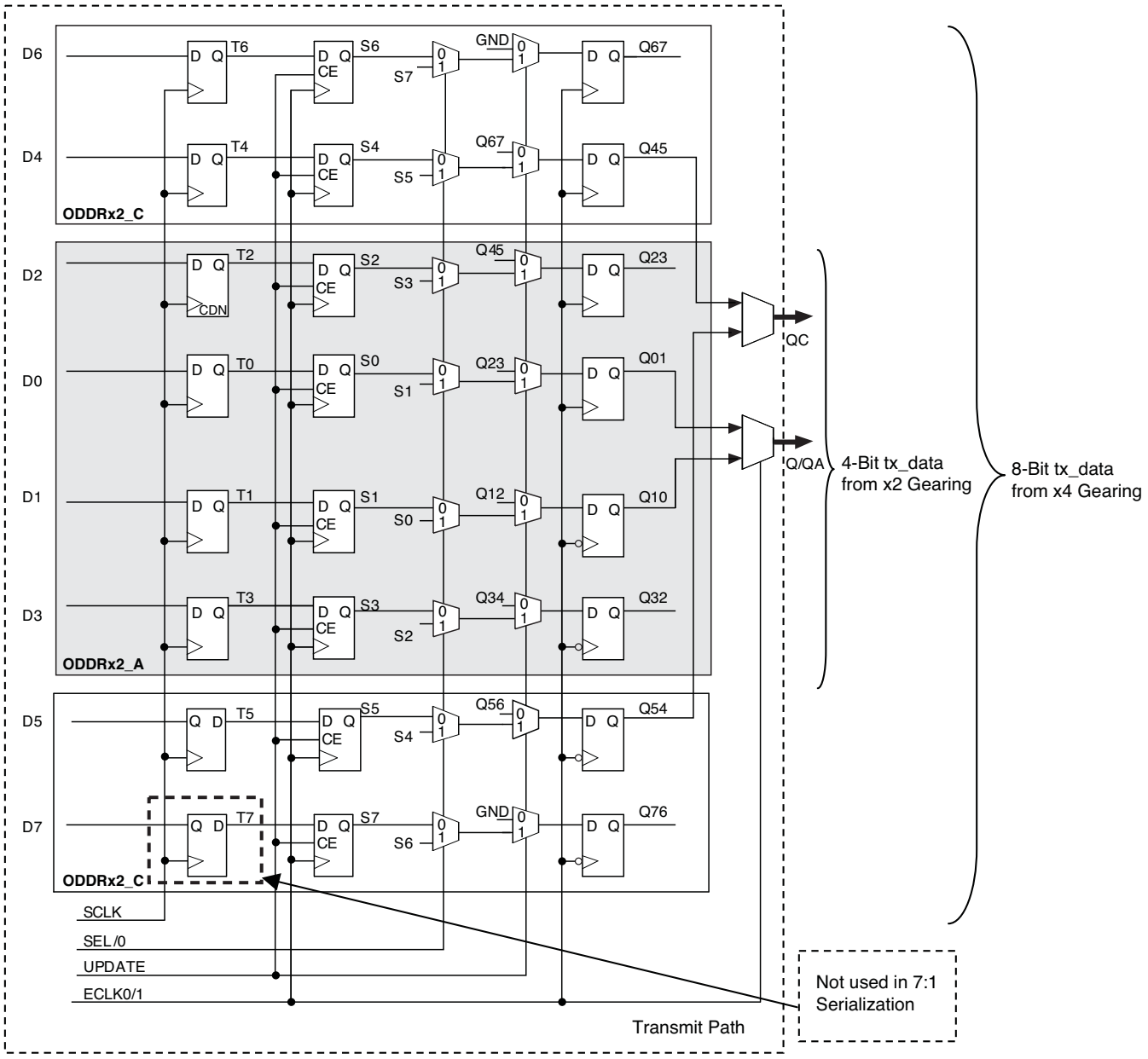


Figure 11-3. Video PIO Cell for x2/x4 and 7:1 Applications





Clock Domain Transfer at PIO Cells

The MachXO2 gearing logic performs serializing and de-serializing of high-speed data in the PIO cells. The clock domain transfer for the data from the high-speed edge clock (ECLK) to the low-speed system clock (SCLK) is guaranteed by design through two internal signals, UPDATE and SEL. The SEL signal toggles between '0' and '1' to sample three bits or four bits of data at a time for the 7:1 gearing. It remains static during the x2/x4 gearings. The UPDATE signal behaves the same for all the gearings to update the register with the correct byte of data. This data is then clocked by the SCLK for downstream processing. Figure 11-3 illustrates the architecture of x2 /x4 input gearing logic.

MachXO2 devices provide logic to support word alignment with minimal FPGA resources. The word alignment results in a shift to the UPDATE, SEL and the SCLK signals. It can be activated by providing an alignment request signal to the ALIGNWD port of the high-speed interface components. ALIGNWD can be asynchronous to the ECLK

domain, but it must be at least two ECLK cycles wide. For the 7:1 gearing, ALIGNWD must be pulsed seven times to loop through a maximum of seven combinations of word orders. For the x2/x4 gearings, ALIGNWD must be pulsed eight times to step through maximum eight possible word orders.

Figures 11-4 and 11-5 provide a timing relationship of UPDATE, SEL, ECLK, and SCLK signals under different gearing requirements. Figures 11-6 and 11-7 show the word alignment procedure for various gearing ratios. The discussion of gearing logic is applicable to both receive and transmit sides of the high-speed interfaces. Refer to reference design RD1093, MachXO2 Display Interface, for more details on implementing word alignment using the 7:1 gearing function.

The clock domain transfer for the DDR memory interface is accomplished by using the built-in, 90°-shifted clock trees for memory READ and WRITE operations. DLL and DQS detection logic are provided to guarantee the correct receive and transmit of data to and from DDR memories.

Figure 11-4. 7:1 Deserializer Timing

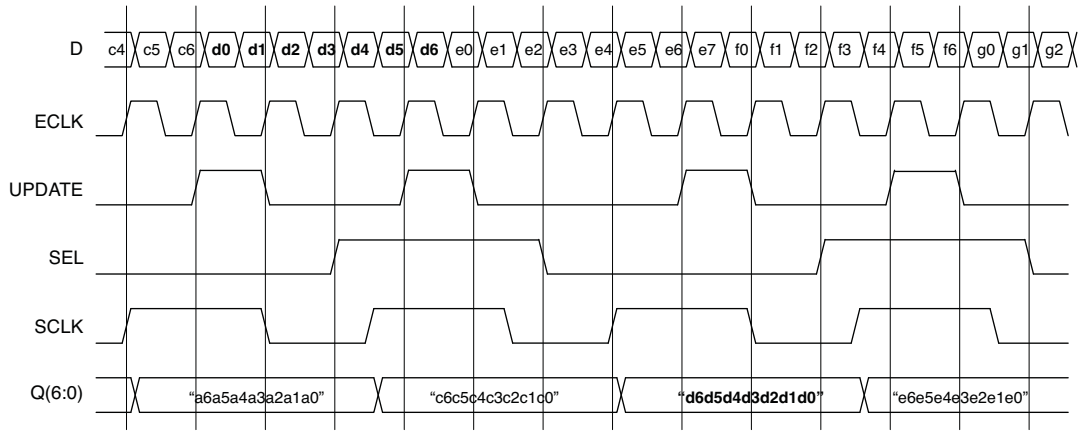


Figure 11-5. x2/x4 Deserializer Timing

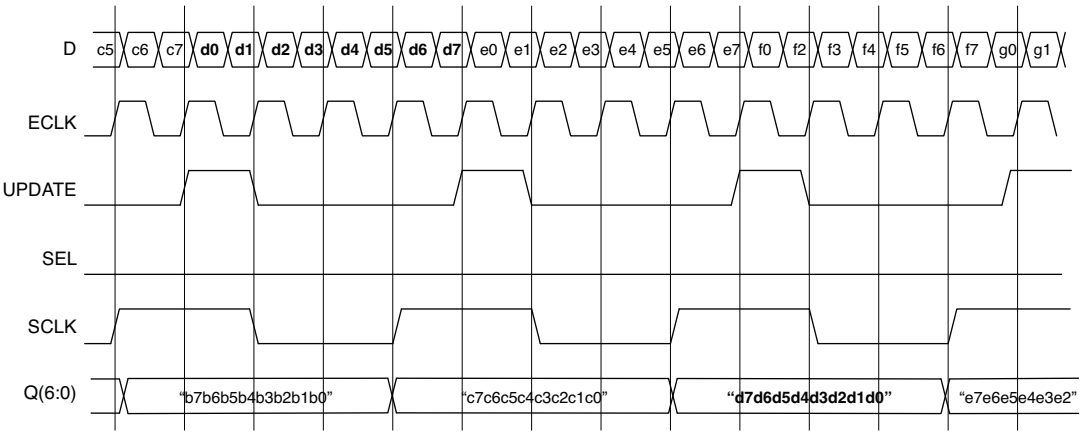


Figure 11-6. 7:1 Deserializer Timing in response to ALIGNWD

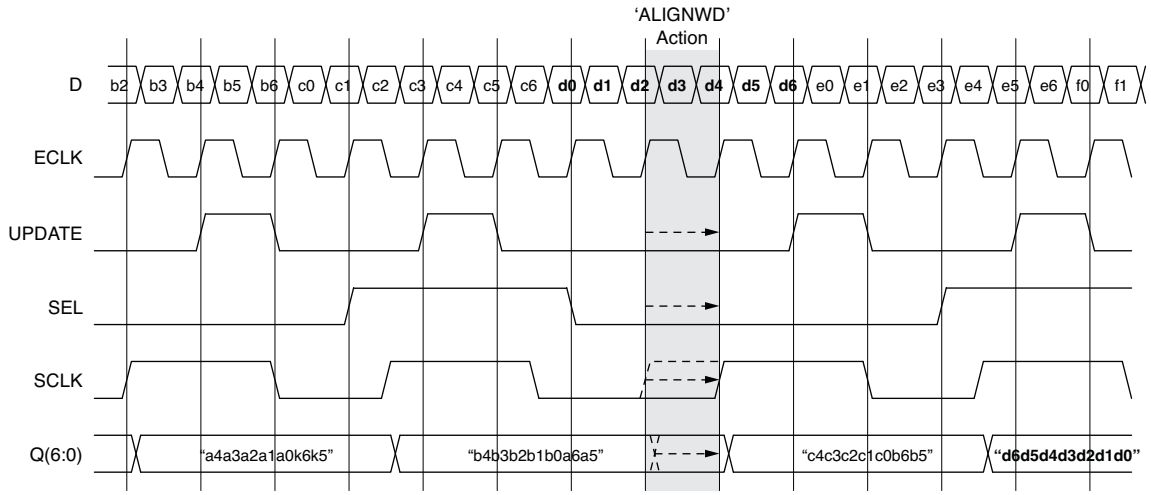
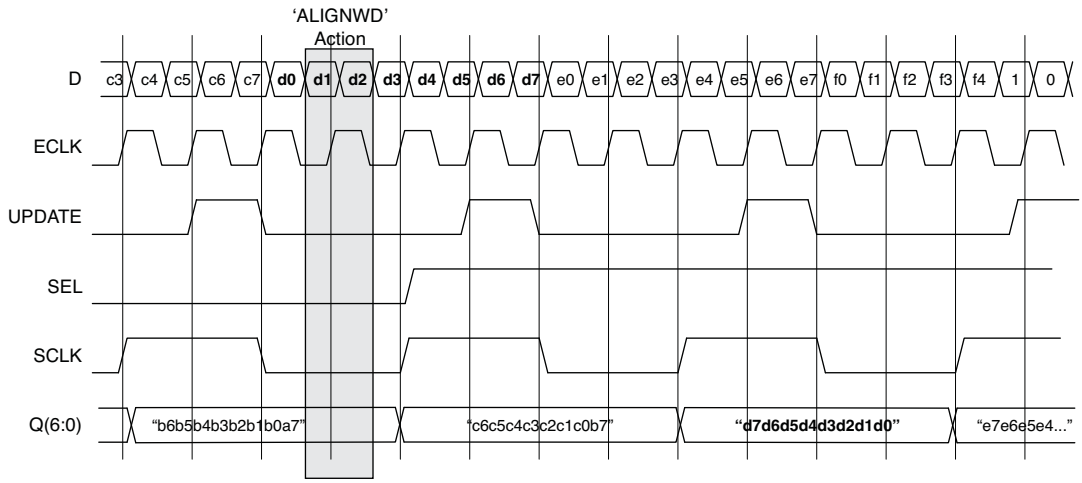


Figure 11-7. x2/x4 Deserializer Timing in response to ALIGNWD



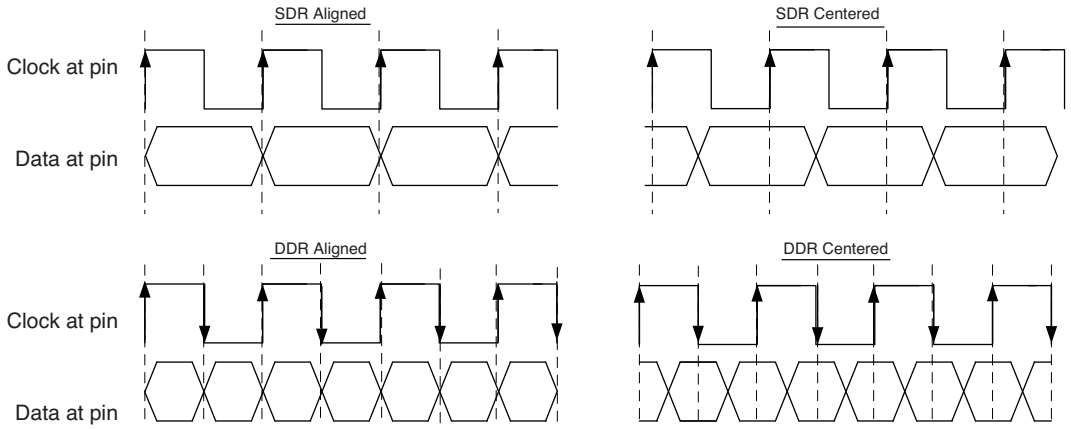
External High-Speed Interface Description

There are two types of external high-speed interface definitions that can be used with the MachXO2 devices: centered and aligned. In a centered external interface, at the device pins, the clock is centered in the data opening. In an aligned external interface, the clock and data transition are aligned at the device pins. This is sometimes called "edge-on-edge".

Figure 11-8 shows external interface waveforms for SDR and DDR. At the receive side, an aligned interface requires clock delay adjustment to position the clock edge at the middle of the data opening to ensure that the capture flip-flop setup and hold times are not violated. Similarly a centered interface at the transmit side will require a clock delay adjustment to position the clock at the center of the data opening for transmission.

Note that centered and aligned interfaces might both be used for a given bus. For example, in a DDR SDRAM memory the clock and data relationship during READ is an aligned interface, while the clock and data relationship during WRITE is a centered interface.

Figure 11-8. External Interface Definition



High-Speed Interface Building Blocks

MachXO2 devices provide dedicated logic blocks for building high-speed interfaces, with each block performing a unique function. Combining various blocks gives ultimate performance of a specific interface. The hardware components in the device are described in this section. The DDR Software Primitives and Attributes section describes the library elements for these components. Refer to TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide, for an in-depth discussion of clocking and PLL architectures.

ECLK

Edge clocks are high-speed, low-skew I/O dedicated clocks. Two edge clocks (ECLK) are available on each of the top and bottom sides for MachXO2-640U, MachXO2-1200/U and higher density devices. The primary clock nets (PCLK) have direct connectivity to ECLKs. The bottom PCLK pins also have minimal routing to PLLs for video applications.

ECLKSYNC

This is the ECLK synchronization block. Each ECLK has its own ECLKCYNC component to synchronize the clock domain transfer of data. This block can also be used to dynamically disable an edge clock to save power during operation.

SCLK

SCLK refers to the system clock of the design. SCLK must use primary clock pins or primary clock nets for high speed interfaces. There are eight primary clock pins (PCLK) available for the MachXO2 device, and eight primary clock nets in the MachXO2 devices.

CLKDIV

Clock dividers are used to generate low-speed system clocks from a high-speed edge clock. There are two clock dividers per top and bottom sides of the MachXO2-640U, MachXO2-1200/U and higher density devices. The ECLK frequency can be divided down by 2, by 3.5, or by 4 through the CLKDIV component.

PLL

A maximum of two PLLs are available in the MachXO2 devices. The number of PLLs varies with the logic density. The MachXO2-640U, MachXO2-1200/U and MachXO2-2000 have one PLL. MachXO2-2000U, MachXO2-4000, and MachXO2-7000 devices have two PLLs. There are pre-assigned dual-purpose I/O pins that drive to PLLs as reference clock inputs.

DQSDLL

A maximum of two DQSDLLs are available in the MachXO2-640U, MachXO2-1200/U and higher density devices. The top-right DQSDLL controls the top and right banks. The bottom-left DQSDLL can be used by the bottom and left banks. This component can be used for both DDR memory and generic high-speed interfaces. The DQSDLL, together with the clock slave delay cell (DLLDEL), is used to create a 90° clock shift/delay for aligned receiver interfaces.

Input DDR (IDDR)

Generic input DDR components support x1, x2, x4, and 7:1 gearing ratios at the receiving side of the PIO cells. The x1 gearing is supported by IDDRX, or the basic PIO cell. It receives 1-bit DDR data and outputs 2-bit wide parallel data synchronized to the SCLK. There is no clock domain transfer involved in the x1 gearing. The x2 gearing is supported by IDDRX2. It receives 1-bit DDR data synchronized to the ECLK and outputs four bits of parallel data synchronized to the SCLK. The same function applies to the IDDRX4, which receives a single bit of DDR data synchronized to the ECLK and outputs eight bits of parallel data synchronized to the SCLK. The 7:1 gearing shares the same structure as the x4 gearing. The 7:1 gearing outputs seven bits of parallel data instead of eight. The generic high-speed interface gearings are supported by the video PIO cells.

Output DDR (ODDR)

Generic output DDR components support x1, x2, x4, and 7:1 gearing ratios at the transmit side of the PIO cells. The x1 gearing is supported by ODDRX, or the basic PIO cell. It serializes the 2-bit data based on SCLK. There is no clock domain transfer involved in x1 gearing. The x2 gearing is supported by ODDRX2. The 4-bit parallel data is clocked by SCLK and is serialized using ECLK. The x4 gearing is supported by ODDRX4. The 8-bit parallel data is clocked by SCLK and is serialized using ECLK. The 7:1 gearing shares the same structure as the x4 gearing. The 7-bit parallel data is serialized by the ECLK. The generic high speed interface gearings are supported by the video PIO cells.

Delays

There are two types of delay available for high-speed interfaces. The first type is the I/O logic delay that can be applied on the input data paths, as shown in the block diagram of PIO architectures at the beginning of the document. Although the 32-tap I/O logic delay can be static or dynamic, only the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices supports dynamic data path delay. The static I/O logic delay (DELAYE) is used by default when configuring the interface in the Lattice design software. Software applies fixed delay values based on the interface used. Dynamic delay (DELAYD) at the bottom-side input data path provides dynamic or user-defined delay. Dynamic delay requires extra ports available on the module to be connected to user logic for delay control. The I/O logic delay is used to achieve the SDR zero hold timing, or match primary clock injection for x1 gearing, or match the edge clock injection for x2/x4 gearings.

The second type of delay is the clock slave delay cell (DLLDEL), which delays the incoming clock by 90° to place the clock in the middle of the data opening. This block is digitally controlled by the DQSDLL through 7-bit control code. There is one clock slave delay cell per primary clock pin. Its input comes from the primary clock pins and its output can drive the primary clock net for the x1 aligned interface or ECLK for x2/x4 aligned interfaces.

DQSBUF

This is the dedicated DQS circuitry for DDR memory. It generates a 90° shift on DQS for memory read operations, and a 90° shift on the SCLK for memory write operations. The clock polarity detection, burst detection, and data valid signals are generated from this block. This is available on the right bank of the MachXO2-640U, MachXO2-1200/U and higher density devices.

IDDRDQS

DDR memory input buffer supports clock domain transfers from DQS to SCLK. These memory PIO cells are used exclusively for DDR memory interfaces and are used for DDR memory READ operations. They are available on the right bank of the MachXO2-640U, MachXO2-1200/U and higher density devices.

ODDRDQS

The DDR memory output buffer supports clock domain transfers from SCLK to DQS. These memory PIO cells are used exclusively for DDR memory interfaces and are used for DDR memory WRITE operations. They are available on the right bank of the MachXO2-1200 and higher density devices.

Generic High-Speed DDR Interfaces

Generic high-speed interfaces, or Generic DDR (GDDR), are supported in MachXO2 using the dedicated logic blocks. This section will discuss the GDDR types, the interface logic, and the software to support the GDDR capability in the silicon.

High-Speed GDDR Interface Types

The GDDR interfaces supported by the MachXO2 device family are pre-defined in the software and characterized in the silicon. Table 11-2 lists all the supported interfaces and gives a brief description of each interface.

Table 11-2. Generic High-Speed I/O DDR Interfaces[1]

Mode	Interface Name	Description	Supporting Device & Sides
RX SDR	GIREG_RX.SCLK	SDR input using SCLK	All devices, all sides
RX GDDRx1 Aligned	GDDRX1_RX.SCLK.Aligned	DDRx1 input using SCLK, data is edge-to-edge with incoming clock	640U, 1200/U, and above, all sides
RX GDDRx1 Centered	GDDRX1_RX.SCLK.Centered	DDRx1 input using SCLK, incoming clock is centered at the data opening	All devices, all sides
RX GDDRx2 Aligned	GDDRX2_RX.ECLK.Aligned	DDRx2 input using ECLK, data is edge-to-edge with incoming clock	640U, 1200/U, and above, bottom
RX GDDRx2 Centered	GDDRX2_RX.ECLK.Centered	DDRx2 input using ECLK, incoming clock is centered at the data opening	640U, 1200/U, and above, bottom
RX GDDRx4 Aligned	GDDRX4_RX.ECLK.Aligned	DDRx4 input using ECLK, data is edge-to-edge with incoming clock	640U, 1200/U, and above, bottom
RX GDDRx4 Centered	GDDRX4_RX.ECLK.Centered	DDRx4 input using ECLK, incoming clock is centered at the data opening	640U, 1200/U, and above, bottom
RX GDDR71	GDDR71_RX.ECLK.7:1	GDDR 7:1 input using ECLK	640U, 1200/U, and above, bottom
TX SDR	GOREG_TX.SCLK	SDR output using SCLK	All devices, all sides
TX GDDRx1 Aligned	GDDRX1_TX.SCLK.Aligned	DDRx1 output using SCLK, data is edge-to-edge with outgoing clock	All devices, all sides
TX GDDRx1 Centered	GDDRX1_TX.SCLK.Centered	DDRx1 output using SCLK, outgoing clock is centered at the data opening	640U, 1200/U, and above, all sides
TX GDDRx2 Aligned	GDDRX2_TX.ECLK.Aligned	DDRx2 output using ECLK, data is edge-to-edge with outgoing clock	640U, 1200/U, and above, top
TX GDDRx2 Centered	GDDRX2_TX.ECLK.Centered	DDRx2 output using ECLK, outgoing clock is centered at the data opening	640U, 1200/U, and above, top
TX GDDRx4 Aligned	GDDRX4_TX.ECLK.Aligned	DDRx4 output using ECLK, data is edge-to-edge with outgoing clock	640U, 1200/U, and above, top
TX GDDRx4 Centered	GDDRX4_TX.ECLK.Centered	DDRx4 output using ECLK, outgoing clock is centered at the data opening	640U, 1200/U, and above, top
TX GDDR71	GDDR71_TX.ECLK.7:1	GDDR 7:1 output using ECLK	640U, 1200/U, and above, top

1. For the “R1” version of the MachXO2 devices GDDRX2, GDDRX4 and GDDR71 modes, ECLKSYNC may have a glitch in the output under certain conditions, leading to possible loss of synchronization. The “R1” versions of the MachXO2 devices have an “R1” suffix at the end of the part number (e.g., LCMXO2-1200ZE-1TG144CR1). For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

The following describes the naming conventions used for each of the interfaces listed in Table 11-2.

- G – Generic
- IREG – SDR input I/O register
- OREG – SDR output I/O register
- DDRX1 – DDR x1 I/O register
- DDRX2 – DDR x2 I/O register
- DDRX4 – DDR x4 I/O register
- DDR71 – DDR 7:1 I/I register
- _RX – Receive interface
- _TX – Transmit interface
- ECLK – Uses ECLK (edge clock) clocking resource at the GDDR interface
- SCLK – Uses SCLK (primary clock) clocking resource at the GDDR interface
- Centered – Clock is centered to the data when coming into the device
- Aligned – Clock is aligned edge-on-edge to the data when coming into the device

High-Speed GDDR Interface Details

This section describes each of the generic high-speed interfaces in detail including the clocking to be used for each interface. For detailed information about the MachXO2 clocking structure, refer to TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide. As listed in Table 11-2, each interface is supported in specific bank locations of the MachXO2 devices. It is important to follow the architecture and various interface rules and preferences listed under each interface in order to build these interfaces successfully. The discussion of each component can be found in the DDR Software Primitives and Attributes section of this document.

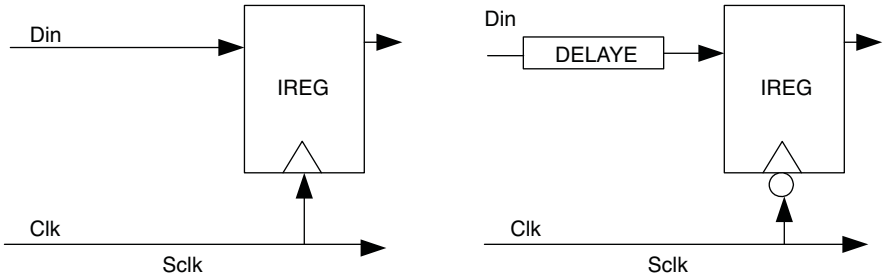
Receive Interfaces

There are eight receive interfaces pre-defined and supported through Lattice IPexpress™ software.

GIREG_RX.SCLK

This is a generic interface for single data rate (SDR) data. The standard I/O register in the basic PIO cell (Figure 11-1) is used for the implementation. An optional inverter can be used to center the clock for aligned inputs. PLLs or DLLs can be used to remove the clock injection delay or adjust the setup and hold times. There are a limited number of DLLs in the architecture and these should be saved for high-speed interfaces when necessary. This interface can either be built using IPexpress, instantiating an I/O register element, or inferred during synthesis.

Figure 11-9. GIREG_RX Interface



The input data path delay cells can be used on the Din path of the interface. A DELAYE element provides a fixed delay to match the SCLK injection time. The dynamic input delay, DELAYD, is not available for this interface.

Figure 11-9 shows possible implementations of this interface.

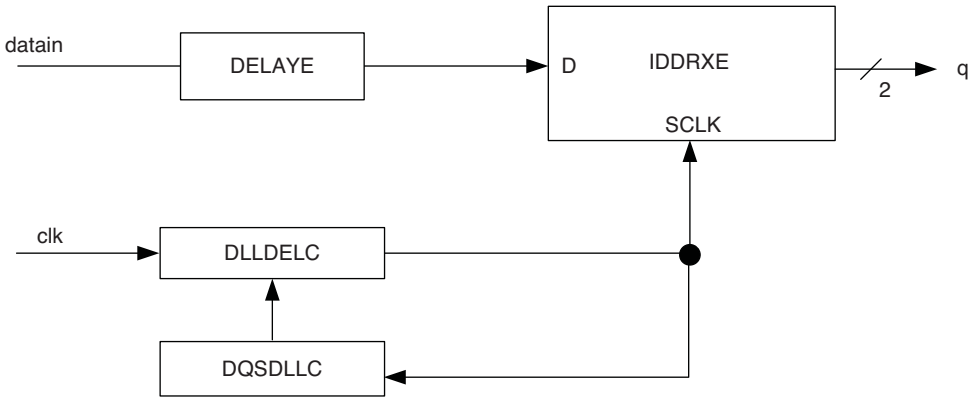
Interface rules:

- Must use a dedicated clock pin PCLK as the clock source

GDDRX1_RX.SCLK.Aligned

This DDR interface uses the SCLK and the DQSDLL to provide a 90° clock shift to center the clock at the IDDRXE. A DELAYE element is used to adjust data delay for the SCLK clock injection time. The DELAYD is not available for the x1 interface.

Figure 11-10. GDDRX1_RX.SCLK.Aligned Interface Using DQSDLL



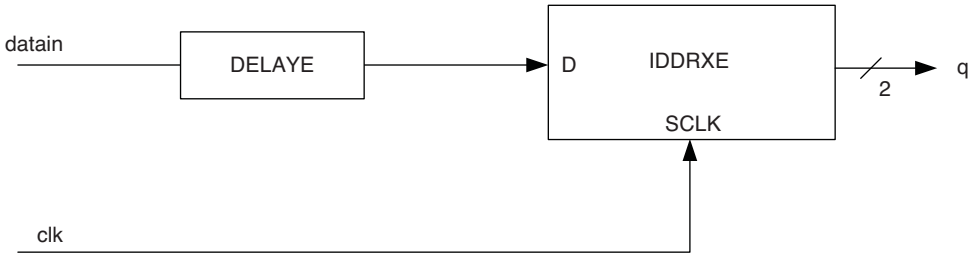
Interface rules:

- Must use a dedicated clock pin PCLK as the clock source for DLLDELC
- A primary clock net must be used to connect DLL outputs to the SCLK port
- The DELAYE value should be set to SCLK_ALIGNED for the best timing
- There are up to two DQSDLLCs per device. This limits the interface to a maximum of two clock frequencies per device.

GDDRX1_RX.SCLK.Centered

This DDR interface uses DELAYE to match the SCLK delay at the IDDRXE. DELAYD is not available for the x1 interface. Since it is a centered interface, the clock edge is already in the middle of the data opening. There is no logic required to shift the clock.

Figure 11-11. GDDRX1_RX.SCLK.Centered



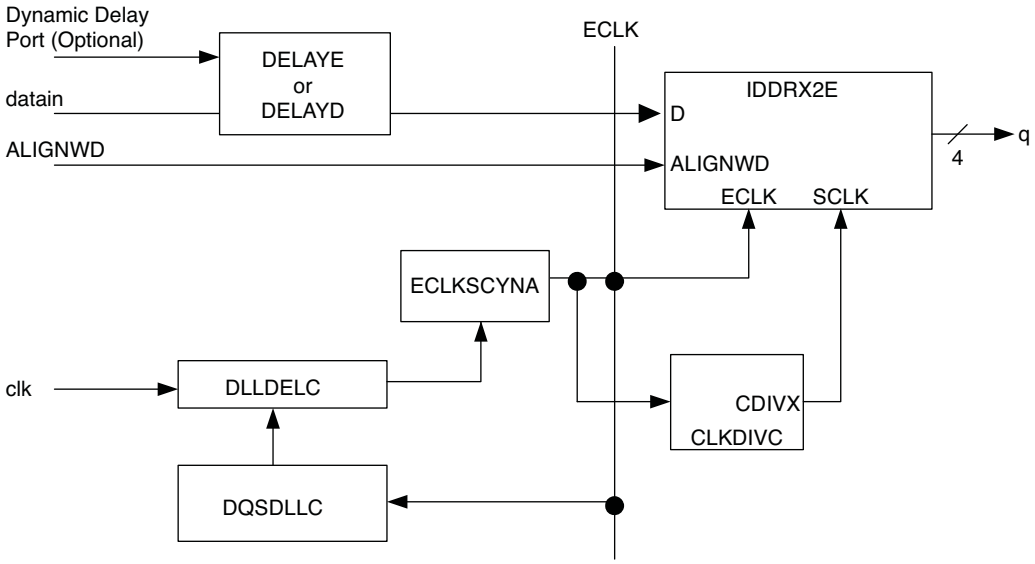
Interface rules:

- Must use a dedicated clock pin PCLK as the clock source
- DELAYE value should be set to SCLK_CENTERED for the best timing
- The clock connected to SCLK should be on a primary clock net

GDDRX2_RX.ECLK.Aligned

This DDR x2 interface uses the DQSDLL to provide a 90° clock shift to center the clock at the IDDRX2E buffer. DELAYE is used to delay data to match the ECLK injection delay. DELAYD can also be used to control the delay dynamically. This interface uses x2 gearing with the IDDRX2E element. This requires the use of a CLKDIVC to provide the SCLK which is half the frequency of the ECLK. The ECLKSYNCA element is associated with the ECLK and must be used to drive the ECLK. The port ALIGNWD can be used for word alignment at the interface.

Figure 11-12. GDDRX2_RX.ECLK.Aligned Interface



Interface rules:

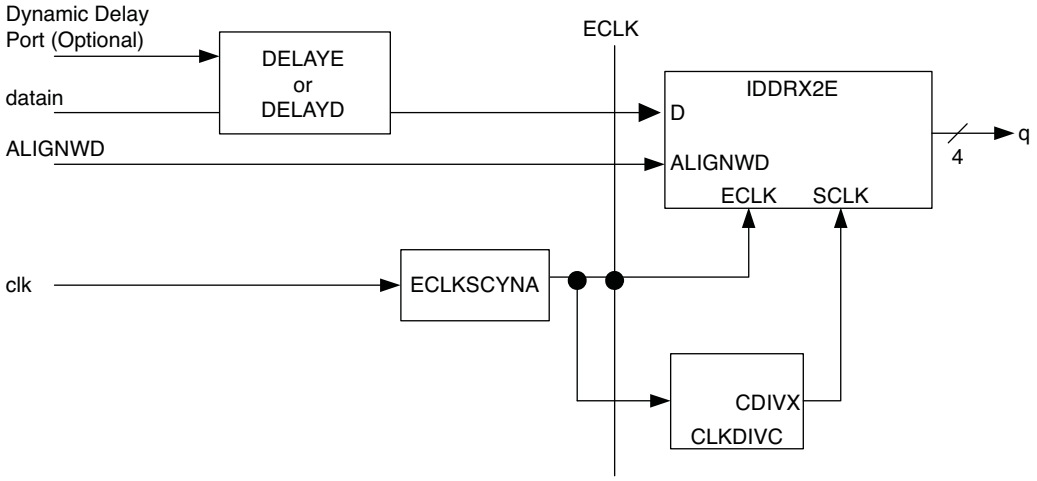
- Must use a dedicated clock pin PCLK as the clock source for DLLDELC
- Clock net routed to SCLK must use primary clock net

- There are up to two DQSDLLC per device. It limits this interface to a maximum of two clock frequencies per device.
- DELAYE should be set to ECLK_ALIGNED
- When DELAYD is used, only one dynamic delay port is needed for the entire bus
- This interface is supported at the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDRX2_RX.ECLK.Centered

This DDR x2 interface uses DELAYE or DELAYD to match edge clock delay at the IDDRX2E. Since this interface uses the ECLK it can be extended to support large data bus sizes for the entire side of the device. This interface uses x2 gearing with the IDDRX2D element. This requires the use of a CLKDIVC to provide the SCLK which is half the frequency of the ECLK. The port ALIGNWD can be used for word alignment at the interface.

Figure 11-13. GDDRX2_RX.ECLK.Centered Interface



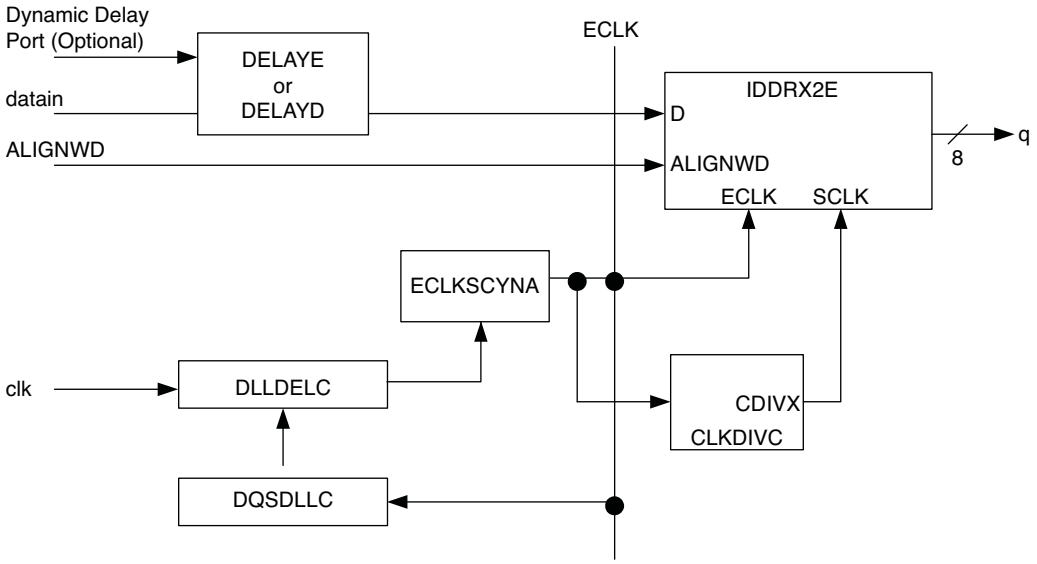
Interface rules:

- Must use a dedicated clock pin PCLK as the clock source for ECLKSYNCA
- Clock net routed to SCLK must use primary clock net
- DELAYE should be set to ECLK_CENTERED
- When DELAYD is used, only one dynamic delay port is needed for the entire bus
- This interface is supported at the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDRX4_RX.ECLK.Aligned

This DDR x4 interface uses the DQSDLL to provide a 90° clock shift to center the edge clock at the IDDRX4B buffer. DELAYE is used to delay data to match the ECLK injection delay. DELAYD can also be used to control the delay dynamically. Since this interface uses the ECLK, it can be extended to support large data bus sizes for the entire side of the device. This interface uses x4 gearing with the IDDRX4B element. This requires the use of a CLKDIVC to provide the SCLK which is one quarter of the ECLK frequency. ECLKSYNCA element is associated with the ECLK and must be used to drive the ECLK. The port ALIGNWD can be used for word alignment at the interface.

Figure 11-14. GDDRX4_RX.ECLK.Aligned Interface



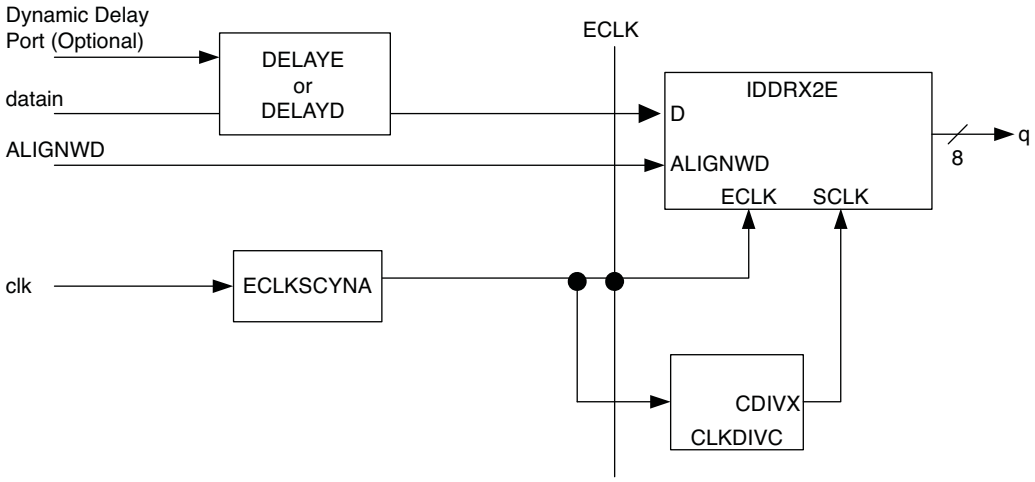
Interface rules:

- Must use a dedicated clock pin PCLK as the clock source for DLLDELC
- Clock net routed to SCLK must use primary clock net
- There are up to two DQSDLLC per device. It limits this interface to have maximum of two clock frequencies per device.
- Data input must use A/B pair of the I/O logic cells for x4 gearing
- DELAYE should be set to ECLK_ALIGNED
- When DELAYD is used, only one dynamic delay port is needed for the entire bus
- This interface is supported at the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDRX4_RX.ECLK.Centered

This DDR x4 interface uses DELAYE or DELAYD to match edge clock delay at the IDDRX4B. Since this interface uses the ECLK it can be extended to support large data bus sizes for the entire side of the device. This interface uses x4 gearing with the IDDRX4B element. This requires the use of a CLKDIVC to provide the SCLK which is one quarter of the ECLK frequency. The port ALIGNWD can be used for word alignment at the interface.

Figure 11-15. GDDRX4_RX.ECLK.Centered Interface



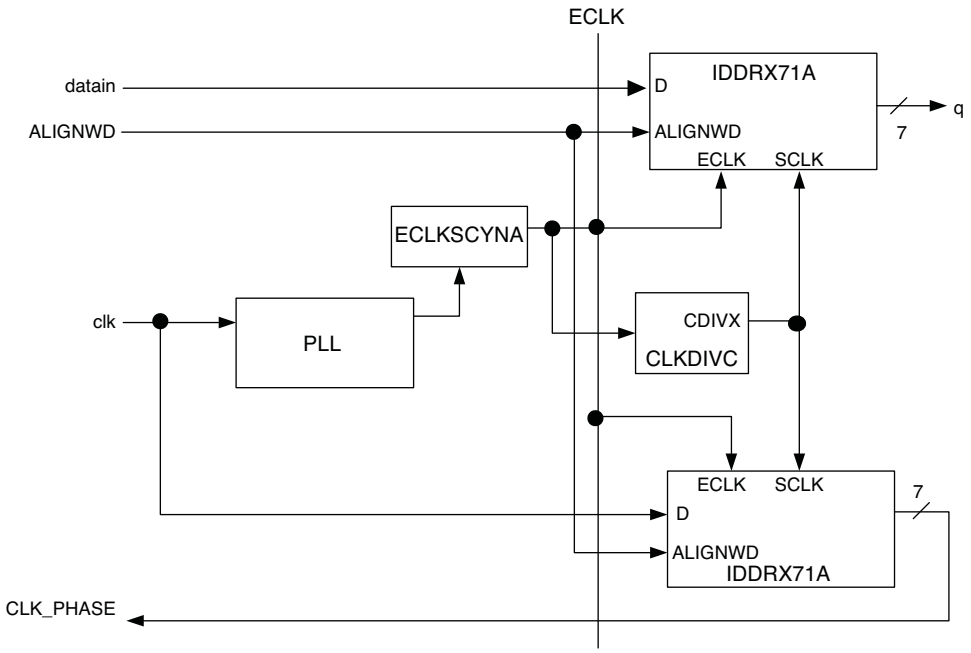
Interface rules:

- Must use a dedicated clock pin PCLK as the clock source for ECLKSYNCA
- Clock net routed to SCLK must use primary clock net
- Data input must use A/B pair of the I/O logic for x4 gearing
- DELAYE should be set to ECLK_CENTERED
- When DELAYD is used, one dynamic delay port is needed for the entire bus
- This interface is supported at the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDR71_RX.ECLK.7:1

The GDDR 7:1 receive interface is unique among the supported high-speed DDR interfaces. It uses the PLL to search the best clock edge to the data opening position during bit alignment process. The PLL steps through the 16 phases to give eight sampling points per data. The data path delay is not used in this interface. CLKDIVC is used to divide down the ECLK by 3.5 due to the nature of the 1:7 deserializing requirement. This means the SCLK is running seven times slower than the incoming data rate. ECLKSYNCA element is associated with the ECLK and must be used to drive the ECLK. The complete 7:1 LVDS video display application requires bit alignment and word alignment blocks to be built in the FPGA resources in addition to the built-in I/O gearing logic and alignment logic. The CLK_PHASE signal is sent to the FPGA side to build the bit alignment logic.

Figure 11-16. GDDR71_RX.ECLK.7:1 Interface



Interface rules:

- Must use a dedicated clock pin PCLK at the bottom side as the clock source for PLL
- Clock net routed to SCLK must use primary clock net
- There are up to two PLLs per device. It limits this interface to have maximum of two clock frequencies per device.
- The data input must use A/B pair of the I/O logic cell for 7:1 gearing
- This interface is supported at the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices

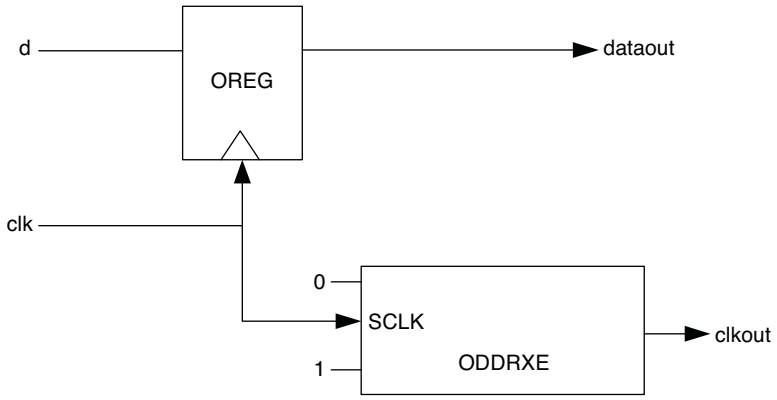
Transmit Interfaces

There are eight transmit interfaces pre-defined and supported through Lattice IPexpress software.

GOREG_TX.SCLK

This is a generic interface for SDR data and a forwarded clock. The standard register in the basic PIO cell is used to implement this interface. The ODDRXE used for the output clock balances the clock path to match the data path. A PLL can also be used to clock the ODDRXE to phase shift the clock to provide a precise clock to data output. There are a limited number of PLLs in the architecture and these should be saved for high-speed interfaces when necessary. This interface can either be built using IPexpress, instantiating an I/O register element, or inferred during synthesis.

Figure 11-17. GOREG_TX.SCLK Interface



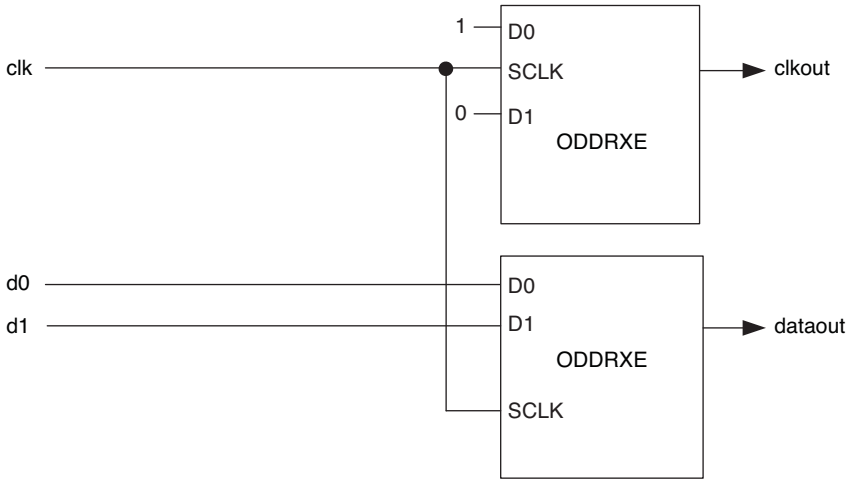
Interface rules:

•The clock source for SCLK must be routed on a primary clock net

GDDRX1_TX.SCLK.Aligned

This output DDR interface provides clock and data that are aligned using a single SCLK. The ODDRXE used for the output clock balances the clock path to match the data path.

Figure 11-18. GDDRX1_TX.SCLK.Aligned Interface



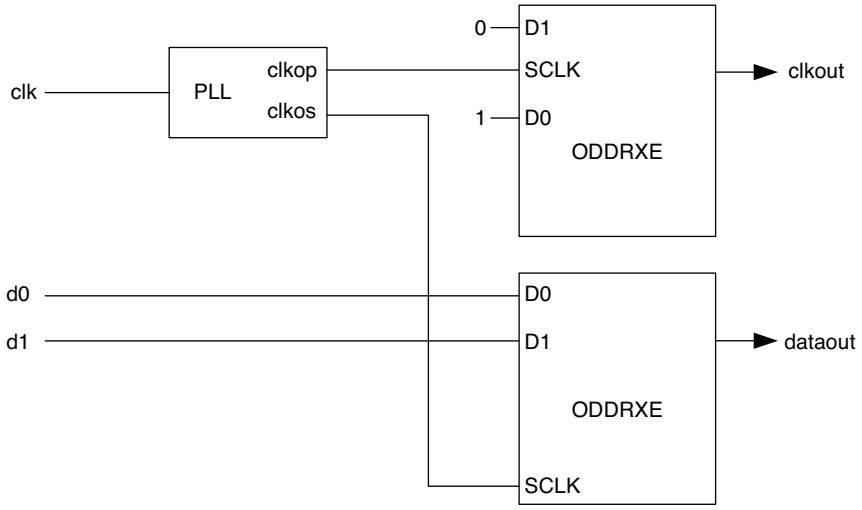
Interface rules:

- The clock source for SCLK must be routed on a primary clock net

GDDRX1_TX.SCLK.Centered

This output DDR interface provides clock and data that are pre-centered. PLL uses clkop and clkos ports to provide the 90° phase difference between the data and the clock. It requires two SCLK resources to drive the output data I/O cell and the output clock I/O cell.

Figure 11-19. GDDRX1_TX.SCLK.Centered Interface



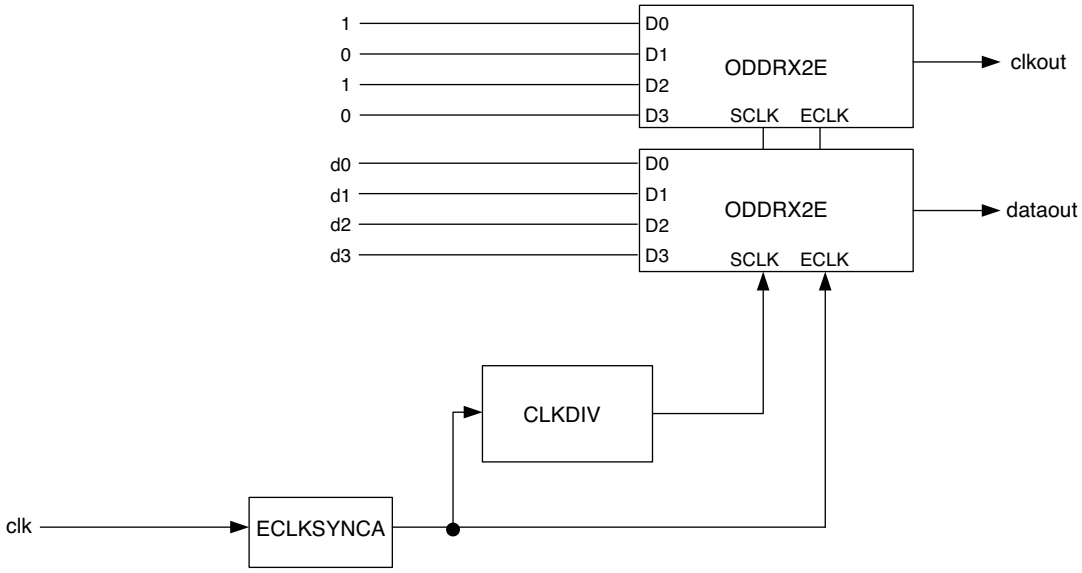
Interface rules:

- SCLK and 90°-shifted SCLK must be routed on primary clock nets

GDDRX2_TX.ECLK.Aligned

This output DDR x2 interface provides clock and data that are aligned. A CLKDIV is used to generate the SCLK which is half of the ECLK frequency. The ECLKSYNC element is used on the ECLK path for data synchronization.

Figure 11-20. GDDRX2_TX.ECLK.Aligned Interface



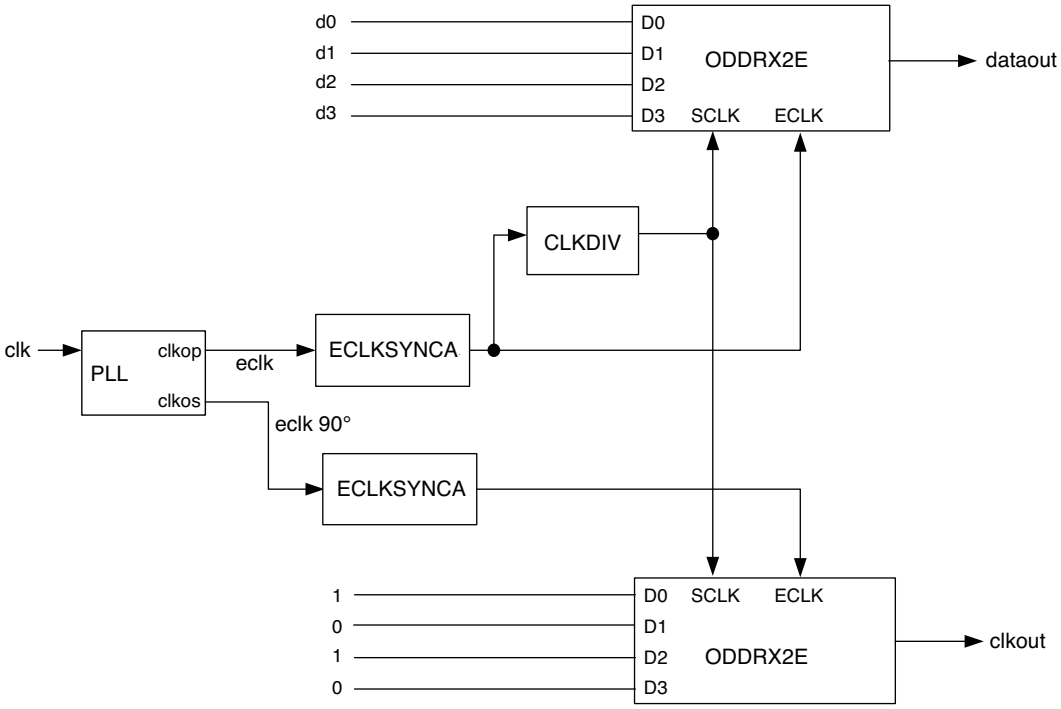
Interface rules:

- Must use edge clock routing resources for the ECLK
- The routing of SCLK must use primary clock net
- This interface is supported at the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDRX2_TX.ECLK.Centered

This output DDR x2 interface provides a clock that is centered at the data opening. The PLL uses clkop and clkos ports to provide the 90° phase difference between the data and the clock. Two ECLK routing resources are used in this interface to drive the output data and the output clock. A CLKDIV is used to generate the SCLK which is half of the ECLK frequency.

Figure 11-21. GDDRX2_TX.ECLK.Centered Interface



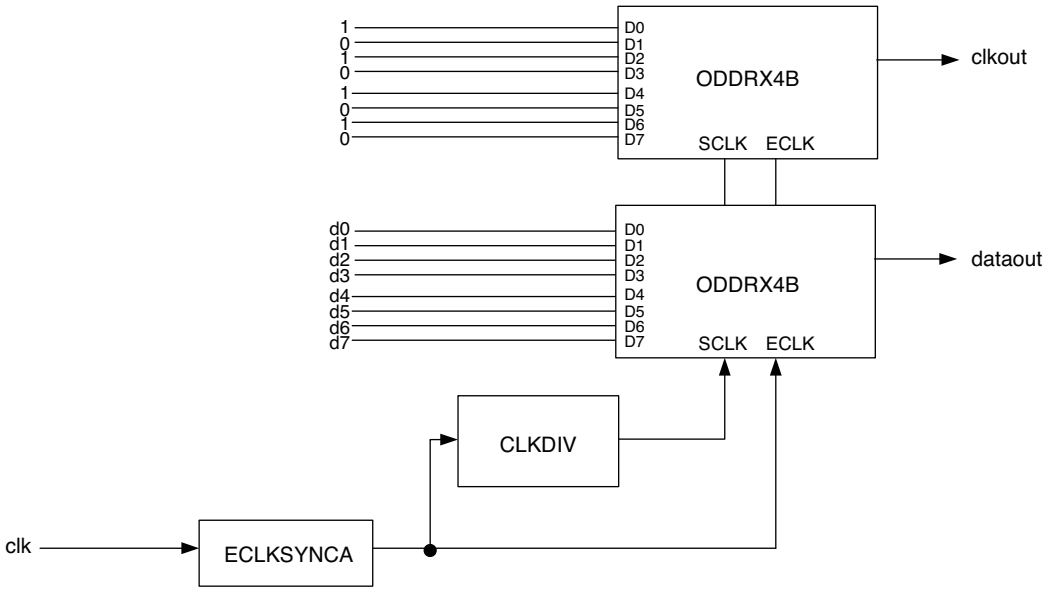
Interface rules:

- Must use edge clock routing resources for the ECLK
- Since two ECLKs are used on this interface, maximum one bus of this interface can be implemented at a time.
- The routing of the SCLK must use primary clock net
- This interface is supported at the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDRX4_TX.ECLK.Aligned

This output DDR x4 interface provides clock and data that are aligned. A CLKDIV is used to generate the SCLK which is a quarter of the ECLK frequency. The ECLKSYNC element is used on the ECLK path for data synchronization.

Figure 11-22. GDDRX4_TX.ECLK.Aligned Interface



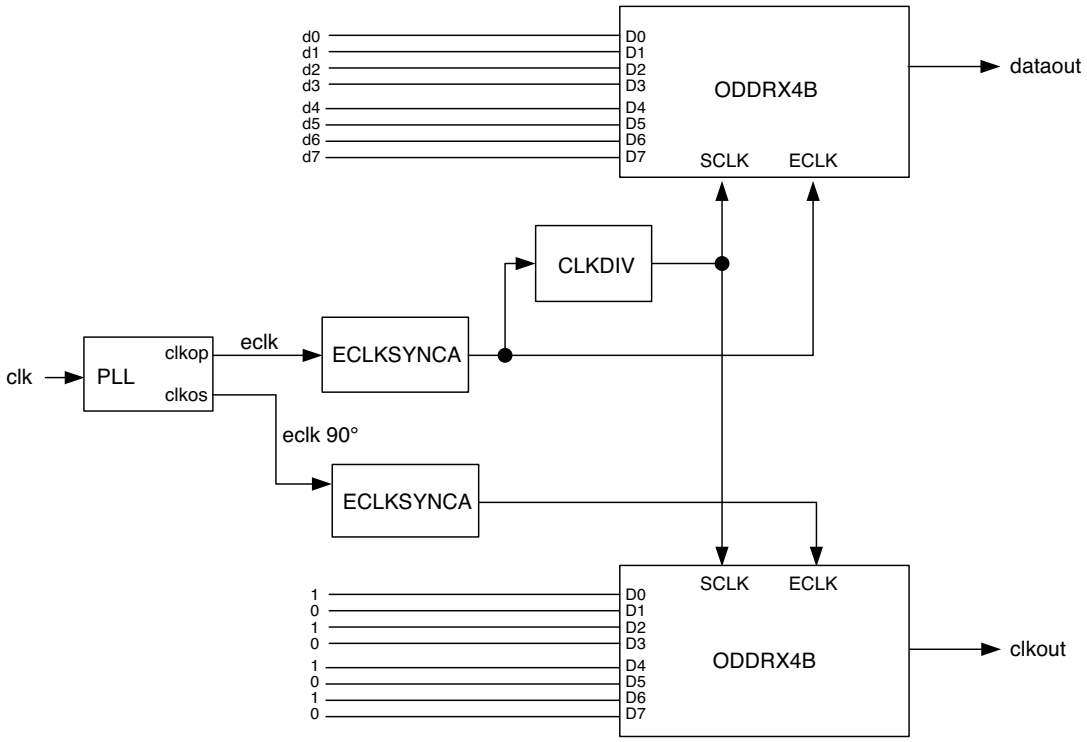
Interface rules:

- Must use edge clock routing resources for the ECLK
- The routing of SCLK must use primary clock net
- Data output must use A/B pair of the I/O logic for x4 gearing
- This interface is supported at the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDRX4_TX.ECLK.Centered

This output DDR x4 interface provides a clock that is centered at the data opening. The PLL uses clkop and clkos ports to provide the 90° phase difference between the data and the clock. Two ECLK routing resources are used in this interface to drive the output data and the output clock. A CLKDIV is used to generate the SCLK which is one-quarter of the ECLK frequency.

Figure 11-23. GDDRX4_TX.ECLK.Centered Interface



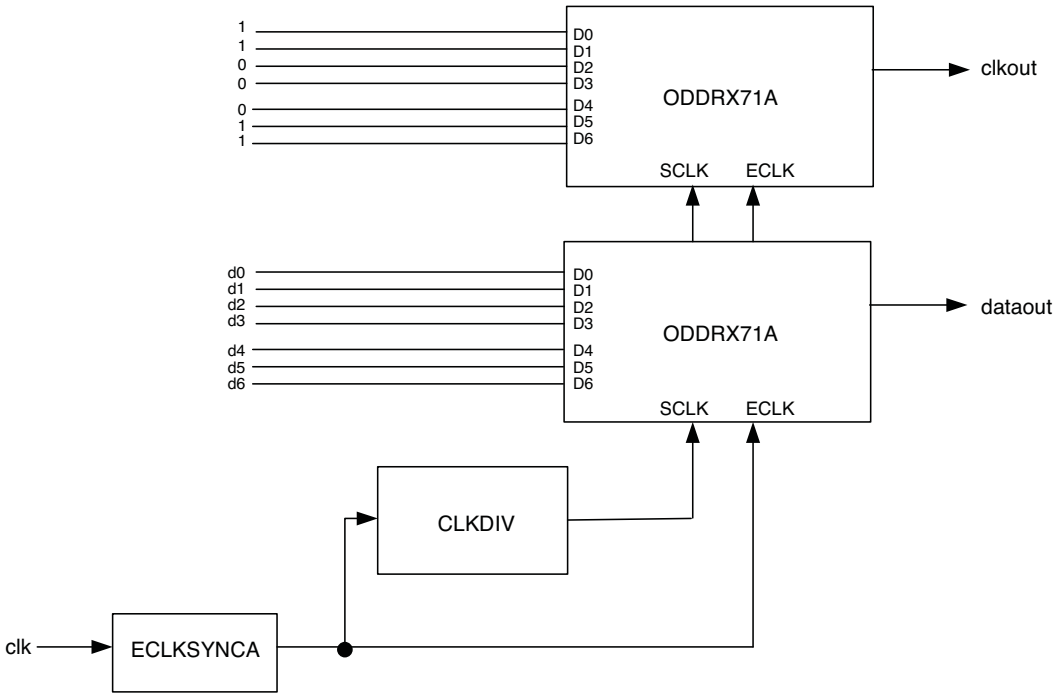
Interface rules:

- Must use edge clock routing resources for the ECLK
- Since two ECLKs are used on this interface, a maximum of one bus of this interface can be implemented at a time
- The routing of the SCLK must use primary clock net
- Data output must use A/B pair of the I/O logic for x4 gearing
- This interface is supported at the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices

GDDR71_TX.ECLK.7:1

The GDDR 7:1 transmit interface is unique among the supported high-speed DDR interfaces. It uses a specific pattern to generate the output clock, known as pixel clock. The CLKDIVC is used to divide down the ECLK by 3.5 due to the nature of the 7:1 serializing requirement. This means the SCLK is running 7 times slower than the transmit data rate. ECLKSYNCA element is associated with the ECLK and must be used to drive the ECLK.

Figure 11-24. GDDR71_TX.ECLK.7:1 Interface



Interface rules:

- Must use edge clock routing resources for the ECLK
- The routing of SCLK must use primary clock net
- Data output must use A/B pair of the I/O logic for 7:1 gearing
- This interface is supported at the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices

Using IPexpress to build Generic High-Speed DDR Interfaces

The IPexpress tool of the Lattice development software should be used to configure and generate all the generic high-speed interfaces described above. IPexpress will generate a complete HDL module including clocking requirements for each of the interfaces described above. In the IPexpress GUI, all the DDR modules are located under **Architecture Modules > IO**. This section covers the SDR, DDR_GENERIC, and GDDR_71 interfaces in IPexpress.

Table 11-3 shows the signal names used in the IPexpress modules. Each signal can be used in all or some specified interfaces. The signals are listed separately for the GDDR receive interfaces and the transmit interfaces.

Table 11-3. Signal Names used by IPexpress Modules

Signal Name	Direction	Description	Supported Interfaces
Receive Interface			
clk	Input	Source synchronous clock	All
reset	Input	Asynchronous reset to the interface, active high	All
datain	Input	Serial data input at Rx interfaces	All
uddcntln	Input	Hold/update control of delay code, active low	x1, x2, x4 Aligned
freeze	Input	Freeze or release DLL, active high	x1, x2, x4 Aligned
alignwd	Input	Word alignment control signal, active high	x2, x4, 7:1
dqsdll_reset	Input	Asynchronous DQSDLL reset, active high	x2, x4 Aligned
clk_s[1]	Input	Slow clock for reset synchronization	x2, x4, 7:1
init	Input	Initialize reset synchronization, active high	x2, x4, 7:1
phase_dir	Input	PLL phase direction	7:1
phase_step	Input	PLL phase step	7:1
sclk	Output	System clock for the FPGA fabric	All
q	Output	Parallel data output of the Rx interfaces	All
lock	Output	DLL or PLL lock	x2, x4, 7:1
eclk	Output	Edge clock generated from the input clock	x2, x4 Aligned, 7:1
rx_ready	Output	Indicate completion of reset synchronization	x2, x4, 7:1
clk_phase	Output	7-bit representation of input clock phase	7:1
Transmit Interface			
clk	Input	Main input clock for Tx interfaces	All
reset	Input	Asynchronous reset to the interface, active high	All
dataout	Input	Parallel input data of the Tx interfaces	All
clk_s[1]	Input	Slow clock for reset synchronization	x2, x4, 7:1
sclk	output	System clock for the FPGA fabric	All
dout	Output	Serial data output for the Tx interfaces	All
clkout	output	Source synchronous clock	All
tx_ready	Output	Indicate completion of reset synchronization	x2, x4, 7:1

1. clk_s can be any slow clock to be used for reset synchronization process. This clock must be slower than eclk.

Building the SDR Interface

As shown in Figure 11-25, users can choose interface type SDR, enter the module name and click **Customize** to open the configuration tab.

Figure 11-25. SDR Interface Selection at the IPexpress Main Window

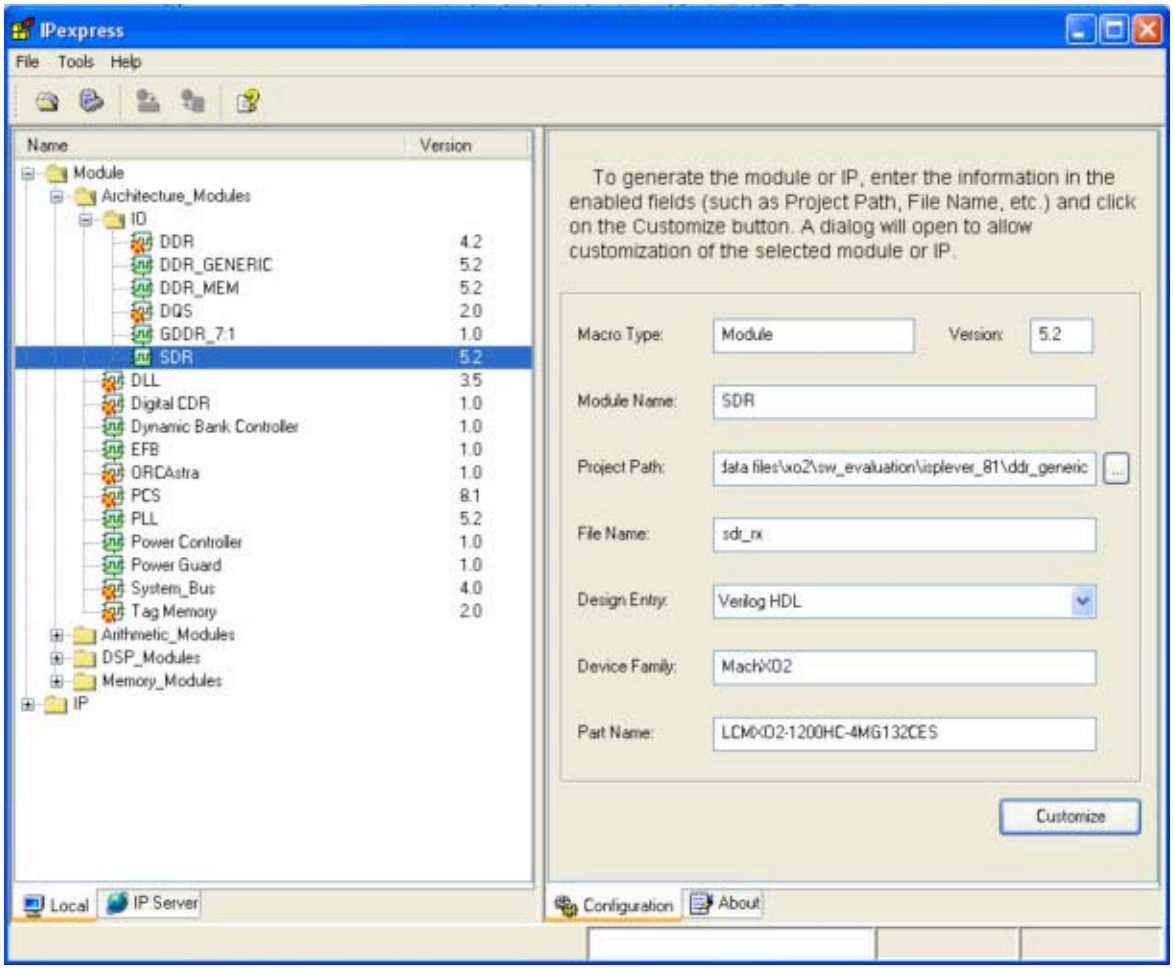


Figure 11-26 shows the Configuration Tab for the SDR module in IPexpress. Table 11-4 lists the various configurations options available for SDR modules.

Figure 11-26. Configuration Tab for the SDR Interfaces

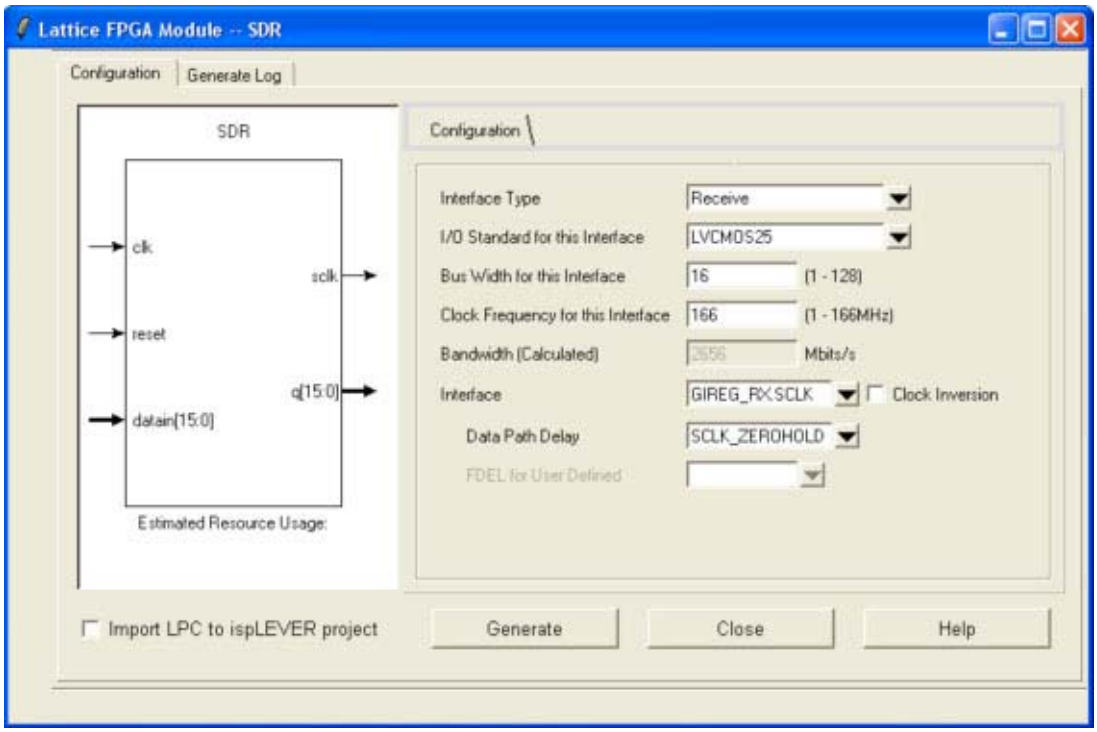


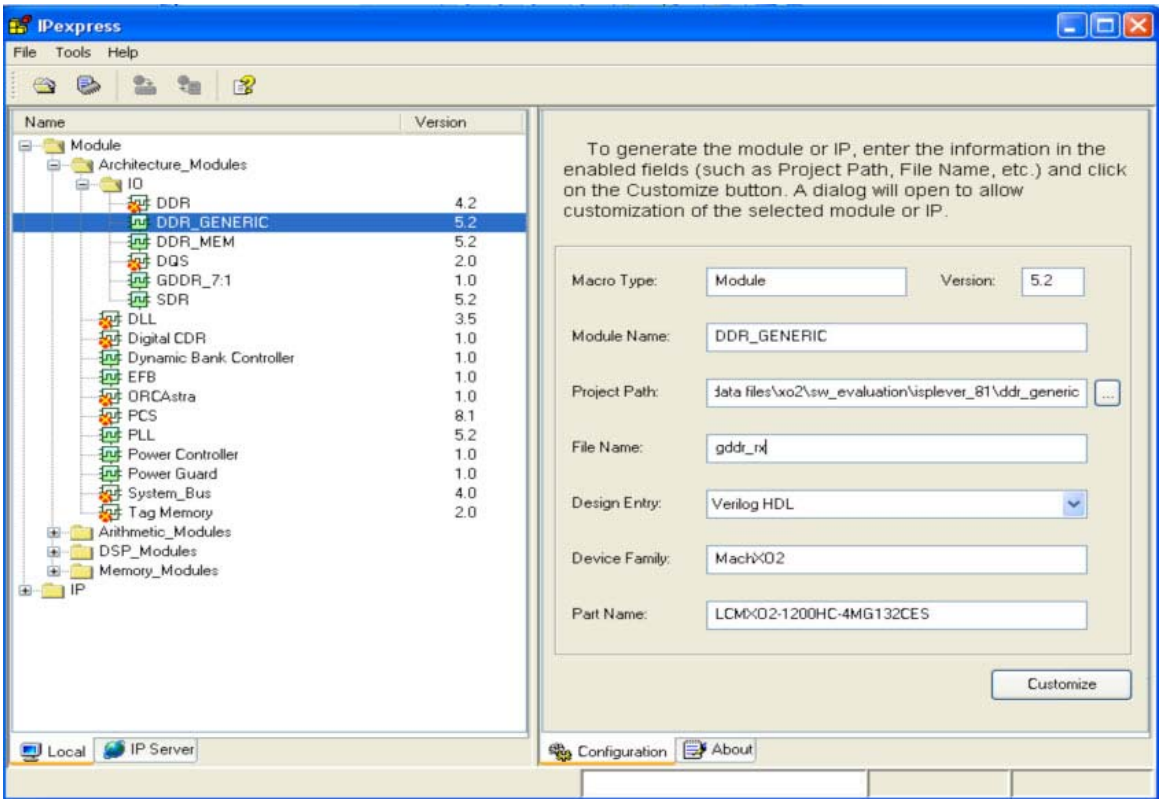
Table 11-4. GUI Options for the SDR Interfaces

GUI Option	Description	Range	Default Value
Interface Type	Types of interfaces	Transmit, Receive	Receive
I/O Standard for this interface	I/O standard to be used for the interface.	Supports all I/O types per selected Interface Type	LVCMOS25
Bus Width for this Interface	Bus size for the interface.	1-128	16
Clock Frequency for this Interface	Speed at which the interface will run	1-166MHZ	166MHz
Interface Bandwidth (calculated)	Calculated from the clock frequency entered.	(calculated)	(calculated)
Interface	Interface selected based on previous entries.	Transmit: GOREG_TX.SCLK Receive: GIREG_RX.SCLK	GIREG_RX.SCLK
Clock Inversion	Option to invert the clock input to the I/O register.	DISABLED, ENABLED	DISABLED
Data Path Delay	Data input can be optionally delayed using the DELAY block.	Bypass, SCLK_ZEROHOLD, User Defined	Bypass
FDEL for User Defined	If Delay type selected above is “User Defined”, delay values can be entered with this parameter.	Delay0 to Delay31	Delay0

Building DDR Generic Interfaces

As shown in Figure 11-27, users can choose interface type DDR_Generic, enter module name and click **Customize** to open the configuration tab.

Figure 11-27. DDR_Generic Interface Selection at the IPexpress Main Window



DDR_Generic interfaces have a Pre-Configuration Tab and a Configuration Tab. The Pre-Configuration Tab allows users to enter information about the type of interface to be built. Based on the entries in the Pre-Configuration Tab, the Configuration Tab will be populated with the best interface selection. The user can also, if necessary, override the selection made for the interface in the Configuration Tab and customize the interface based on design requirements. The following figures show the two tabs of the DDR_Generic modules in IPexpress. Tables 11-5 and 11-6 list the various configuration options available for DDR_Generic modules.

Figure 11-28. Pre-Configuration Tab of the DDR_Generic Interfaces

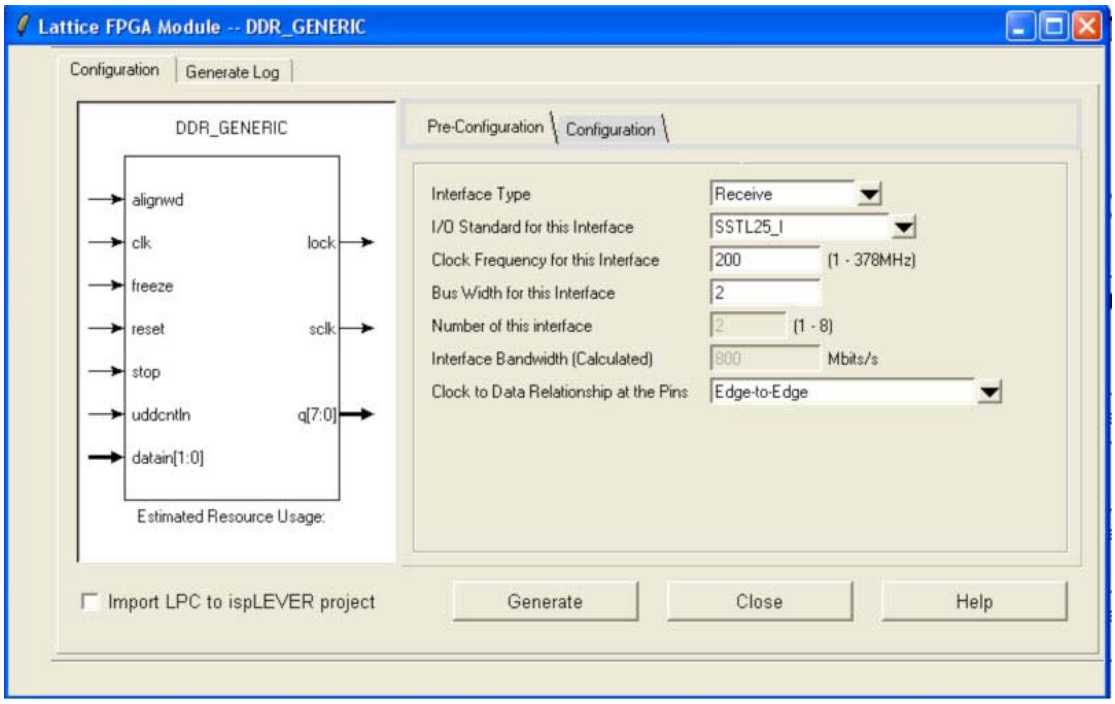
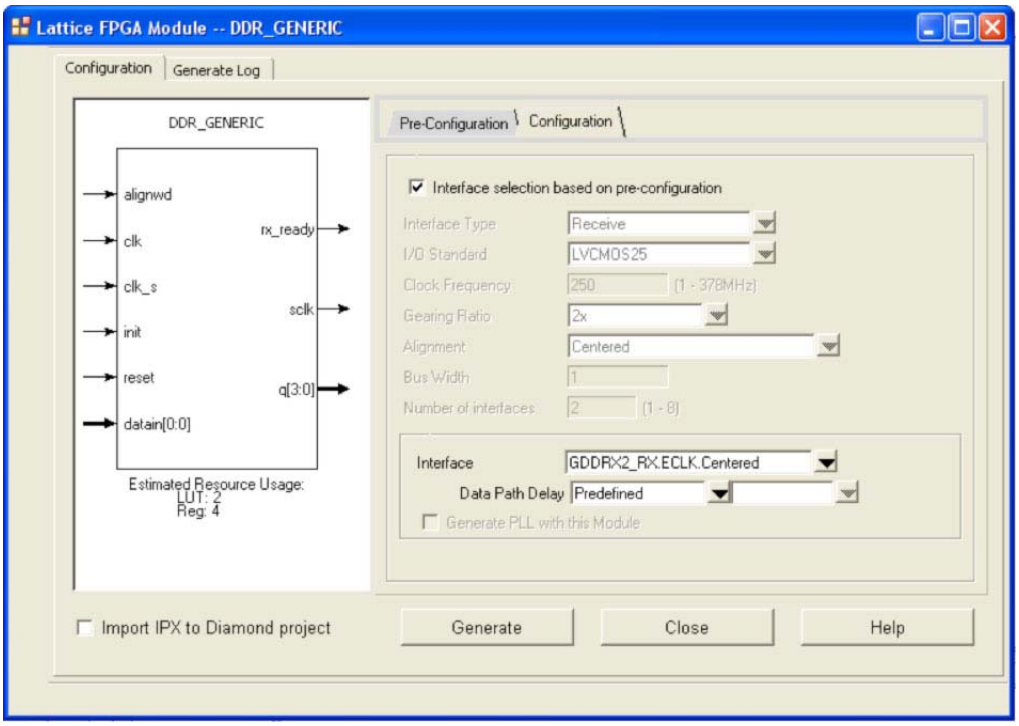


Table 11-5. GUI Options for the Pre-Configuration Tab of DDR_Generic Modules

GUI Option	Description	Range	Default Value
Interface Type	Types of interface	Transmit, Receive	Note 1
I/O Standard for this interface	I/O Standard for this interface	Supports all I/O types per selected Interface Type	LVCMOS25
Clock Frequency for this Interface	Speed at which the interface will run	1-378MHz (for HP) 1-266MHz (for LP)	Note 1
Bus Width for this Interface	Bus size for the interface	Various depending on the interface selected	Note 1
Number of this Interface	Maximum number of buses supported	(calculated)	(calculated)
Interface Bandwidth (calculated)	Calculated from the clock frequency and bus width	(calculated)	(calculated)
Clock to Data Relationship at the Pins	Select the type of external interfaces	Edge-to-Edge, Centered	Note 1

1. All fields of the Pre-Configuration tab are blank as default.

Figure 11-29. Configuration Tab of the DDR_Generic Modules



Based on the selections made in the Pre-Configuration Tab, the Configuration Tab is populated with the selections as shown in Figure 11-30. The checkbox at the top of this tab indicates that the interface is selected based on entries in the Pre-Configuration Tab. The user can choose to change these values by disabling this entry. Note that IPexpress chooses the most suitable interface based on selections made in the Pre-Configuration Tab.

Table 11-6. GUI Options of the Configuration Tab of the DDR_Generic Modules

GUI Option	Description	Range	Default Value
Interface Selection based on pre-configuration	Indicates interface is selected based on selection made in the Pre-configuration tab. Disabling this checkbox allows users to select gearing ratio, delay types etc.	ENABLED, DISABLED	ENABLED
Interface Type	Types of interfaces	Transmit, Receive	Receive
I/O Standard	I/O standard for this interface	All I/O types per selected Interface Type	LVCMOS25
Clock Frequency	Speed at which the interface will run	1-378 MHz (for HP) 1-266 MHz (for LP)	200MHz 100MHz
Gearing Ratio	Choose the gearing ratio of the interface	x1, x2, x4	x1
Alignment	Determine the type of external interfaces	Edge-to-Edge, Centered	Edge-to-Edge
Bus Width	Bus size for the interface	1-128	4
Number of Interfaces	Maximum number of buses supported	1-8	calculated
Interface	A list of the supported GDDR interfaces	Dependent of the Gearing ratio and Alignment choice	GDDRX1_RX.SCLK.Aligned
Data Path Delay[1]	Data input can be optionally delayed using the DELAY block.	Bypass, Predefined, User defined, Dynamic	Predefined
Generate PLL with this Module[2]	Option to generate PLL with this module or not to generate PLL with this module.	Enabled, Disabled	Enabled

1. When “User Defined” is selected, the delay value field will be enabled to allow users to select the delay values 0 to 31. When “Dynamic” is selected, a 5-bit delay port will be added to the module. “Dynamic” can only be used for x2 and x4 receive interfaces.
2. This option is only available for interfaces that are using a PLL. This includes, GDDRX1_RX.SCLK.Aligned, GDDRX1_TX.SCLK.Centered, GDDRX2_TX.ECLK.Centered, and GDDRX4_TX.ECLK.Centered interfaces.

If the Pre-Configuration tab is used, the gearing ratio of the interface is determined by the speed of the interface. Table 11-7 shows how the gearing ratio is selected.

Table 11-7. Gearing Ratio Selection by the Software

Device Type	Speed of the Interface	Gearing Ratio
High Performance (HP) devices	=< 166MHz	x1
	> 166MHz and =< 266MHz	x2
	> 266MHz	x4
Low Power (LP) devices	=< 70MHz	x1
	>70MHz and =< 133MHz	x2
	>133MHz	x4

Building a Generic DDR 7:1 Interface

As shown in Figure 11-30, users can choose interface type GDDR_71, enter module name and click **Customize** to open the Configuration tab. The Configuration Tab GUI options are listed in this section. The DDR 7:1 interface is a very specific application so the GUI options are relatively simple. Most of the necessary logic is built into the software for ease of use.

Figure 11-30. GDDR_71 Interface Selection at the IPexpress Main Window

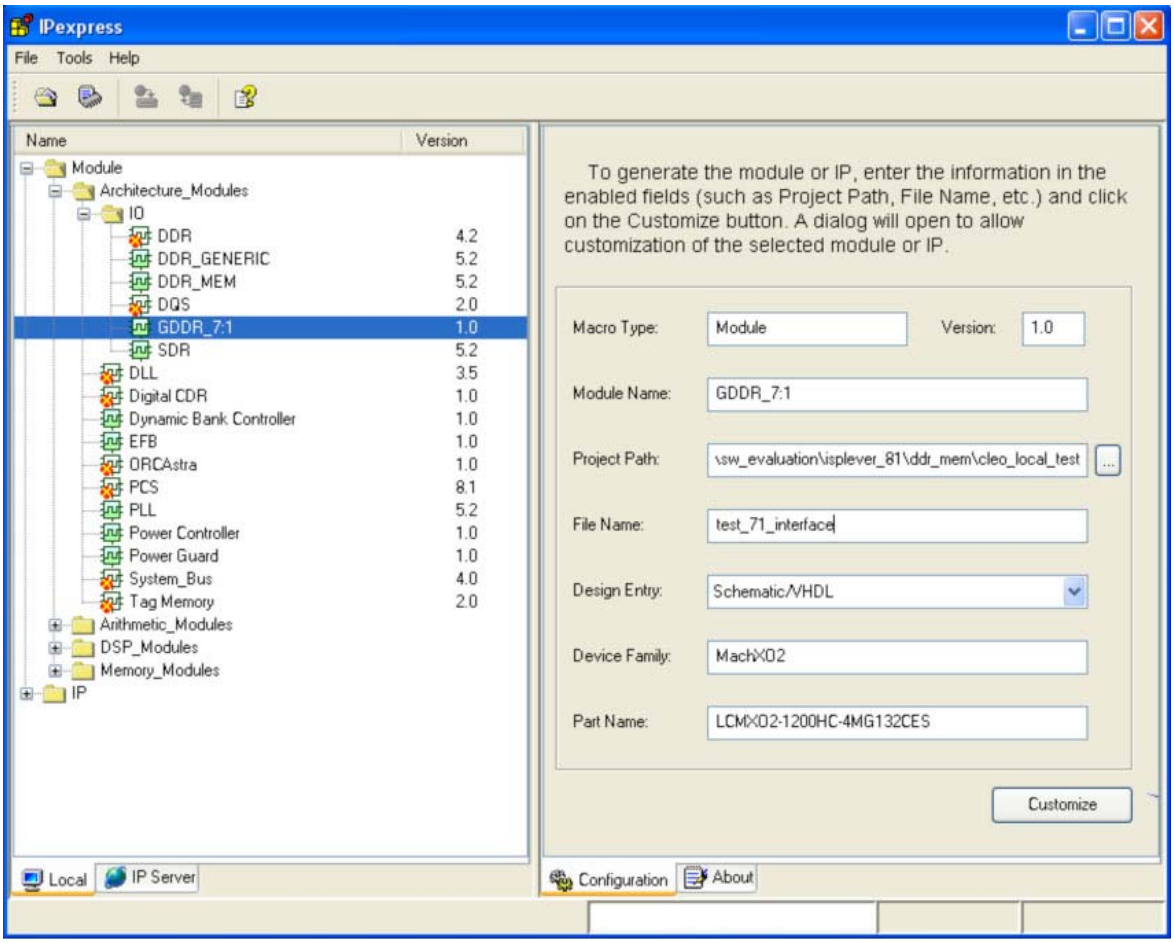


Figure 11-31. Configuration Tab of GDDR_71

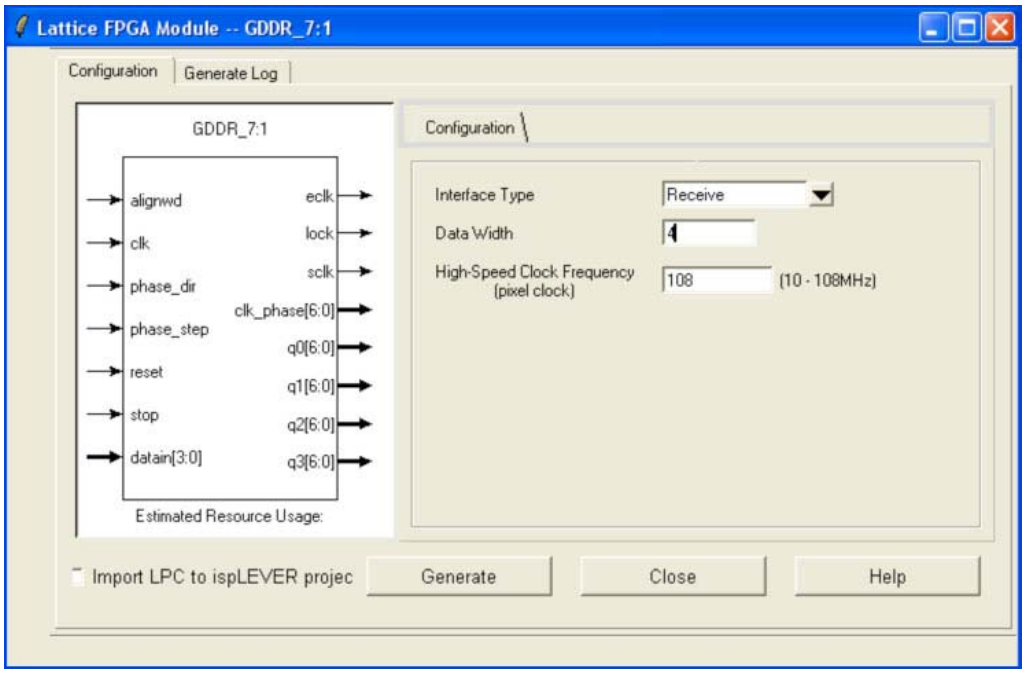


Table 11-8. GUI Options of the Configuration Tab for GDDR_71

GUI Option	Description	Range	Default Value
Interface Type	Types of interfaces	Transmit, Receive	Receive
Data Width	The number of incoming channels	1-16	4
High Speed Clock Frequency (Pixel Clock)	Pixel clock frequency for 7:1 LVDS interface	10-108	108

Generic High-Speed DDR Design Guidelines

I/O Logic Cells and Gearing Logic

Each Programmable IO Cell (PIC) has four programmable I/Os (PIOs), which form two pairs of I/O buffers. Each PIO by itself can support a x1 gearing ratio. A pair of PIOs, either the A/B pair or C/D pair, can support a x2 gearing ratio. Support of a x4 or 7:1 gearing ratio will take up all four PIOs in one PIC block. The x4 or 7:1 gearing ratio can only be supported when the A/B pair pins are available in the package, and are independent of the availability of the C/D pins. The total number of x2 interfaces available in a specific package is determined by the total number of A/B and C/D pairs. The total number of x4/7:1 interfaces available in a specific package is determined by the total number of A/B pairs.

Table 11-9. Gearing Logic Supported by Mixed Mode of I/O Logic Cells

	I/O Logic A	I/O Logic B	I/O Logic C	I/O Logic D
x2 gearing (A/B pair)	IDDRX2 or ODDRX2	Not available	Basic I/O registers or x1 gearing	Basic I/O registers or x1 gearing
x2 gearing (C/D pair)	Basic I/O registers or x1 gearing	Basic I/O registers or x1 gearing	IDDRX2 or ODDRX2	Not available
x4 gearing	IDDRX4 or ODDRX4	Not available	Basic I/O registers or x1 gearing	Basic I/O registers or x1 gearing
7:1 gearing	IDDRX71 or ODDRX71	Not available	Basic I/O registers or x1 gearing	Basic I/O registers or x1 gearing

High-Speed ECLK Bridge

The high-speed ECLK bridge is used to enhance communication of ECLKs across the MachXO2 device and is mainly used for high-speed video applications. It is available on MachXO2-640U, MachXO2-1200/U and higher density devices. The bridge allows a clock source to drive the edge clocks on the top and bottom edges of the device with minimal skew. The inputs to the bridge include primary clock pins from the top and bottom sides, PLL outputs from both sides, and clock tree routings.

Two bridge muxes are available in the ECLK bridge: one for each ECLK on the same side of the device. These muxes allow dynamic switching between two edge clocks. Refer to TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide, for ECLK bridge connectivity details.

The ECLK bridge supports all the generic high-speed interfaces except the high-speed x2 and x4 receive interfaces. The ECLK bridge component must be instantiated in the design in order to use the bridge function or to use it for routing purpose.

Reset Synchronization Requirement

The generic DDR interfaces are built with multiple dedicated circuits that are optimized for high-speed applications. It is therefore necessary to make sure all the components, such as CLKDIV and IDDR/ODDR, start with the same high-speed edge clock cycle to maintain the clock domain crossing margin between ECLK and SCLK, and to avoid bus bit-order scrambling due to the various delay of the reset pulse.

The ECLKSYNCA component and a particular reset sequence are required to guarantee a successful system implementation for interfaces using x2, x4, and 7:1 gearings. Figures 11-32 and 11-33 show the timing requirements for receive interfaces and transmit interfaces. The RX_STOP or TX_STOP signal must be connected to the STOP port of the ECLKSYNCA component. The RX_RST or the TX_RST should be connected to the reset port of the ODDR/IDDR and CLKDIV components. The RX_ECLK or TX_ECLK are the outputs of the ECLKSYNCA components. It is necessary to have a minimum of two ECLK cycles between the RST and STOP signals as shown in the figures below. The receive interface reset process should not start until the transmit interface reset process is complete in a loopback implementation. The clock signal used to generate the minimum two ECLK delay can be any clock that is slower than the ECLK frequency.

Figure 11-32. Reset Synchronization for Receive Interfaces

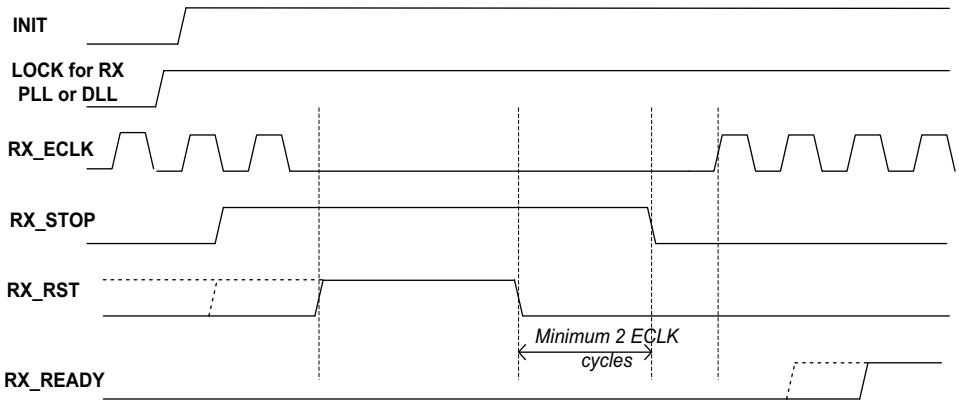
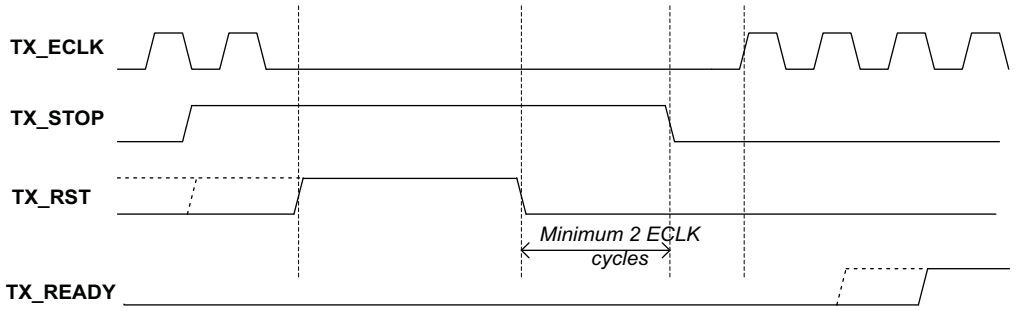


Figure 11-33. Reset Synchronization for Transmit Interfaces



These timing requirements are built into the generic DDR x2/x4/7:1 modules when they are generated by IPexpress. The RX_STOP/TX_STOP, RX_RST/TX_RST, and RX_ECLK/TX_ECLK are internal signals when the modules are generated by IPexpress. The reset timing requirements must be followed and implemented in RTL code when the generic DDR interfaces are built outside of IPexpress.

Timing Analysis for High-Speed GDDR Interfaces

It is recommended that users run Static Timing Analysis in the software for each of the high-speed interfaces. This section describes the timing preferences to use for each type of interface and the expected Trace results. The preferences can either be entered directly in the preference file (.lpf file) or through the Spreadsheet View graphical user interface.

The External Switching Characteristics section of the MachXO2 Family Data Sheet should be used along with this section. The data sheet specifies the actual values for these constraints for each of the interfaces.

Frequency Constraints

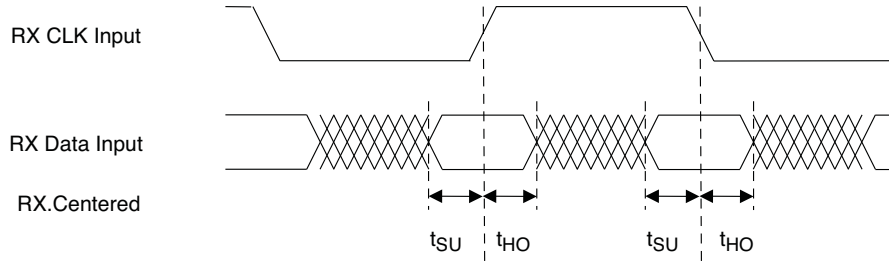
It is required that the user explicitly specify FREQUENCY (or PERIOD) PORT preferences to all input clocks in the design. This preference may not be required if the clock is generated out of a PLL or DLL or is input to a PLL or DLL. Refer to the High-Speed GDDR Interface Details section of this document for all the clock pin and clock routing requirements.

Setup and Hold Time Constraints

All of the receive interfaces can be constrained with setup and hold preferences.

Receive Centered Interface: Figure 11-34 shows the data and clock relationship for a Receive Centered Interface. Since the clock is centered to the data, it often provides sufficient setup and hold time at the device interface.

Figure 11-34. Receiver RX.CLK.Centered Waveforms



Users must specify in the software preference the amount of setup and hold time available. These parameters are listed in the figure as t_{SU} (setup time) and t_{HO} (hold time). They can be directly provided using the INPUT_SETUP and HOLD preference as shown below:

```
INPUT_SETUP PORT “Data” <tSU> ns HOLD <tHO> ns CLKPORT “CLK”;
```

Where: Data = Input Data Port; CLK = Input Clock Port.

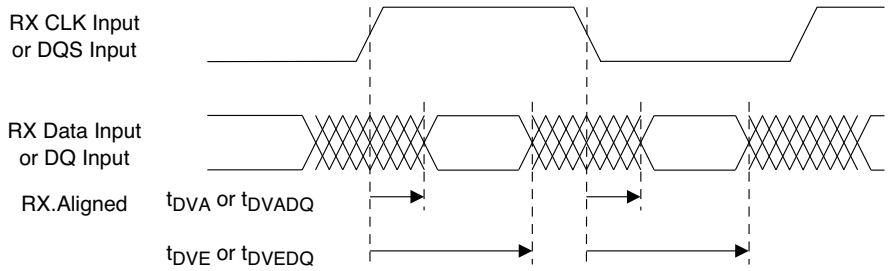
The External Switching Characteristics section of the MachXO2 Family Data Sheet specifies the minimum setup and hold times required for each of the high-speed interfaces running at maximum speed. For designs not running at the maximum speed, the Static Timing Analysis tool in the software can be used to calculate the setup and hold time values.

Using a GDDRX2_RX.ECLK.Centered interface running at 250MHz as an example, the preference can be set like this. The software will provide the minimum requirement of t_{SU} and t_{HO} for the interface.

```
INPUT_SETUP PORT Data 0.500000 ns HOLD 0.500000 ns CLKPORT "CLK";
```

Receive Aligned Interface: Figure 11-35 shows the data and clock relationship for a Receive Aligned Interface. The clock is aligned edge-to-edge with the data. The DDR memory at the receive side has the same timing behavior.

Figure 11-35. Receiver RX.CLK.Aligned and MEM DDR Input Waveforms



The worst case data may occur after the clock edge, and therefore has a negative setup time when entering the device. For this interface, the worst case setup time is specified by t_{DVA}, which is the data valid after the clock edge. The worst case hold time is specified as t_{DVE}, which is the data hold after clock. The setup and hold time for this interface can be specified as below.

```
INPUT_SETUP PORT Data <-tDVA > ns HOLD < tDVE> ns CLKPORT “CLK”;
```

Where: Data = Input Data Port; CLK= Input Clock Port

A negative number is used for SETUP time as the data occurs after the clock edge in this case. The External Switching Characteristics section of the MachXO2 Family Data Sheet specifies the maximum t_{DVA} and minimum t_{DVE} values required for each of the high-speed interfaces running at maximum speed. The data sheet numbers for this preference are listed in UI (Unit Intervals). One UI is equal to half of the clock period. Hence these numbers will need to be calculated from the clock period used.

For the GDDRX2_RX.ECLK.Aligned interface running at a speed of 250MHz (UI = 2.0ns)

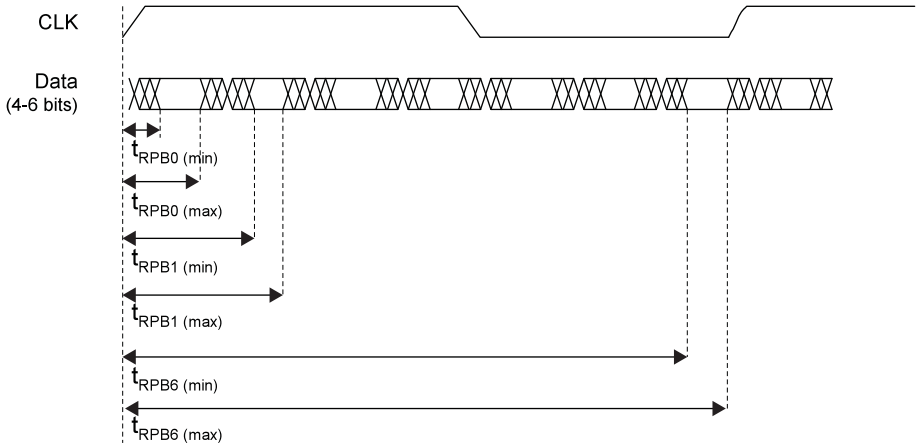
t_{DVA} = 0.32UI = 0.64ns, t_{DVE} = 0.70UI = 1.4ns

The preference for this case is:

```
INPUT_SETUP PORT Data -0.640000 ns HOLD 1.400000 ns CLKPORT "CLK";
```

Receive 7:1 LVDS Interface: The 7:1 LVDS interface is a unique GDDR interface, which uses one cycle of the pixel clock to align the seven data bits. Figure 11-36 shows the timing of this interface, where t_{RPBi} is the input stroke position for bit i. For this interface, the maximum setup time for bit0 is specified by the t_{RPB0} min, while the minimum hold time is specified as t_{RPB0} max. The t_{RPBi} min and t_{RPBi} max form the boundary of the input strobe position for biti of this interface. The values can be found in the External Switching Characteristics section of the MachXO2 Family Data Sheet.

Figure 11-36. Receiver GDDR71_RX. Waveforms



It is recommended to use the MachXO2 Display Interface Reference Design (RD1093) for 7:1 interface implementation.

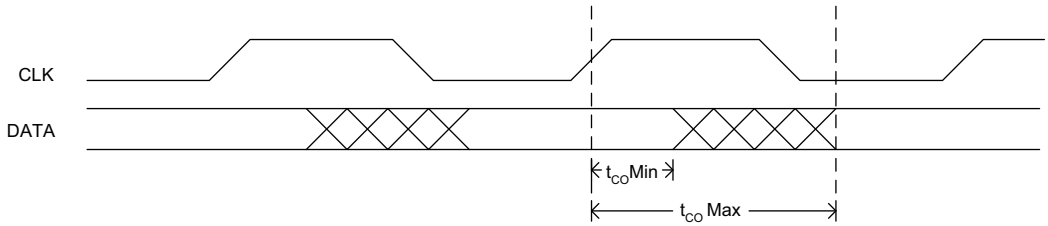
Receive Dynamic Interfaces: Static Timing Analysis will not show timing for all the dynamic interface cases as the either the clock or data delay will be dynamically updated at run time.

Clock-to-Out Constraints

All of the transmit (TX) interfaces can be constrained with clock-to-out constraints to detect the relationship between the clock and data when leaving the device.

Figure 11-37 shows how the clock-to-out is constrained in the software. Minimum t_{CO} is the minimum time after the clock edge transition that the data will not transit. So any data transition must occur between the t_{CO} minimum and maximum values.

Figure 11-37. t_{CO} Minimum and Maximum Timing Analysis



t_{CO} Min = Data cannot transition BEFORE Min
t_{CO} Max = Data cannot transition AFTER Max

Transmit Centered Interfaces: The transmit clock is expected to be centered with the data when leaving the device. Figure 11-38 shows the timing for a centered transmit interface. DDR memory transmit side has the same timing behavior as the transmit centered interface.

Figure 11-38. Transmitter TX.CLK.Centered and MEM DDR Output Waveforms

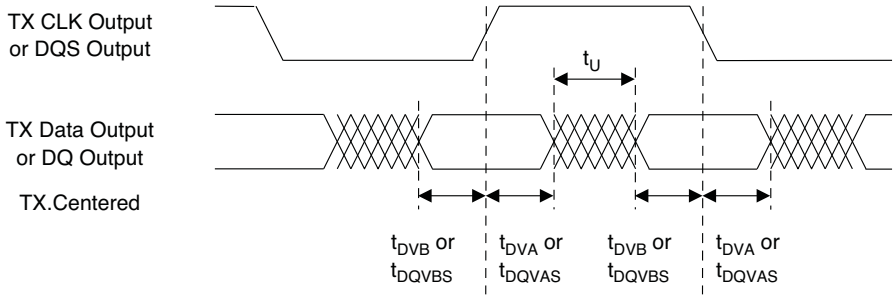


Figure 11-38 shows that the maximum value after which the data cannot transit is $-t_{DVB}$. The t_{DVB} is also the data valid before clock edge value. The minimum value before which the data cannot transition is $-(t_U + t_{DVB})$, where t_U is the period of time the data is in transition. This is also the data valid after clock value. A negative sign is used because in this particular case where clock is forwarded centered-aligned to the data, these two conditions occur before the clock edge.

The MachXO2 Family Data Sheet specifies the t_{DVB} and t_{DVA} values at maximum speed. But we do not know the t_U value, so the minimum t_{CO} can be calculated using the following equations

t_{CO} Min. = $-(t_{DVB} + t_U)$

$\frac{1}{2}T = t_{DVA} + t_{DVB} + t_U$

$-(t_{DVB} + t_U) = t_{DVA} - \frac{1}{2}T$

t_{CO} Min. = $t_{DVA} - \frac{1}{2}T$

The clock-to-out time in the software can be specified as:

```
CLOCK_TO_OUT PORT “Data” MAX <-tDVB> MIN <tDVA-1/2 Clock Period> CLKPORT “CLK”
CLKOUT PORT “Clock”;
```

Where: Data = Data Output Port; Clock = Forwarded Clock Output Port; CLK = Input Clock Port

The values for t_{DVB} and t_{DVA} can be found in the External Switching Characteristics section of the MachXO2 Family Data Sheet for the maximum speed.

For a GDDRX2_TX.SCLK.Centered interface running at 250MHz, the preference would be:

```
CLOCK_TO_OUT PORT "Data" MAX -0.670000 ns MIN -1.330000 ns CLKPORT "CLK" CLKOUT
PORT "Clock";
```

Transmit Aligned Interfaces: In this case, the clock and data are aligned when leaving the device. Figure 11-39 shows the timing diagram of this interface.

Figure 11-39. Transmitter TX.CLK.Aligned Waveforms

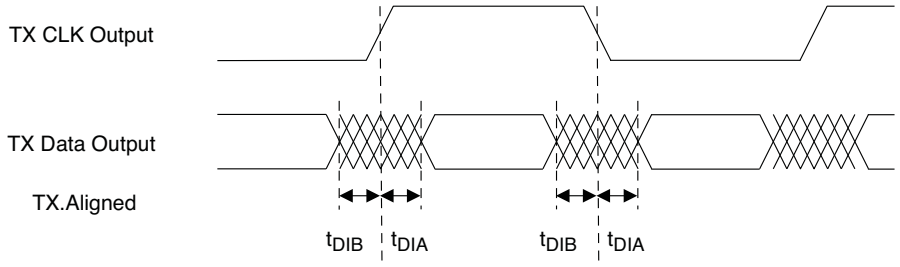


Figure 11-39 shows that maximum value after which the data cannot transition is t_{DIA}. This is the data invalid after the clock value. The minimum value before which the data cannot transition is $-t_{DIB}$, which is also the data invalid before the clock value. A negative sign is used for the minimum value because the minimum condition occurs before the clock edge.

The clock to out time in the software can be specified as:

```
CLOCK_TO_OUT PORT “Data” MAX <tDIA> MIN <-tDIB> CLKPORT “CLK” CLKOUT PORT “Clock”;
```

Where: Data = Data Output Port; Clock = Forwarded Clock Output Port; CLK = Input Clock Port

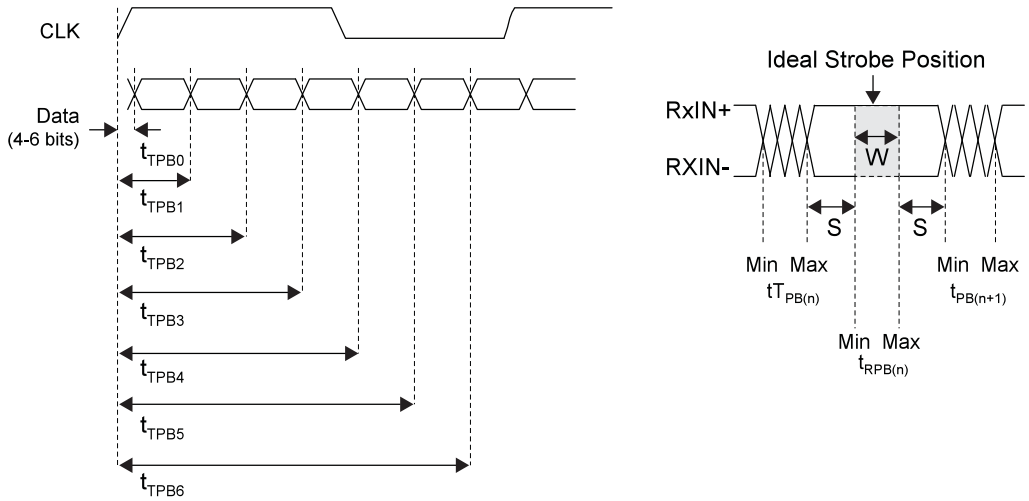
The t_{DIA} and t_{DIB} values are available in the External Switching Characteristics section of the MachXO2 Family Data Sheet for maximum speed.

For a GDDRX2_TX.Aligned interface running at 250MHz, $t_{DIA} = t_{DIB} = 0.215$ns. The preference would be:

```
CLOCK_TO_OUT PORT "Data" MAX 0.215000 ns MIN -0.215000 ns CLKPORT "CLK" CLKOUT
PORT "Clock”;
```

Transmit 7:1 LVDS Interface: The 7:1 LVDS interface is a unique GDDR interface, which uses one cycle of the pixel clock to transmit the seven data bits. Figure 11-40 shows the timing of this interface. For this interface, the transmit output pulse position for bit0 is bounded by the t_{TPB0} min and t_{TPB0} max. The values for t_{TPBi} can be found in the External Switching Characteristics section of the MachXO2 Family Data Sheet.

Figure 11-40. Transmitter GDDR71_TX. Waveforms



Timing Rule Check for Clock Domain Transfers

Clock Domain Transfers within the IDDR and ODDR modules are checked by Trace automatically when these elements are used in a design. Clock domain transfers occur in the GDDR X2, X4, 7:1 modules where there are fast-speed and slow-speed clock inputs.

No special preferences are needed to run this clock domain transfer check in the software. The clock domain transfer checks are automatically done by the software and reported in the Trace report under the section called “Timing Rule Check”. The report lists the timing for both input and output GDDR blocks where a clock domain transfer occurs.

DDR/DDR2/LPDDR SDRAM Interfaces Overview

A DDR SDRAM interface transfers data at both the rising and falling edges of the clock. DDR2 is the second generation of the DDR SRDRAM memory, whereas LPDDR is a low-power interface aimed at battery powered applications.

The DDR, DDR2 and LPDDR SDRAM interfaces rely on the use of a data strobe signal, DQS, for high-speed operation. The DDR and LPDDR SDRAM interfaces use a single-ended DQS strobe signal and the DDR2 interface has the option to use a differential DQS strobe. The figures below show typical DDR, DDR2, and LPDDR SDRAM interface signals. SDRAM interfaces are typically implemented with eight DQ data bits per DQS. An x16 interface will use two DQS signals and each DQS is associated with eight DQ bits. Both DQ and DQS are bi-directional ports and are used to read and write to the DDR memory devices.

When reading data from the external memory, data coming into the device is edge-aligned relative to the DQS signal. This DQS strobe signal needs to be phase shifted 90° before the FPGA logic can sample the read data. When writing to a DDR/DDR2/LPDDR SDRAM, the memory controller (FPGA) must shift the DQS by 90° to center-align with the data signals (DQ). A clock signal is also provided to the memory. This clock is provided as a differential clock (CLKP and CLKN) to minimize duty cycle variations. The memory also uses these clock signals to generate the DQS signal during a read via a DLL inside the memory. Note that the DLL that is typically used on standard DDR devices does not exist on LPDDR devices in order to save power. The figures below show DQ and DQS timing relationships for read and write cycles.

During read, the DQS signal is low for some duration after it comes out of tristate. This state is called “Preamble”. The state when the DQS is low before it goes into tristate is the “Postamble” state. This is the state after the last valid data transition.

DDR SDRAM also requires a Data Mask (DM) signal to mask data bits during write cycles. Note that the ratio of DQS to data bits is independent of the overall width of the memory. An 8-bit interface will have one strobe signal.

DDR SDRAM interfaces use the SSTL25 Class I/II I/O standards, DDR2 SDRAM interface uses the SSTL18 Class I/II, and LPDDR SDRAM interface uses the LVCMOS18 standards. DDR2 has an option to use either single-ended or differential DQS.

The following table and figures give an overview of the DDR memory specifications, typical interfaces, and pin-level DQ and DQS relationships.

Table 11-10. DDR / DDR2 and LPDDR Specification for MachXO2-640U, MachXO2-1200/U and Higher Density Devices

	DDR	DDR2	LPDDR
Data Rate	190 to 300Mbps	266 to 300 Mbps	0 to 300Mbps
DQS	Single Ended	Single Ended /Differential	Single Ended
Interface	SSTL25	SSTL18	LVCMOS18
Termination	External	On-die	None

Figure 11-41. Typical DDR SDRAM Interface

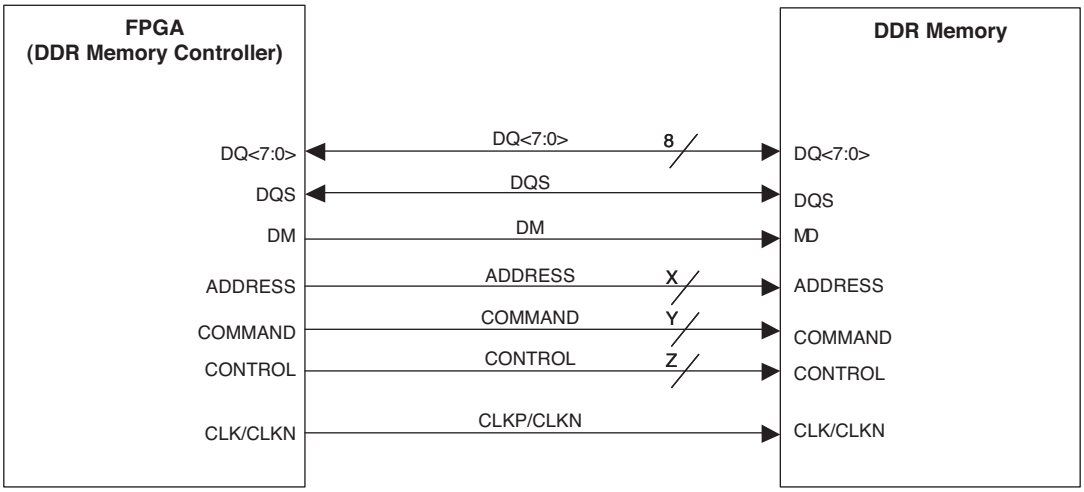


Figure 11-42. Typical DDR2 SDRAM Interface

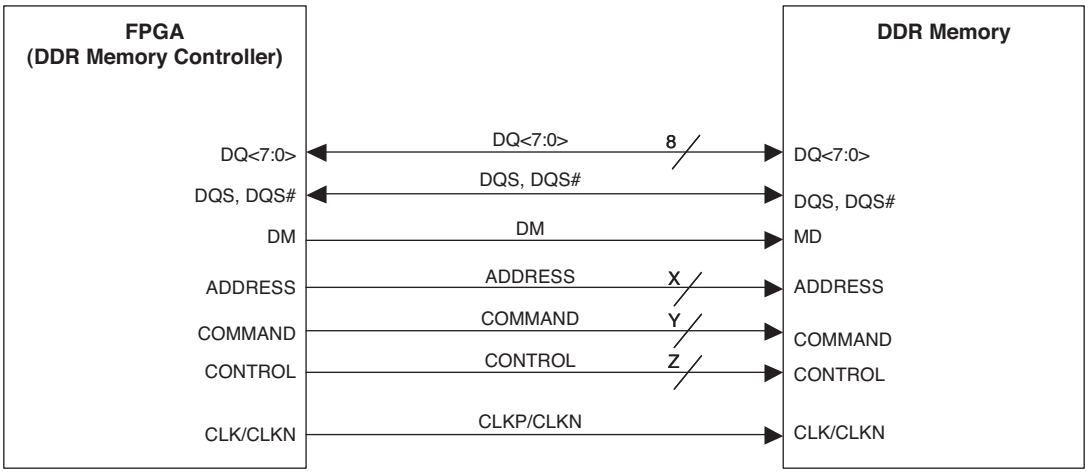


Figure 11-43. Typical LPDDR SDRAM Interface

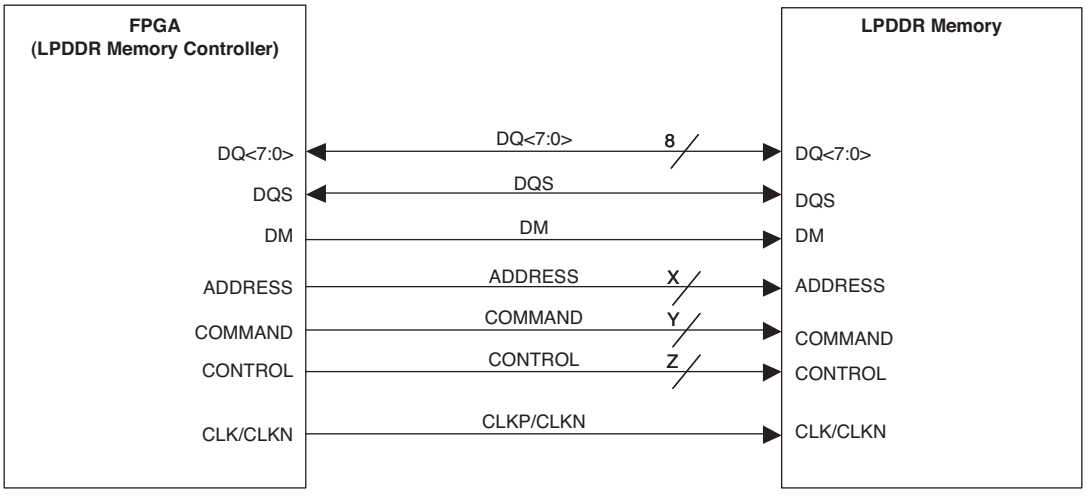


Figure 11-44. DQ-DQS Relationship During READ

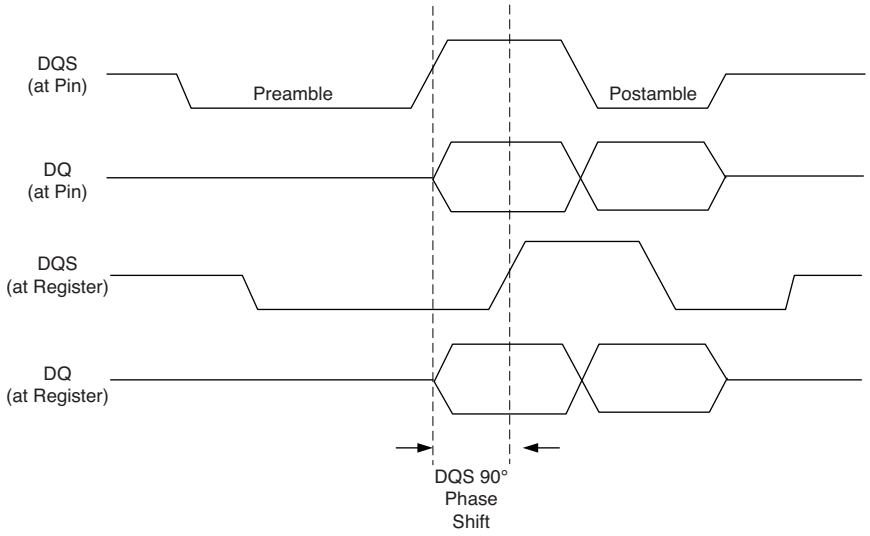
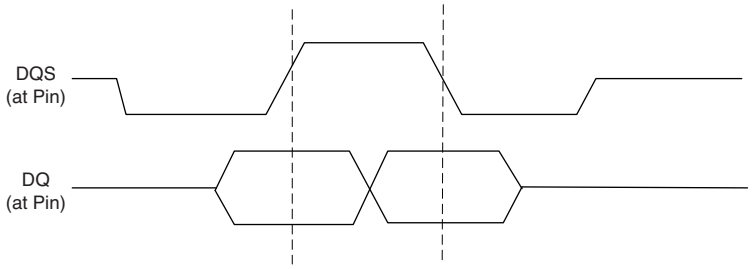


Figure 11-45. DQ-DQS Relationship During WRITE



DDR/DDR2/LPDDR SDRAM Interfaces Implementation

As described in the DDR memory overview section, all the DDR SDRAM interfaces rely on the use of a data strobe signal, DQS, for high-speed operation. When reading data from the external memory device, data coming into the MachXO2 device is edge-aligned with respect to the DQS signal. The MachXO2 will shift DQS by 90° before using it to sample the read data. When writing to a DDR SDRAM from the memory controller, the MachXO2 device must generate a DQS signal that is center-aligned with the DQ, the data signals. This is accomplished by ensuring a DQS strobe is shifted 90° relative to DQ data.

MachXO2-640U, MachXO2-1200/U and higher density devices have dedicated DQS support circuitry for generating appropriate phase shifting for DQS. This dedicated circuitry is independent of the generic DDR circuitry and is only available on the right side of the device. The circuitry uses x1 gearing for memory interfaces which results in the memory interface running at 2X frequency relatively to the core performance.

For both DDR2 and LPDDR, XO2 is targeted for embedded chip-to-chip implementation, which requires the fanout of DQ/DQS to be one, whereas fanout of CLKP/CLKN and address/command can potentially be two.

The DQS phase shift circuit uses a frequency referenced DLL to generate delay control signals associated with each of the dedicated DQS pins, and is designed to compensate for process, voltage and temperature (PVT) variations. The frequency reference is provided through one of the PCLK global clock pins. The dedicated DDR support circuit is also designed to provide comfortable and consistent margins for the data sampling window. This section describes how to implement the read and write sections of a DDR memory interface.

DQS Grouping

A DQS group generally consists of eight DQ pins, one DM pin, one DQS pin or two DQS/DQSN pairs, and one Vref pin. This requires at least 11 I/O logic cells to implement a complete DDR, DDR2, or LPDDR SDRAM interface. MachXO2-640U, MachXO2-1200/U and higher density devices support DQS-14 group to allow DQS signals to span across 12 I/O cells with two reserved for DQS and DQSN. The memory circuitry supported at the right side of the device allows up to 16-bit wide DDR/DDR2/LPDDR bus implementations. This requires two DQS group to be used together.

Each DQS signal spans across 14 I/Os. If a single-ended DQS is used, up to 13 I/Os spanned by the DQS can be used to implement an 8-bit DDR memory interface. The user can assign any eight of the I/O pads within the DQS-14 group to be the DQ data pins. If DQSN is not used, it can be used as a DQ or DM for the DQS group. The MachXO2 device family allows any I/O pin to be used as the Vref pin. This, together with the DQS-14 group structure, increases the flexibility of pin locking for DDR memory interfaces. Refer to the MachXO2 Family Data Sheet for the details of the DQS and DQS-14 group locations.

DQS Circuitry

The DQS circuitry (DQSBUF) is designed to simplify the implementation of DDR memory interfaces. It integrates several functions, including:

- DQS preamble and postamble management
- DLL-compensated DQS delay
- Data valid and data burst detection

DQS Preamble and Postamble Management
DDR/DDR2/LPDDR SDRAM memory interfaces require preamble and postamble states prior or proceeding the read or write bursts. The pre- and postamble management circuitry is built into the MachXO2 devices. The circuitry ensures clean DQS pulse inside the device, and turns on the DQS for internal logic during preamble, and turn off DQS after the last falling edge of the signal. The clean DQS inside the device helps to guarantee the downstream logic for clock polarity detection (DDRCLKPOL). This signal is used to control the polarity of the clock to the synchronizing registers.

DLL-Compensated Delay Logic
The master DLL (DQSDLL) works in conjunction with delay logic in the DQSBUF to provide the necessary 90° phase shift. The DQSBUF receives the 7-bit delay control code from the DQSDLL at the right side of the chip. The 7-bit delay control is used by the two clock slave delay lines in the DQSBUF at the right side of the device for DDR memory implementation. One is used for the 90° clock shift for the READ operation, and the other is to shift the data by 90° during the WRITE operation.

The DQS circuitry (DQSBUF) generates the DQSR90 by shifting the ECLK by 90°. The DQSR90 signal is then sent to all the DDR memory I/O logic cells at the right side of the chip for clocking the input data during memory READ operations.

DQSW90 is also generated by the DQSBUF and is used for memory WRITE operations. It is the 90° shift of the SCLK signal and is used to send out the DDR data. This meets the requirement of placing the DQS edge at the middle of the data opening of DQ during WRITE.

Data Valid and Data Burst Detection
DQSBUF is also responsible for generating the data valid signal which indicates to the FPGA that valid data is transmitted out the input DDR registers to the FPGA core. The data valid signal is level sensitive and matches data at the FPGA core boundary.

Data burst detection in the DQSBUF is used to position the read pulse in the optimized location for a DDR READ operation. It is critical that a sufficient burst length (BL) is used in the training process. The MachXO2 DDR memory

implementation requires at least two consecutive BL2 (BL=2), or one BL4 (BL=4), or a multiple of these burst lengths to be used during training session to allow the device to detect the correct read pulse position.

I/O Logic Data Path

The DDR memory I/O logic, or the memory PIO cell, is shown in Figure 11-2 at the beginning of the document. At the DDR memory input path, the first set of DDR registers is used to de-mux the DDR data at the positive and negative edges of the phase-shifted DQS signal. The second set is used to transfer the demuxed data from the DQS domain (DQSR90) to the SCLK domain. The final set of registers is used to clock the input data one more time with the SCLK based on DDRCLKPOL. At the output side, the DQS and DQ share the same logic. The parallel data is muxed out by the write clock, DQSW90. The DQS output pulse is controlled by SCLK.

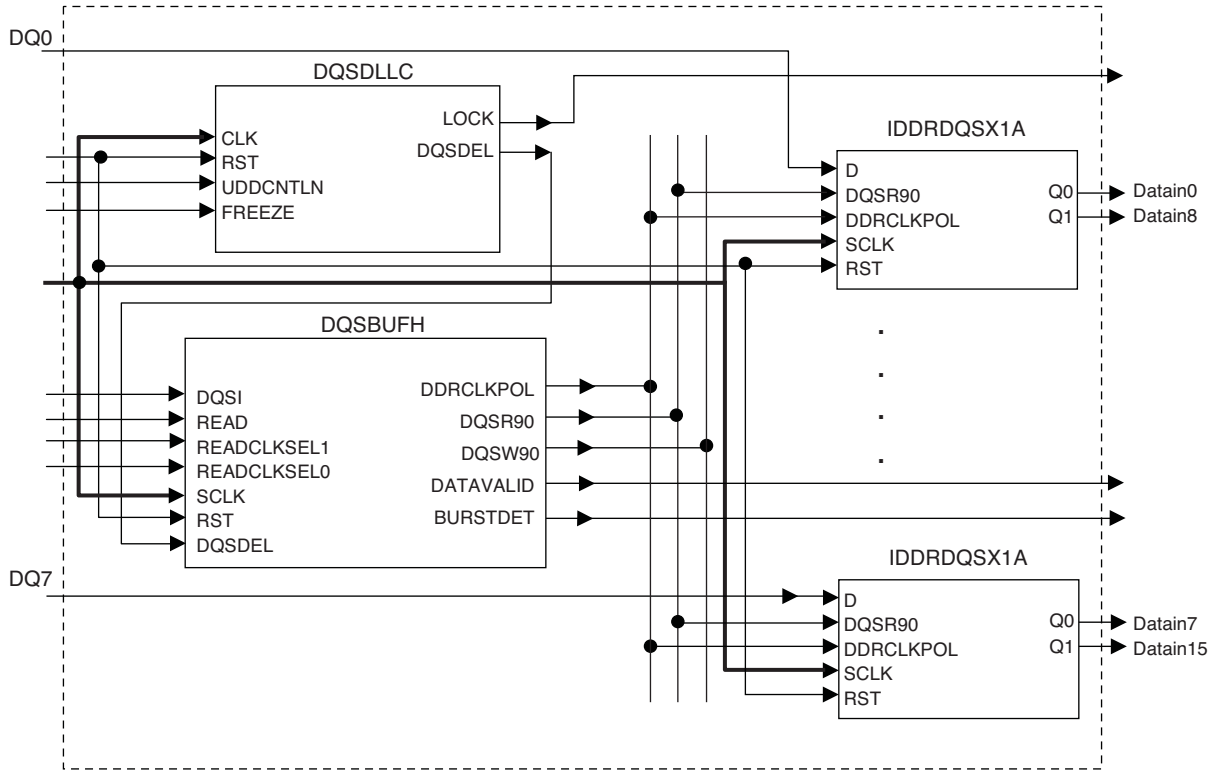
More details of the building blocks of the DDR memory can be found in the DDR Software Primitives and Attributes section.

DDR/DDR2/LPDDR Memory READ Implementation

MachXO2-640U, MachXO2-1200/U and higher density devices support the DDR/DDR2/LPDDR memory interface functions using the DDR memory module generated through the IPexpress tool. Using IPexpress, a designer can generate the different modules required to read the data coming from the DDR/DDR2 /LPDDR memory.

The DDR/DDR2/LPDDR read side is implemented using the following three software elements: DQSDLL represents the DLL used for calibration; IDDRDQS implements the input DDR registers; DQSBUF represents the DQS delay block, the clock polarity control logic and the data valid module. The DQSR90 is distributed to each IDDRDQS cell through a dedicated clock tree. The routing of the SCLK must use one of the primary clock nets.

Figure 11-46. DDR/DDR2/LPDDR READ Implementation



DDR/DDR2/LPDDR Memory WRITE Implementation

To implement the write portion of a DDR memory interface, two streams of single data rate data must be multiplexed together with data transitioning on both edges of the clock. In addition, DQS must arrive at the memory pins

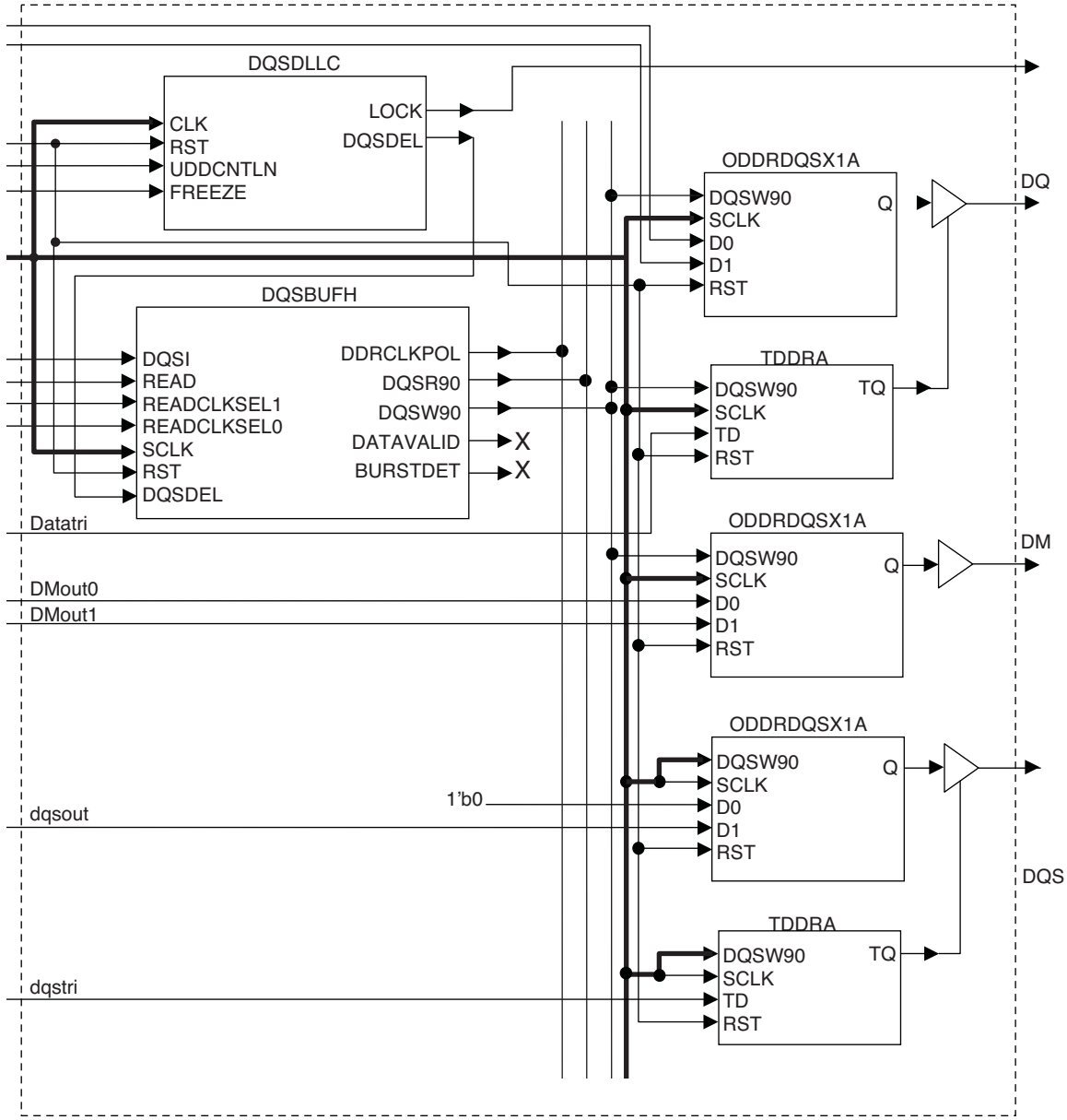
center-aligned with data (DQ), and edge-aligned with the differential output clock (CLKP/CLKN). Along with these signals, Address/Command and Data Mask (DM) signals also need to be generated. IPexpress should be used to generate this interface.

The DQSW90 is distributed to each ODDRDQS cell through a dedicated clock tree. The routing of the SCLK must use one of the primary clock nets.

DDR/DDR2/LPDDR Write Data (DQ) and Strobe (DQS) Generation

Figure 11-47 shows the DQ/DQS generation for a WRITE operation. DQS90 is a 90°-shifted version of SCLK. As discussed above, DQSW90 is used to shift the DQ instead of the DQS by 90°. Generation of the DQs and DM uses DQSW90, while generation of DQS uses SCLK. This ensures that the edge of DQS is in the middle of the DQ, and DM. DQS is single-ended for DDR and LPDDR, but differential for the DDR2.

Figure 11-47. DDR/DDR2/LPDDR WRITE Implementation

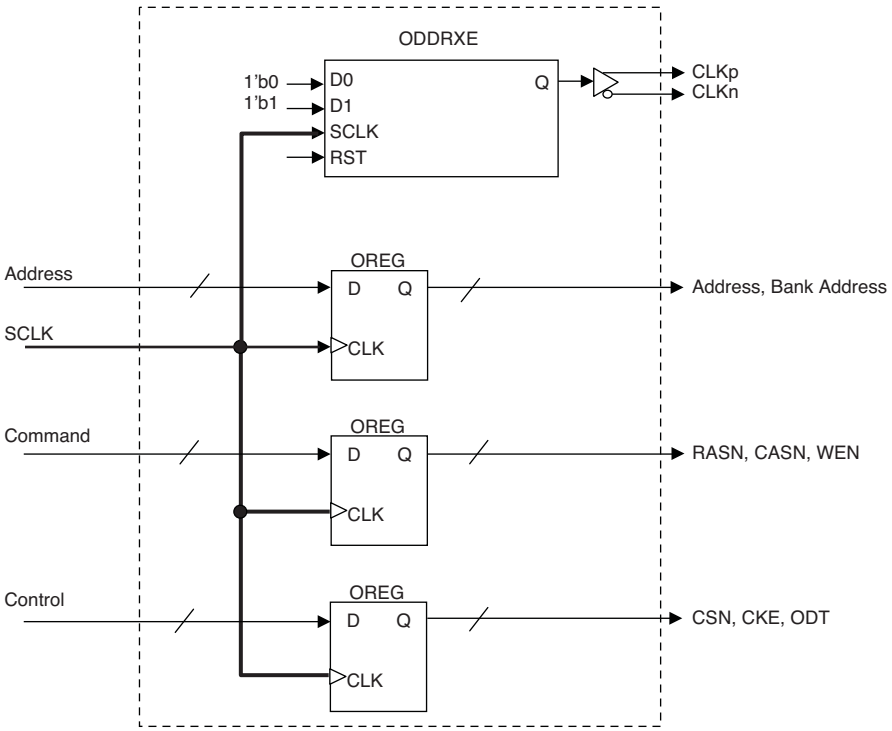


DDR/DDR2/LPDDR Write Clock, Address and Command Generation

The memory WRITE operation requires the controller to generate a differential clock, address lines, and commands for the SDRAM memory. The clock is generated using the ODDRX (x1) primitive. The inputs of the ODDRX primitive are tied to constants to generate an output clock. When interfacing to the DDR SDRAM memories, CLKP should be connected to the SSTL25D I/O standard. When interfacing to DDR2 memories it should be connected to SSTL18D I/O standard to generate the differential clock outputs. When interfacing to LPDDR SDRAM memories, CLKP should be connected to the LVCMOS18 I/O standard. Generating the CLKN in this manner will prevent skew between the two signals.

The address and command signals for DDR/DDR2/LPDDR SDRAM interfaces are generated using the regular output register OREG. The same SCLK is used for the DQ/DQS, as well as clock, address, and command generation. The routing of SCLK must use one of the primary clock nets.

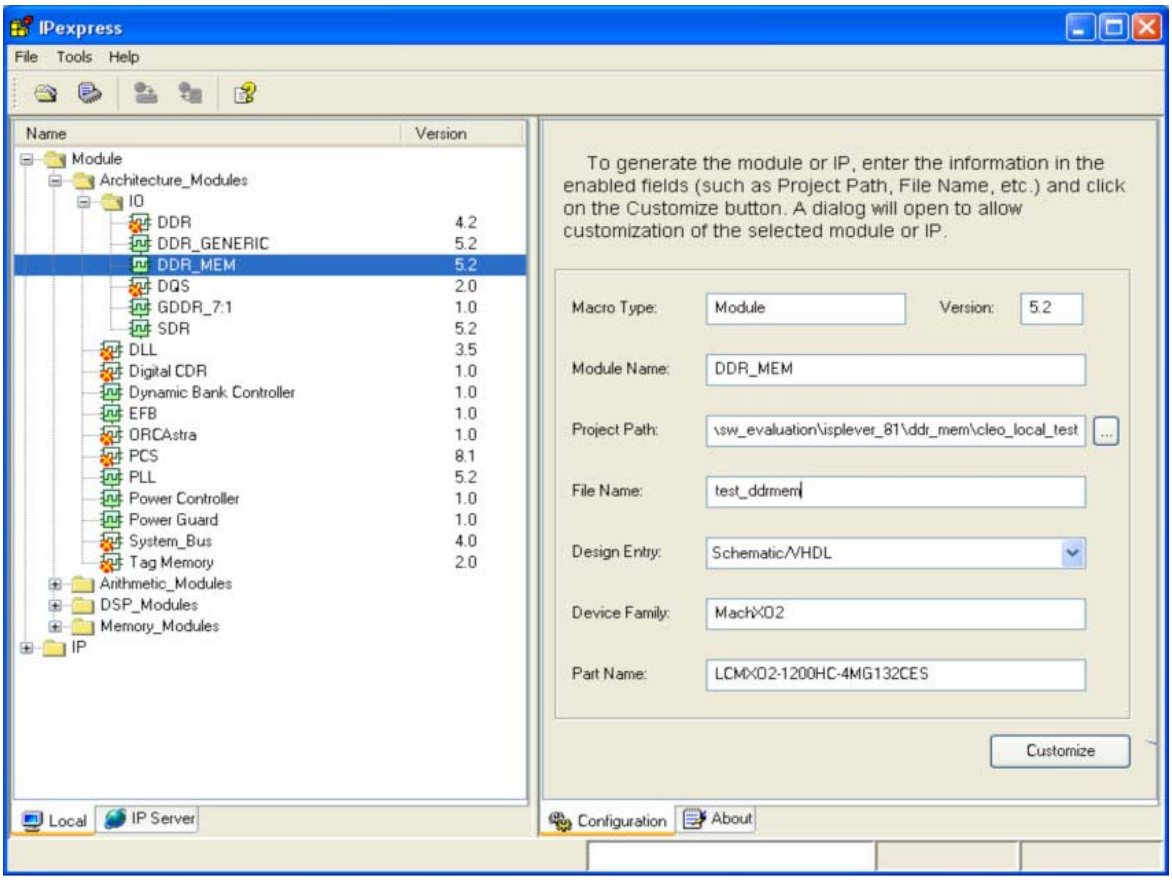
Figure 11-48. CLK, Address and Command Control Pin Generation



DDR Memory Interface Generation Using IPexpress

The IPexpress tool should be used to configure and generate all the DDR/DDR2/LPDDR interfaces described above. In the IPexpress GUI, all the DDR memory modules are located under **Architecture Modules > IO**. DDR_MEM is used to generate DDR memory interfaces as shown in Figure 11-49.

Figure 11-49. IPexpress Main Window



There are two tabs in the DDR_MEM module. The Configuration Tab shown in Figure 11-50 is used to choose among DDR, DDR2, and LPDDR and their corresponding parameters.

Figure 11-50. Configuration Tab for DDR_MEM

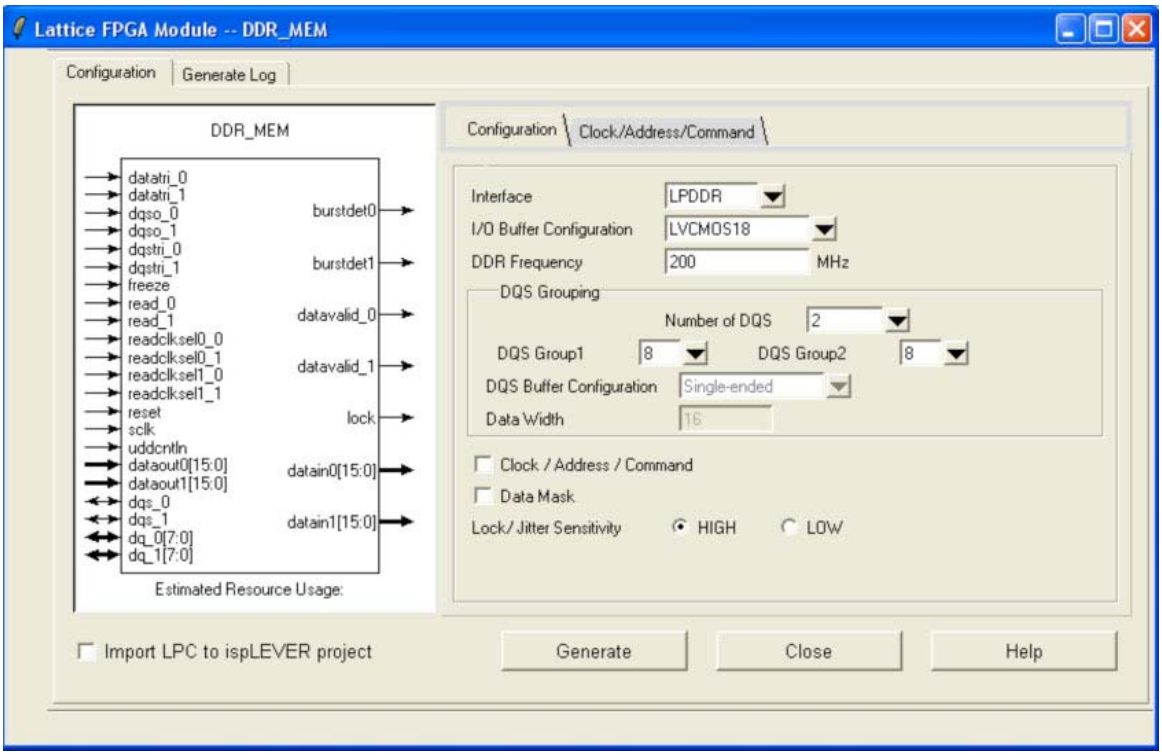


Table 11-11 details the parameters/options for each of the GUI options on the DDR_MEM Configuration Tab.

Table 11-11. Options of the Configuration Tab of the DDR_MEM Module

GUI Option	Description	Range	Default Value
Interface Type	Types of Interface	DDR, DDR2, LPDDR	DDR
I/O Buffer Configuration	I/O standard associated with each interface type. This is automatically set based on the interface selected	SSTL25_I, SSTL18_I, LVCMOS18	SSTL25_I
DDR Frequency	The speed the DDR memory interface is running.	DDR = 83-100MHz DDR2 = 125 - 150MHz LPDDR = 0-133MHz	DDR = 100 MHz DDR2 = 150 MHz LPDDR = 133 MHz
Number of DQS	Number of DQS groups available	1, 2	2
Number of DQ for DQS Group1 and DQS Group 2	Data width for the DQS group	1 to 8	8
DQS Buffer Configuration for DDR2	DQS buffer type	Single-ended, Differential	Single-ended
Data Width (calculated)	Data width for the interface	1-16	(calculated)
Clock/Address/Command	Clock/address/command interface will be generated when this option is checked.	Enabled, Disabled	Disabled
Data Mask	Data mask signal will be generated when this option is checked	Enabled, Disabled	Disabled
Lock/Jitter Sensitivity	Lock Sensitivity attribute for DQSDLL. Low means less sensitive to jitter.	High, Low	Low

If the user selects to generate the Clock/Address/Command signals using IPexpress, then the settings in the Clock/Address/Command Tab are active and can be set up as required.

Figure 11-51. Clock/Address/Command Tab for DDR_MEM

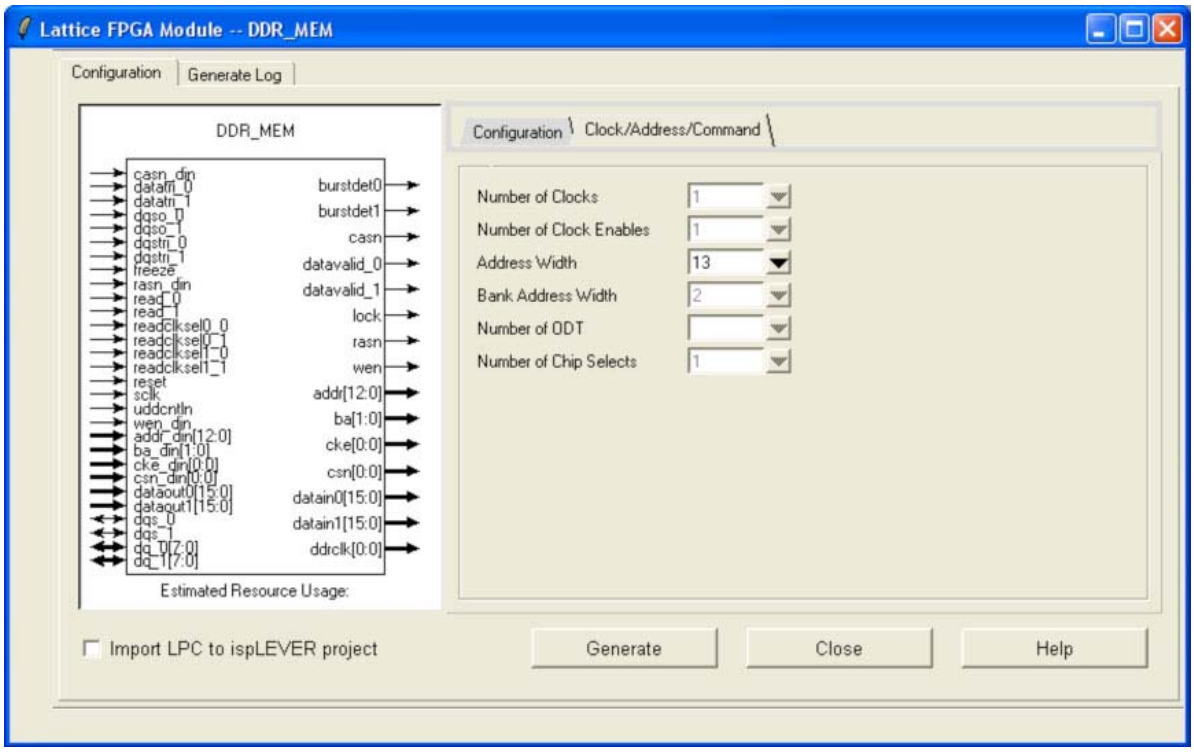


Table 11-12. Options of the Configuration Tab of the DDR_MEM Module

GUI Option	Range	Default Value
Number of Clocks	DDR/DDR2: 1, 2LPDDR: 1	1
Number of Clock Enables	DDR/DDR2: 1, 2 LPDDR: 1	1
Address Width	DDR: 12 – 14 DDR2: 13 – 16 LPDDR: 12 -14	DDR: 13DDR2: 14LPDDR: 13
Bank Address Width	DDR: 2 DDR2: 2, 3 LPDDR: 2	DDR: 2 DDR2: 2 LPDDR: 2
Number of ODT	DDR: blank DDR2: 1, 2 LPDDR: blank	DDR: Blank DDR2: 1 LPDDR: Blank
Number of Chip Selects	DDR / DDR2: 1, 2 LPDDR: 1	DDR/DDR2: 1 LPDDR: 1

DDR Memory DQ/DQS Design Rules and Guidelines

Listed below are some rules and guidelines for implementing DDR memory interfaces in MachXO2-640U, MachXO2-1200/U and higher density devices.

- DDR memory I/O logic and DQS circuitry are available for the bank on the right side of the device. Correspondingly, DDR memory interface can only be implemented at the right side bank of the device.
- There are two DQSDLLs in the device, one for the left and bottom banks, and one for the right and top banks. Only the one for right and top banks can be used for DDR memory interface implementations.
- The delay control code generated by the DQSDLL is applied to all the DDR memory I/O logic on the right side bank.
- When implementing a DDR1 SDRAM interface, all interface signals should be connected to the SSTL25 I/O standard.
- For DDR2 SDRAM interfaces, signals should be connected to the SSTL18 I/O standard.
- For LPDDR SDRAM interfaces, signals should be connected to the LVCMOS18 standard.

- All DDR memory interfaces require a differential clock signal. The clock signal should be connected to corresponding differential I/O standard specified by the memory interface.
- The use of differential DQS is optional for DDR2. If differential DQS is used it should be connected to SSTLD18 I/O standard.

DDR/DDR2/LPDDR Pinout Guidelines

- The DQS-DQ association rule must be followed.
  - All associated DQs (8 or 4) to a DQS must be in the same DQS-14 group.
- The data mask (DM) must be part of the corresponding DQS-14 group.
  - Example: DM[0] must be in the DQS-14 group that has DQ[7:0], DQS[0].
- DQS must be allocated to a dedicated DQS pad.
  - DQSN pad is used when differential DQS is selected.
- Do not assign any signal to the DQSN pad if SSTL18D is applied to the DQS pad.
  - The software automatically places DQS# when SSTL18D is applied.
- The clock to the PLL used to generate the outputs must be locked to the correct dedicated PLL pin input.
- In addition to the DQS-14 group, a DDR memory interface typically needs an additional 20-24 pins such as CLK/CLKN, address and command pins depending on memory density. Pins in banks other than the right side of the device can be used for these signals.
- The MachXO2 device family allows any I/O to be used as Vref pin. The Vref for DDR/DDR2 must be part of the DQS-14 group. Refer to TN1202, MachXO2 sysIO Usage Guide, for Vref assignment rules.

DDR Software Primitives and Attributes

Software primitives used for all the generic DDR interfaces and DDR memory interface implementation are discussed in this section. The primitives are divided according to their usage. Some of them are used for generic DDR interfaces only, some of them are for DDR memory DQS logic, and others are control functions shared by both generic or memory DDR interfaces. The DDR input primitives will be discussed first, followed by the DDR output primitives, then the DDR control logic primitives.

Table 11-13. MachXO2 DDR Software Primitives

Type	Primitive	Usage
Data Input	IDDRXE	Generic DDR x1
	IDDRX2E	Generic DDR x2
	IDDRX4B	Generic DDR x4
	IDDR71A	Generic DDR 7:1
	IDDRDQSX1A	DDR memory
Data Output	ODDRXE	Generic DDR x1
	ODDRX2E	Generic DDR x2
	ODDRX4B	Generic DDR x4
	ODDR71A	Generic DDR 7:1
	ODDRDQSX1A	DDR memory
DQS Tristate	TDDRA	DDR memory
DQSBUF Logic	DQSBUFH	DDR memory
DLL	DQSDLLC	Master DLL for Generic x2, x4, and DDR Memory
Input Delay	DELAYD	Delay block with dynamic control for Generic x2, x4
	DELAYE	Delay block with fixed delays for Generic DDR x1, x2, x4
	DLLDELC	Clock slave delay cell for Generic DDR x2, x4

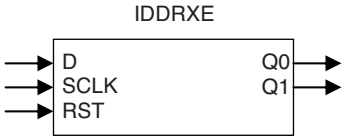
Input DDR Primitives

The input DDR primitives represent the modules used to capture both the GDDR data and the DDR data coming from a memory interface. There are several modes for the DDR input registers to implement different gearings for GDDR interfaces. The memory DDR uses a different primitive than the GDDR primitives. For all the data ports of the input primitives, Q0 of the parallel data is the first bit received.

IDDRXE

The primitive implements the input register block in x1 gearing mode. This is used only for the generic DDR x1 interface (gearing ratio 1:2) available on all sides of the MachXO2 devices. It uses a single clock source, SCLK, for the entire primitive so it does not involve a clock domain transfer.

Figure 11-52. IDDRXE Symbol

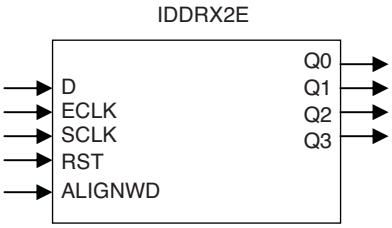


The internal register structure for this primitive is based on the basic PIO cell, as shown in Figure 11-1. The first set is the DDR register to capture the data at both edges of the SCLK. The second set is the synchronization registers to transfer the captured data to the FPGA core.

IDDRX2E

The primitive implements the input register block in x2 gearing mode. This is used only for the generic DDR x2 interface (gearing ratio 1:4) on the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices. Its registers are designed to use edge clock routing on the GDDR interface and the system clock for FPGA core. The edge clock is connected to the ECLK port, while the system clock is connected to the SCLK port. This primitive can be used on both A/B or C/D pairs of I/O cells at the bottom side of the device.

Figure 11-53. IDDRX2E Symbol

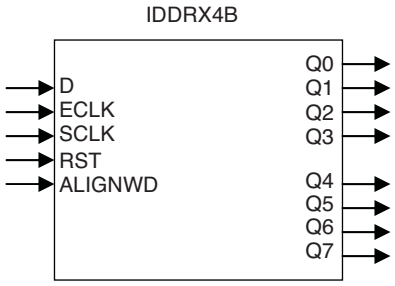


The internal register structure for this primitive is based on the video PIO cell, as shown in receive path of Figure 11-3. The first set of the registers is the DDR register to capture the data at both edges of the ECLK. The second set is the synchronization registers to hold the data ready for clock domain transfer. The third set of registers performs the clock domain transfer from ECLK to SCLK.

IDDRX4B

The primitive implements the input register block in x4 gearing mode. This is used only for the generic DDR x4 interface (gearing ratio 1:8) on the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices. Its registers are designed to use edge clock routing on the GDDR interface and the system clock on the FPGA side. The edge clock is connected to the ECLK port, while the system clock is connected to the SCLK port. This primitive can only be used on the A/B pair of the I/O cells at the bottom side of the device.

Figure 11-54. IDDRX4B Symbol

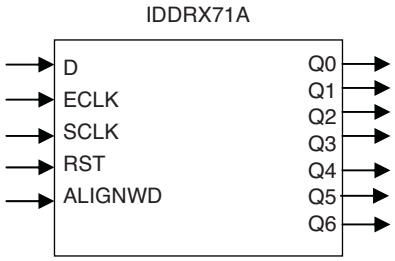


The 1:8 gearing of IDDRX4B uses two of the 1:4 gearing and shares the basic architecture with IDDRX2E. The internal register structure for this primitive is based on the video PIO cell, as shown in receive path of figure 1c. The first set of registers is the DDR register to capture the data at both edges of the ECLK. The second set of registers is synchronization registers to hold the data ready for clock domain transfer. The third set of registers performs the clock domain transfer from ECLK to SCLK.

IDDRX71A

The primitive implements the input register block in 7:1 gearing mode. This is used only for the generic DDR 71 interface (gearing ratio 1:7) on the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices. Its registers are designed to use edge clock routing on the GDDR interface and the system clock on the FPGA side. The edge clock is connected to the ECLK port, while the system clock is connected to the SCLK port. This primitive, like the input x4 gearing primitive, can only be used on the A/B pair of the PIO cell at the bottom side of the device.

Figure 11-55. IDDRX71A Symbol

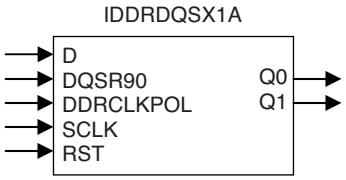


The 1:7 gearing of IDDRX71A shares the same architecture as the 1:8 gearing of the IDDRX4B primitive. It depends on an internal control signal to select three bits or four bits data at a time. The internal register structure for this primitive is based on the video PIO cell, as shown in the receive path of Figure 11-3. The first set of the registers is the DDR register to capture the data at both edges of ECLK. The second set is the synchronization registers to hold the data ready for clock domain transfer. The third set of registers performs the clock domain transfer from ECLK to SCLK

IDDRDQSX1A

This primitive is the DDR memory input buffer. It can only be used on the right side of the MachXO2-640U, MachXO2-1200/U and higher density devices for DQ or DQS input pins. This primitive comprises three stages of registers. The first stage of DDR register captures the incoming data at the rising and falling edges of the DQSR90 signal, which is the 90° shifted DQS strobe, generated by the DQS circuitry (DQSBUF). The captured data is then transferred to the second stage of registers with the rising edge SCLK or falling edge SCLK depending on the polarity of the DDRPOLCLK signal, which also is generated from DQSBUF to guarantee the clock domain crossing. The final stage of registers is re-clocked by the rising edge of SCLK to provide the full clock cycle transition to the core.

Figure 11-56. IDDRXQSX1A Symbol



The internal register structure for this primitive is based on the input path of the memory PIO cell, as shown in Figure 11-2.

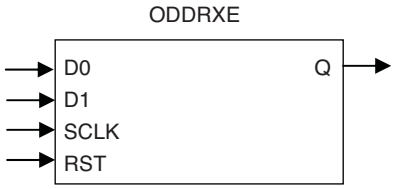
Output DDR Primitives

The output DDR primitives represent the output DDR module used to multiplex two data streams before sending them out to the GDDR interface or to the DDR memory device. There are several modes for the DDR output registers to implement different gearings for GDDR interfaces. The memory DDR output uses a different primitive than the GDDR interfaces. For all the data ports of the output primitives, D0 of the parallel data is the first bit transmitted.

ODDRXE

This primitive will implement the output register block in x1 gearing mode. It can be used in all sides of the MachXO2 devices. A single primary clock source, SCLK from the FPGA core, is used for this primitive.

Figure 11-57. ODDRXE Symbol

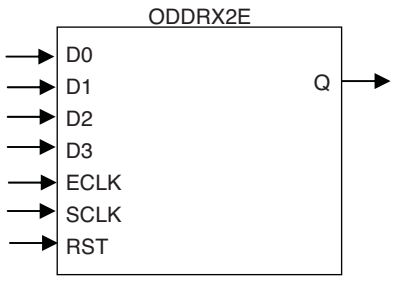


The internal register structure for this primitive is based on the basic PIO cell, as shown in Figure 11-1. The SCLK is used to multiplex between the 2-bit parallel data to generate a serial data stream.

ODDRX2E

The primitive implements the output register block in x2 gearing mode. This is used only for the generic DDR x2 interface (gearing ratio 4:1) on the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices. Its registers are designed to use the system clock on the FPGA side and the edge clock at the DDR interface. The edge clock is connected to ECLK port, while the system clock is connected to SCLK port. This primitive can be used on both the A/B or C/D pairs of PIO cells at the top side of the device.

Figure 11-58. ODDRX2E Symbol

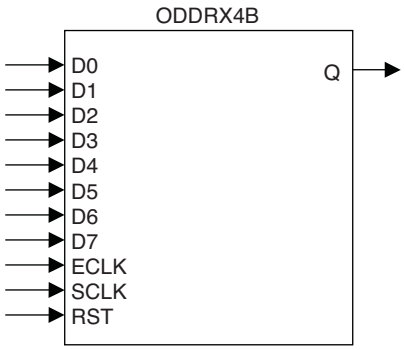


The internal register structure for this primitive is based on the video PIO cell, as shown in transmit path of Figure 11-3. The parallel data is registered by SCLK at the first set of registers. At each update, the parallel data is clocked in by SCLK and is held by the second set of registers. The third set of registers has the data ready to be multiplexed out as serial data by the ECLK.

ODDRX4B

The primitive implements the output register block in x4 gearing mode. This is used only for the generic DDR x4 interface (gearing ratio 8:1) on the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices. Its registers are designed to use the system clock on the FPGA side and the edge clock at the GDDR interface. The edge clock is connected to the ECLK port, while the system clock is connected to the SCLK port. This primitive can only be used on the A/B pair of I/O cells at the top side of the device.

Figure 11-59. ODDRX4B Symbol

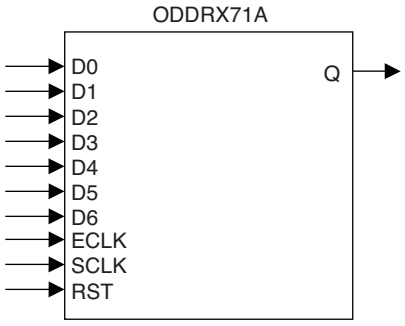


The internal register structure for this primitive is based on the video PIO cell, as shown in the transmit path of Figure 11-3. The parallel data is registered by SCLK at the first set of registers. At each update, the parallel data is clocked in by SCLK and is held by the second set of registers. The third set of registers has the data ready to be multiplexed out as serial data by ECLK.

ODDRX71A

The primitive implements the output register block in 7:1 gearing mode. This is used only for the generic DDR 71 interface (gearing ratio 7:1) on the top side of the MachXO2-640U, MachXO2-1200/U and higher density devices. Its registers are designed to use the system clock on the FPGA side and the edge clock routing on the GDDR interface. The edge clock is connected to the ECLK port, while the system clock is connected to the SCLK port. This primitive can only be used on the A/B pair of I/O cells at the top side of the device.

Figure 11-60. ODDRX71A Symbol



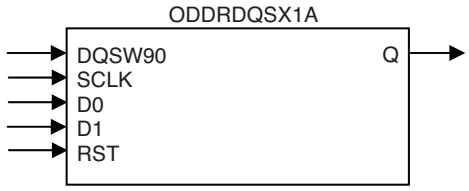
The 7:1 gearing of ODDRX71A shares the same architecture as the 8:1 gearing of the ODDRX4B primitive. It depends on the internal control signal to select the three or four bits of data at a time for transmission. The internal register structure for this primitive is based on the video PIO cell, as shown in the transmit path of Figure 11-3. The parallel data is registered by SCLK at the first set of registers. At each update, the parallel data is clocked in by SCLK and is held by the second set of registers. The third set of registers has the data ready to be multiplexed out as serial data by ECLK.

ODDRDQSX1A

This primitive is the DDR memory output buffer. It can only be used on the right side of the MachXO2-640U, MachXO2-1200/U and higher density devices for DQ or DQS output pins. For DQ output data, the parallel data is

clocked into this primitive using the system clock, SCLK. The captured data is multiplexed out to the pin by the DQSW90 signal, which is the 90° shift of SCLK, which is generated by the DQS circuitry (DQSBUF). For the DQS output, it uses the same output buffer architecture but the clock is controlled by the SCLK to ensure the DQS is not shifted 90° like the DQ signals. This ensures the DQS to be center aligned with the DQ for transmission.

Figure 11-61. ODDRDQSX1A Symbol

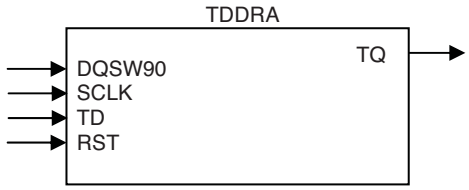


The internal register structure for this primitive is based on the output path of the memory PIO cell, as shown in Figure 11-2.

TDDRA

This is a special tri-state primitive used at the right side of the MachXO2-640U, MachXO2-1200/U and higher density devices for DQ or DQS pins. Its register structure is shown in the Tristate Register block of memory PIO cell, Figure 11-2.

Figure 11-62. TDDRA Symbol



When this component is used for DQ pins, the DQSW90 port should be driven by the DQSW90 signal from the DQSBUF component. When this component is used for DQS, the DQSW90 port should be driven by SCLK. This primitive has an attribute associated with it as listed in the table. This attribute is to ensure the DQS pin is centered at the data, DQ, during DDR memory write.

Table 11-14. TDDRA Attributes

Attribute	Description	Values	Software Default
DQSW90_INVERT	Select clock polarity for the TDDRA for DQS pin. Only used for DQS during DDR Write	ENABLED, DISABLED	DISABLED

DDR Control Logic Primitives

The DDR primitives discussed below include the DLL, the DQS circuitry, and the delay elements. The DLL is shared between GDDR and DDR memory. The DQS circuitry is only used for DDR memory. The delay elements are for data paths or the clock slave delay paths.

DQSBUFH

This primitive represents the DQS circuitry used in DDR memory interface. It generates a shifted version of DQS for DDR memory READ and DDR WRITE operations. It provides preamble and postamble detection, data valid detection, and the many functions necessary for DDR memory interface applications.

Figure 11-63. DQSBUFH Symbol

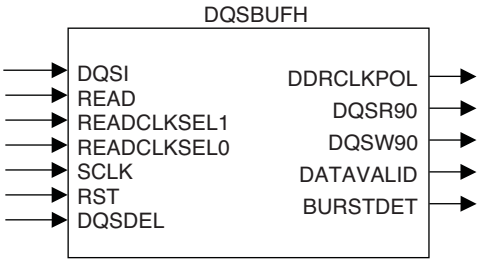
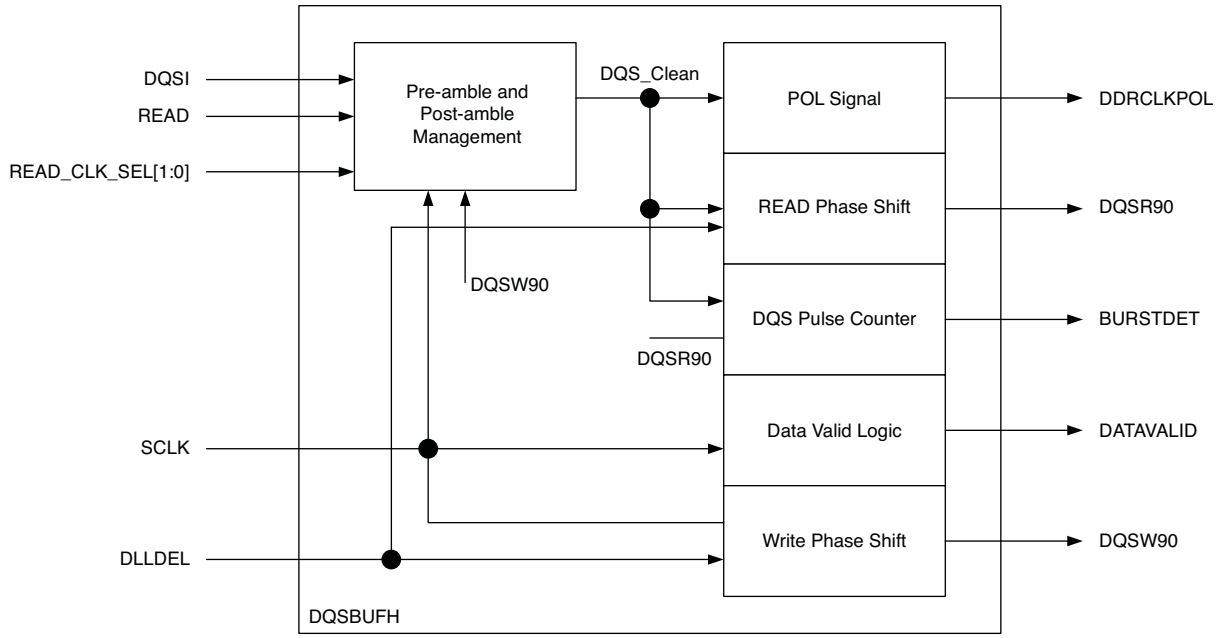


Table 11-15. DQSBUFH signals

Signal	I/O	Description
DQSI	I	DQS signal from pin
READ	I	Signal for DDR read mode, from FPGA logic
READCLKSEL1, READCLKSEL0	I	Select read clock source and polarity control for READ pulse position control in T/4 precision. The 4 positions are the rising/falling edges of SCLK or DQSW90. The signals come from FPGA logic.
SCLK	I	System clock
RST	I	RESET for this block
DQSDEL	I	DQS slave delay control from DQSDLLC
DDRCLKPOL	O	SCLK polarity control
DQSR90	O	DQS phase shifted by 90° output
DQSW90	O	SCLK phase shifted by 90° output
DATAVALID	O	Data valid signal for READ mode
BURSTDET	O	Burst detection signal

Figure 11-64 gives an overview of the functional blocks within this DQSBUFH primitive.

Figure 11-64. DQSBUFH Block Diagram



The DQS_Clean is a digitally-generated DQS signal which is a glitch-less version of the DQSI for downstream logic. The READCLKSEL signal is driven by the user logic and is part of the DDR memory controller IP. It selects one of the four phase options between the SCLK and the signal DQSW90 to start the READ process.

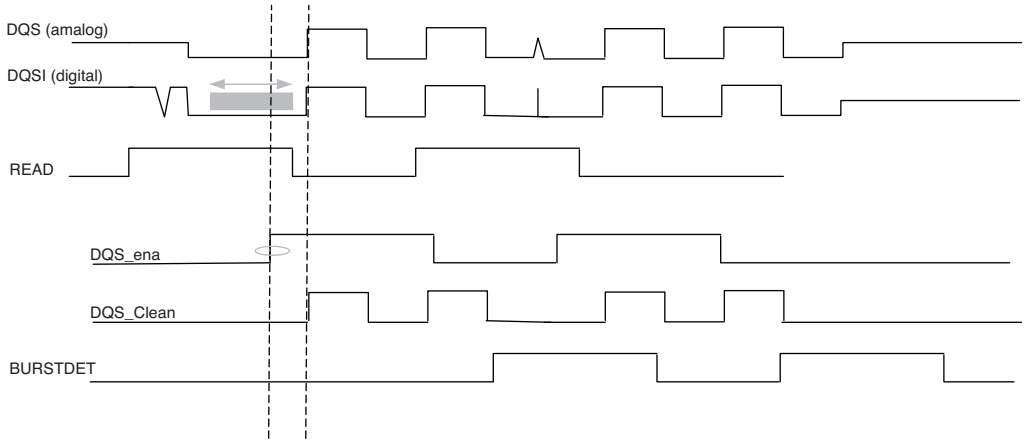
The DDR Clock Polarity (DDRCLKPOL) signal is generated based on the phase of SCLK at the first DQS transition. If the DQS transition happens when the SCLK is high, then DDRCLKPOL is high and the rising edge of SCLK is used to clock the data into the FPGA core. Otherwise, DDRCLKPOL is low and the falling edge of SCLK is used to clock the data into the FPGA core.

The DQSR90 is the 90° phase shift of the DQS_Clean signal based on the DQSDEL delay control code from the DQSDLL. It is distributed to the DQSR90 tree and is used by all the DDR memory incoming data, DQ, to clock the incoming data during the READ operation. The DQSW90 is the 90° phase shift of the SCLK. It is used as a select signal to serialize the outgoing data during the WRITE operation. The DQS buffer during WRITE operation has a 90° phase difference from the data, DQ.

BURSTDET is used during DDR memory READ operations for read pulse positioning optimization at the start of a READ process. A minimum burst length (BL) of 4 must be used in the training process at start up. This can be done with two consecutive BL2 (BL=2), even number of BL2, or any multiplication of BL4 (BL=4) or longer burst lengths. The BURSTDET signal is also used to perform periodic read position calibration during the READ operation. The DDR memory controller IP monitors the BURSTDET signal with necessary steps to ensure the optimized read pulse position is found.

The READ signal to the DQSBUF block is generated by the DDR memory controller IP. The READ signal will go high after the READ command to control the DDR-SDRAM. This should normally precede the DQS preamble by one cycle, but may overlap the trailing bits of a prior read cycle. The READ signal is used to activate an internal signal, DQS_ena within the shaded region of Figure 11-65. If the DQS_ena is activated within this region, BURSTDET will go high and the DQS_Clean signal will be turned on. The READ signal has to last as long as the burst length.

Figure 11-65. READ Pulse Positioning Optimization



DQSDLLC

The DQSDLLC is the on-chip DLL, which generates the 90° phase shift required for the DQS signal. Only one DQSDLLC can be used for the DDR implementations on one-half of the device. The clock input to this DLL should be at the same frequency as the DDR interface. The DQSDLLC generates the delay based on this clock frequency and the update control input to this block. The DQSDLLC updates the dynamic delay control code (DQSDEL) to the DQS delay block when this update control (UDDCNTLN) input is asserted. Otherwise, the update will be in the hold condition. The active low signal on UDDCNTLN updates the DQS phase alignment.

Figure 11-66. DQSDLLC Symbol

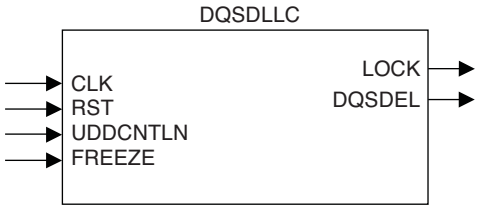


Table 11-16. DQSDLLC Signals

Signal	I/O	Description
CLK	I	Input clock to the DLL, same frequency as DDR interface
RST	I	DLL reset control
UDDCNTLN	I	Update/hold control to delay code before adjustment. Active low signal updates the delay code.
FREEZE	I	Use to freeze or release DLL input CLK
LOCK	O	DLL lock signal
DQSDEL	O	DLL delay control code to slave delay cells, connect to DQSDEL of the DQSBUFH element

The DQS delay can be updated for PVT variation using the UDDCNTLN input. The DQSDEL is updated when the UDDCNTLN is held low. The DQSDEL can be updated when variations are expected. It can be updated anytime except during a DDR memory READ or WRITE operation.

The FREEZE input port of this component is used to freeze or release the DLL. When FREEZE goes high, the device will freeze the DLL to save power while the delay code is preserved. When FREEZE goes low, it will release the DLL to resume operation. FREEZE must be applied to the DQSDLLC before the clock stops.

By default, this DLL will generate a 90° phase shift for the DQS strobe based on the frequency of the input reference clock to the DLL. The user can control the sensitivity to jitter by using the LOCK_SENSITIVITY attribute. This configuration bit can be programmed to be either HIGH or LOW. Lock_sensitivity HIGH means more sensitive to jitter. It is recommended that the bit be programmed LOW.

The DQSDLLC supports a wide range of frequencies up to 400 MHz. The FIN attribute associated with this primitive allows the user to set the DLL frequency. It is possible to bypass the DLL locking process when the frequency becomes very low. The attribute FORCE_MAX_DELAY can be used for this purpose. When FORCE_MAX_DELAY is set in the software, the DLL will not go through the locking process. Instead, DLL will be locked to maximum delay steps. The effect of the FORCE_MAX_DELAY attribute will not be reflected in the simulation model. The simulation model always models the 90° phase shift for DLL. Refer to the MachXO2 Family Data Sheet for the range of frequencies when the FORCE_MAX_DELAY becomes effective.

Table 11-17. Attribute for DQSDLLC

Attribute	Description	Values	Software Default
LOCK_SENSITIVITY	Jitter sensitivity	HIGH, LOW	LOW
FIN	Input clock frequency of DLL	Range supported by DLL	100 MHz
FORCE_MAX_DELAY	Bypass DLL locking procedure at low frequency and sets the maximum delay setting	YES, NO	NO

DELAYE

Input data going to the DDR registers can optionally be delayed using the delay block, DELAYE. The 32-tap DELAYE block is used to compensate for clock injection delay times. The amount of the delay is determined by the software based on the type of interface implemented using the attribute DEL_MODE. Users are allowed to set the delay by choosing the USER_DEFINED mode for the block. When in USER_DEFINED mode, user must manually set the number of delay steps to be used. Each delay stay would generate ~105ps of delay. It is recommended to

use the PREDEFINED mode for all generic DDR interfaces. If an incorrect attribute value is used for a given interface, the DELAYE setting will be incorrect and the performance of the DDR interface will not be optimal. The DELAYE block is applicable to the receive mode of the DDR interfaces. It is available for all input register paths at all sides of a MachXO2 device.

Figure 11-67. DELAYE Symbol

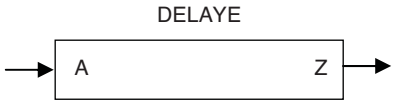


Table 11-18. DELAYE Signals

Signal	I/O	Description
A	I	DDR input from sysIO buffer
Z	O	Output with delay

Table 11-19. DELAYE attributes

Attribute	Description	Value	Software Default
DEL_MODE	Fixed delay value depending on interface and user-defined delay values	SCLK_ZEROHOLD ECLK_ALIGNED ECLK_CENTERED SCLK_ALIGNED SCLK_CENTERED USER_DEFINED	USER_DEFINED
DEL_VALUE	User-defined value	DELAY0…DELAY31	DELAY0

Table 11-20. DEL_MODE Values Corresponding to the GDDR Interface

Interfaces Name	DEL_MODE values
GIREG_RX.SCLK	SCLK_ZERHOLD
GDDRX1_RX.SCLK.Aligned	SCLK_ALIGNED
GDDRX1_RX.SCLK.Centered	SCLK_CENTERED
GDDRX2_RX.ECLK.Aligned	ECLK_ALIGNED
GDDRX2_RX.ECLK.Centered	ECLK_CENTERED
GDDRX4_RX.ECLK.Aligned	ECLK_ALIGNED
GDDRX4_RX.ECLK.Centered	ECLK_CENTERED
GDDRX71_RX.ECLK.71	Bypass
GOREG_TX.SCLK	N/A
GDDRX1_TX.SCLK.Centered	N/A
GDDRX1_TX.SCLK.Aligned	N/A
GDDRX2_TX.ECLK.Aligned	N/A
GDDRX2_TX.ECLK.Centered	N/A
GDDRX4_TX.ECLK.ALIGNED	N/A
GDDRX4_TX.ECLK.CENTERED	N/A
GDDRX_TX.ECLK.7:1	N/A

DELAYD

At the bottom side of the MachXO2-640U, MachXO2-1200/U and higher density devices, input data going to the DDR registers can also be delayed by the DELAYD block. Unlike the DELAYE block where the delay is determined during the operation of the device, the DELAYD block allows user to control the amount of data delay while the device is in operation. This block receives 5-bit (32 taps) delay control. The 5-bit delay is dynamically controlled by

the user logic through the delay port. Each delay step would generate ~105ps of delay.

Figure 11-68. DELAYD Symbol

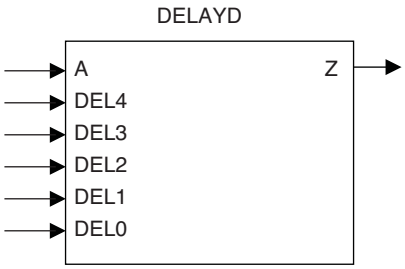


Table 11-21. DELAYD signals

Signal	I/O	Description
A	I	Data input from I/O buffer
DEL4, DEL3, DEL2, DEL1, DEL0	I	Dynamic delay input port from FPGA logic
Z	O	Output with delay

DLLDELC

This is the clock slave delay cell, which is used to generate a 90° delay in all receive aligned interfaces. The 90° delay is calculated based on the input clock to the DQSDLLC element. The amount of delay required is based on the delay control code, DQSDEL, generated from the DQSDLLC.

Figure 11-69. DLLDELC Symbol

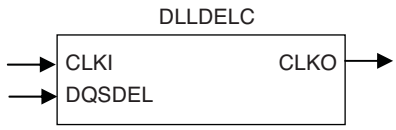


Table 11-22. DLLDELC Signals

Signal	I/O	Description
CLKI	I	Data Input from I/O buffer
DQSDEL	I	Dynamic delay inputs from DQSDLLC
CLKO	O	Output with delay

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
January 2011	01.1	Updated for ultra-high I/O (“U”) devices.
April 2011	01.2	Added signal names of IPexpress modules and timing analysis information.
July 2011	01.3	Updated Generic High-Speed I/O DDR Interfaces table with footnote about migration from MachXO2-1200-R1 to Standard (non-R1) devices
February 2012	01.4	Updated document with new corporate logo.
		Document status changed from Preliminary to Final.
April 2013	01.5	Updated the Attribute for DQSDLLC table.
		Updated delay information for DELAYE and DELAYD blocks.

Memory Usage Guide for MachXO2 Devices

February 2012 Technical Note TN1201

Introduction

This technical note discusses the memory usage for the Lattice MachXO2™ PLD family. It is intended to be used by design engineers as a guide in integrating the EBR and PFU based memories for these devices in ispLEVER®.

The architecture of these devices provides resources for memory intensive applications. The sysMEM™ Embedded Block RAM (EBR) complements the distributed PFU-based memory. Single-Port RAM, Dual-Port RAM, Pseudo Dual-Port RAM, FIFO and ROM memories can be constructed using the EBR. LUTs and PFU can implement Distributed Single-Port RAM, Dual-Port RAM and ROM.

The capabilities of the EBR Block RAM and PFU RAM are referred to as primitives and are described later in this document. Designers can utilize the memory primitives in two ways:

- Via **IPexpress™** – The IPexpress GUI allows users to specify the memory type and size that is required. IPexpress takes this specification and constructs a netlist to implement the desired memory by using one or more of the memory primitives.
- Via the **PMI (Parameterizable Module Instantiation)** – PMI allows experienced users to skip the graphical interface and utilize the configurable memory modules on the fly from the ispLEVER Project Navigator. The parameters and the control signals needed either in Verilog or VHDL can be set. The top-level design will have the parameters defined and signals declared so the interface can automatically generate the black box during synthesis.

In addition to familiar Block RAM and PFU RAM primatives, MachXO2-640 and higher density devices provide a new User Flash Memory (UFM) block, which can be used for a variety of applications including storing a portion of the configuration image, storing and initializing EBR data, storing PROM data or as a general purpose non-volatile user Flash memory. The UFM block connects to the device core through the embedded function block WISHBONE interface. Designers can also access the UFM block through the JTAG, I^2C and SPI interfaces of the device. The UFM block offers the following features:

- Non-volatile storage up to 256Kbits
- Byte addressable for read access. Write access is performed in 128-byte pages.
- Program, erase, and busy signals
- Auto-increment addressing
- WISHBONE interface
- External access is provided through JTAG, I^2C and SPI interfaces

For more information on the UFM, please refer to TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

The remainder of this document discusses these approaches, utilizing IPexpress, PMI inference, memory modules and memory primitives.

Memories in MachXO2 Devices

All MachXO2 devices contain an array of logic blocks called PFUs surrounded by Programmable I/O Cells (PICs). In addition, all but the smallest MachXO2 device (MachXO2-256) contain sysMEM EBR blocks. This is shown in Figures 12-1 and 12-2.

Figure 12-1. Top View of the MachXO2-1200 Device

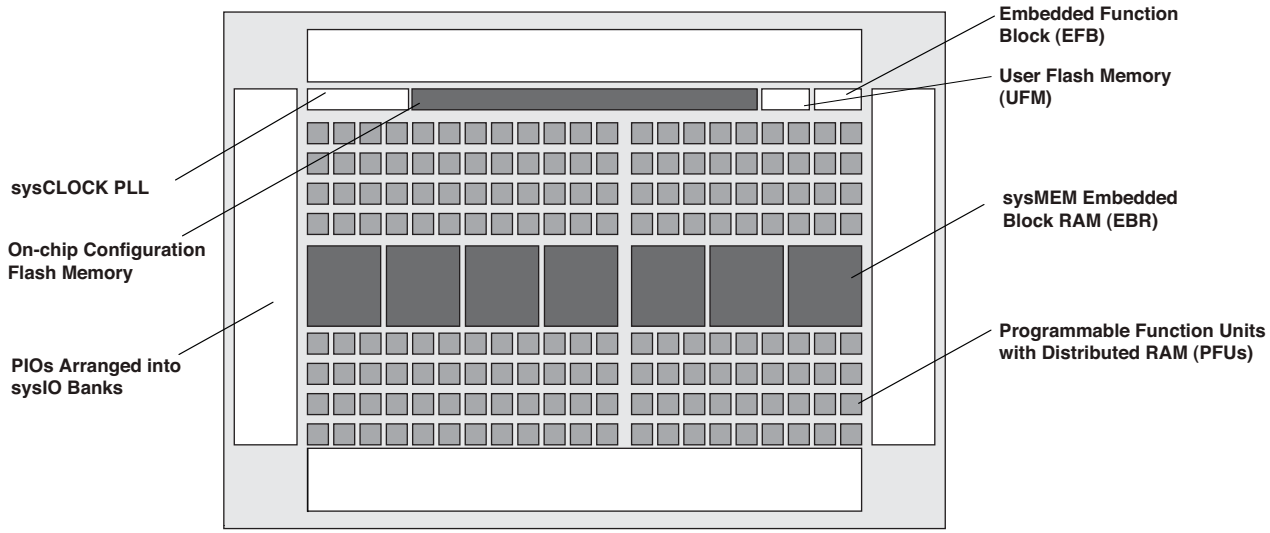
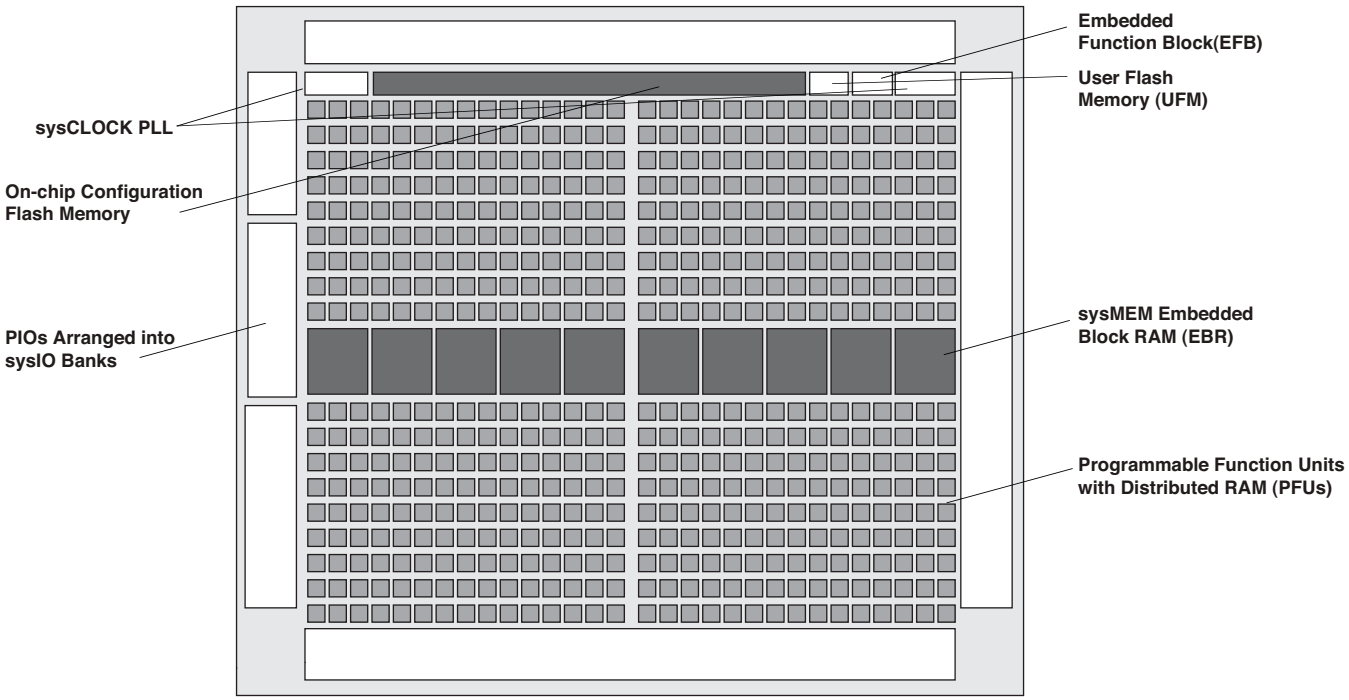


Figure 12-2. Top View of the MachXO2-4000 Device



The PFU contains the building blocks for logic and Distributed RAM and ROM. Some PFUs provide the logic building blocks without the distributed RAM. This document describes the memory usage and implementation for both Embedded Memory Blocks (EBRs) and Distributed RAM of the PFU. Refer to the MachXO2 Family Data Sheet for details on the hardware implementation of the EBR and Distributed RAM.

Utilizing IPexpress

Designers can utilize IPexpress to easily specify a variety of memories in their designs. These modules will be constructed using one or more memory primitives along with general purpose routing and LUTs as required. The available primitives are:

- Single Port RAM (RAM_DQ) – EBR based
- Dual Port RAM (RAM_DP_TRUE) – EBR based
- Pseudo Dual Port RAM (RAM_DP) – EBR based
- Read Only Memory (ROM) – EBR based
- First In First Out Memory (FIFO_DC) – EBR based
- Distributed Single Port RAM (Distributed_SPRAM) – PFU based
- Distributed Dual Port RAM (Distributed_DPRAM) – PFU based
- Distributed ROM (Distributed_ROM) – PFU based
- RAM-based Shift Register - PFU based

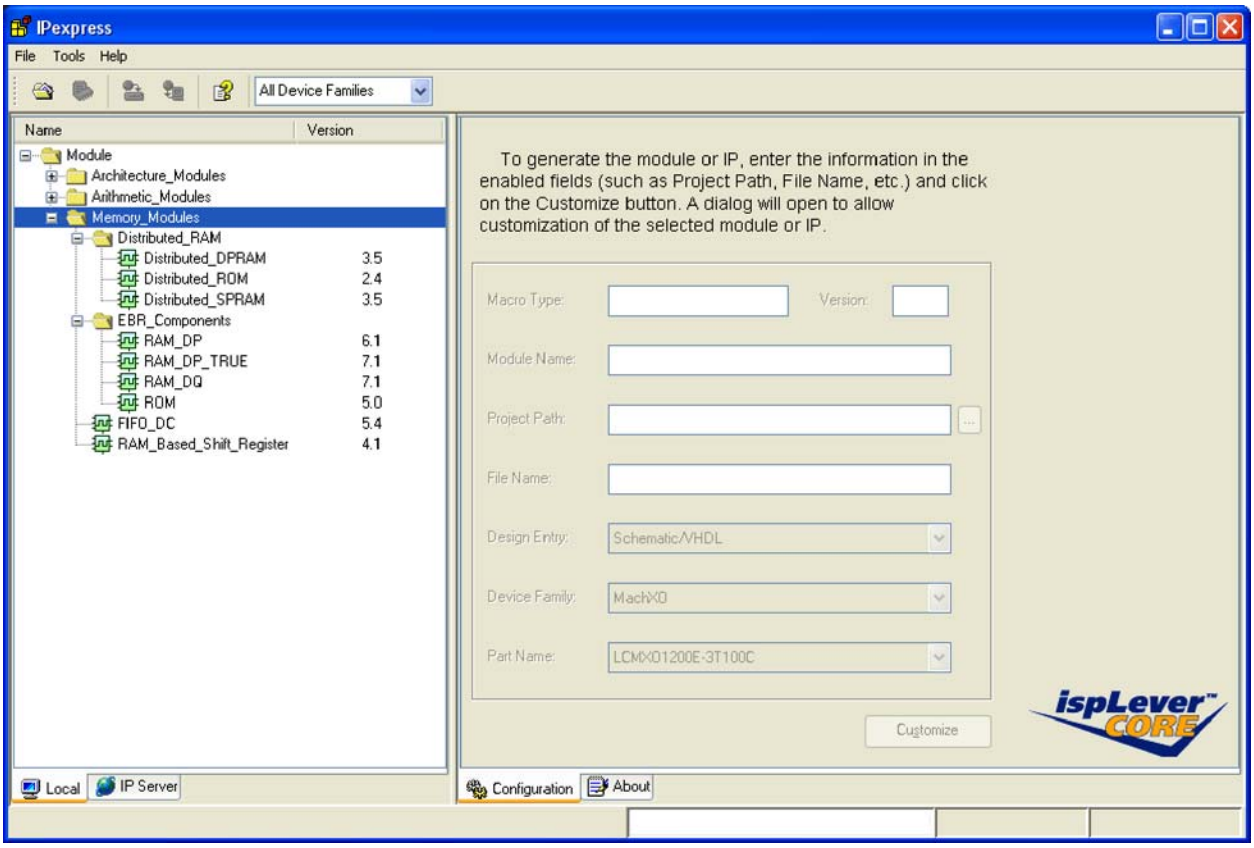
IPexpress Flow

For generating any of these memories, create (or open) a project for the MachXO2 devices.

From the Project Navigator, select **Tools > IPexpress**. Alternatively, users can also click on the IPexpress button in the toolbar when MachXO2 devices are targeted in the project.

This opens the IPexpress window as shown in Figure 12-3.

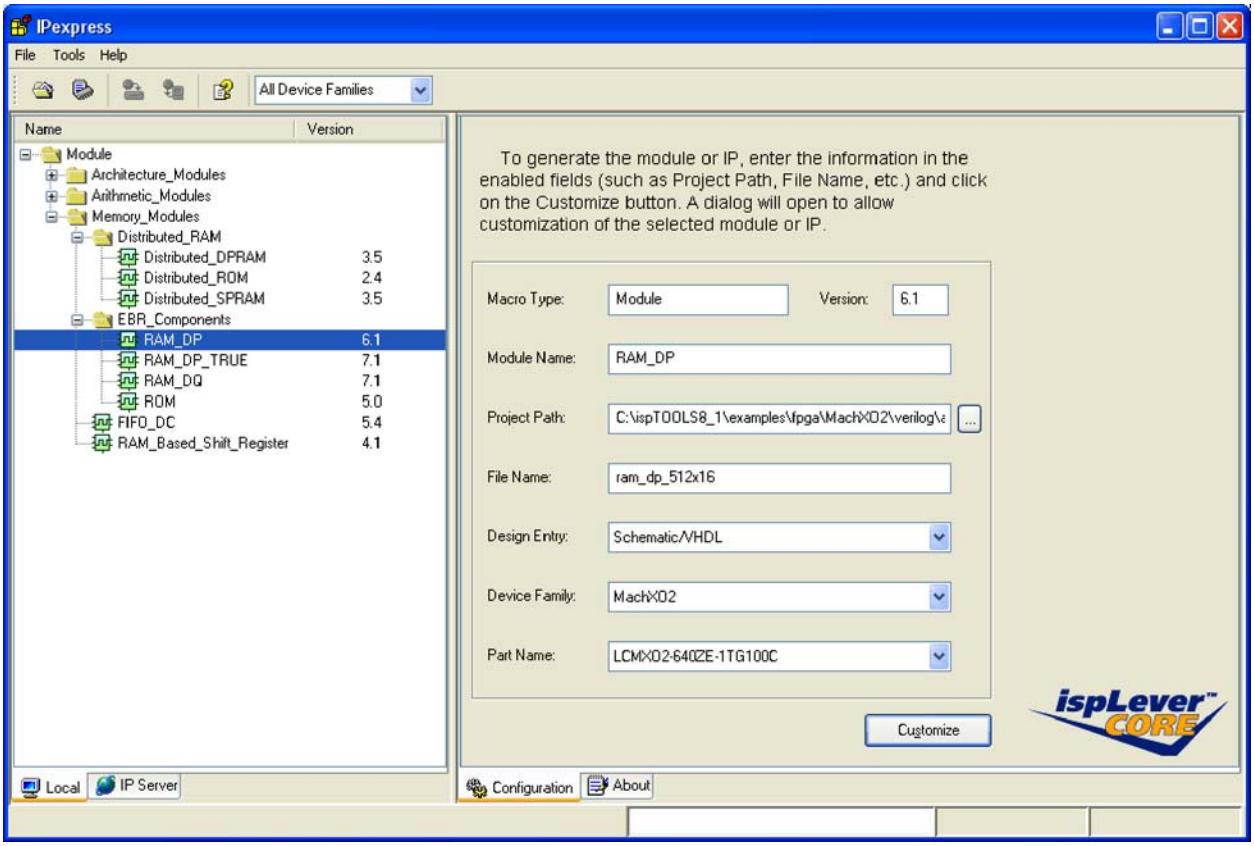
Figure 12-3. IPexpress – Main Window



The left pane of this window has the Module Tree. The EBR-based Memory Modules are under the **EBR_Components** and the PFU-based Distributed Memory Modules are under **Distributed_RAM** as shown in Figure 12-3.

As an example, let us consider generating an EBR-based Pseudo Dual Port RAM of size 512x16. Select RAM_DP under the EBR_Components. The right pane changes as shown in Figure 12-4.

Figure 12-4. Example Generating Pseudo Dual Port RAM (RAM_DP) Using IPexpress



In this right pane, options like the **Macro Type** and **Module_Name** are device and selected module dependent. These cannot be changed in IPexpress.

Users can change the directory where the generated modules will be placed by clicking the **Browse** button in the **Project Path**.

The **File Name** text box allows users to specify an entity name for the module they are about to generate. Users must provide this entity name.

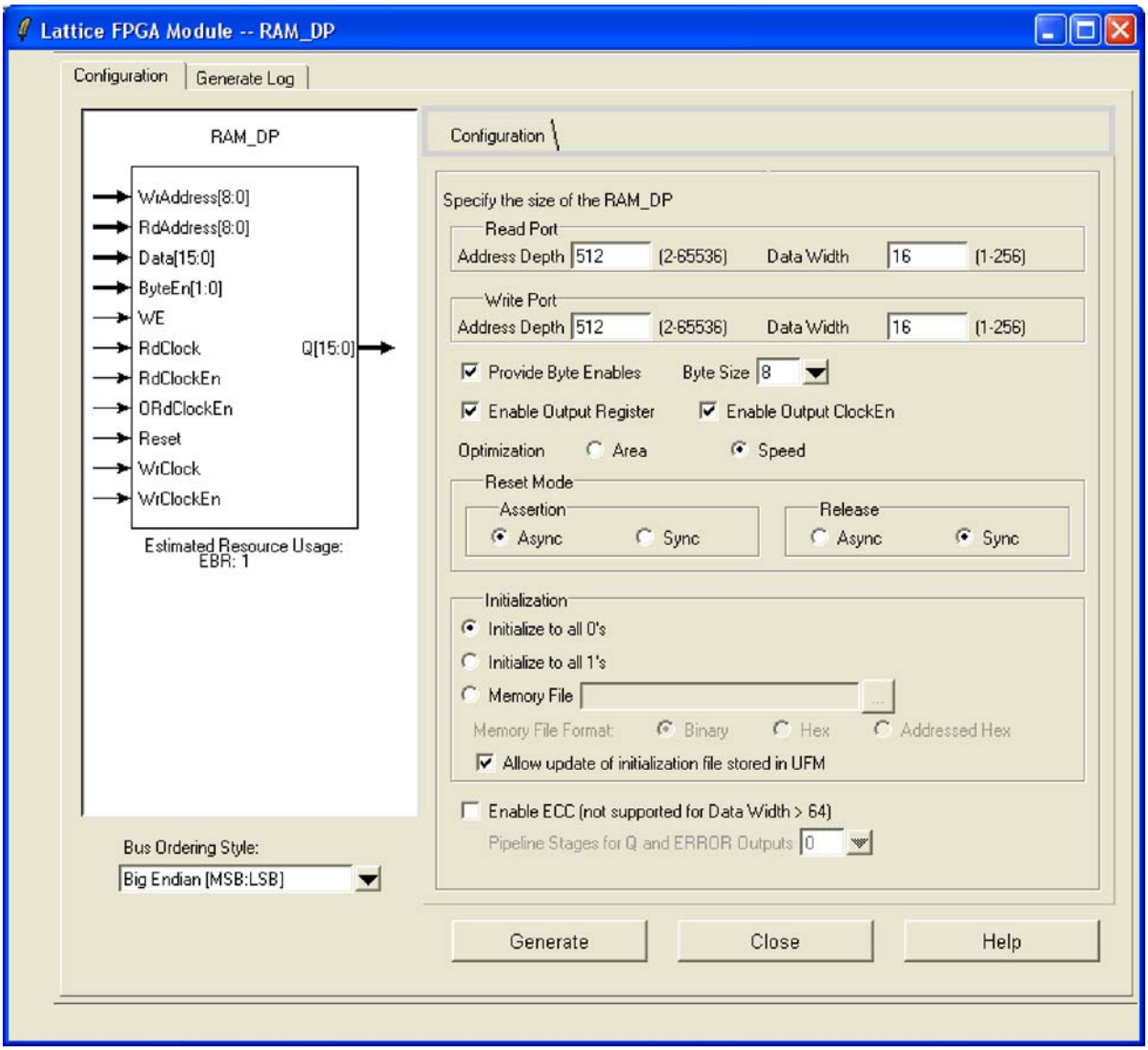
Design entry, Verilog or VHDL, by default, is the same as the project type. If the project is a VHDL project, the selected design entry option will be “VHDL”, and “Verilog-HDL” if the project type is Verilog-HDL. Schematic support may also be selected with either HDL type.

When launched from within Project Navigator, the **Device Family** and **Part Name** pull-down menus are filled in by default and cannot be changed by the user. However, when IPexpress is launched as a stand-alone application, these menus allow users to select different devices within a device family, MachXO2 in this example.

When finished, click the **Customize** button.

This opens another window where users can customize the RAM (Figure 12-5).

Figure 12-5. Example Generating Pseudo Dual Port RAM (RAM_DP) Module Customization



The left side of this window shows the block diagram of the module. The right side includes the Configuration tab where users can choose options to customize the RAM_DP such as (e.g. specify the address port sizes and data widths).

Users can specify the address depth and data width for the **Read Port** and the **Write Port** in the text boxes provided. In this example we are generating a Pseudo Dual Port RAM of size 512 x 16. Users can also create RAMs of different port widths for Pseudo Dual Port and True Dual Port RAMs.

The Input Data and the Address Control is always registered, as the hardware only supports the clocked write operation for the EBR-based RAMs. The check box **Enable Output Register** inserts the output registers in the Read Data Port, as the output registers are optional for EBR-based RAMs.

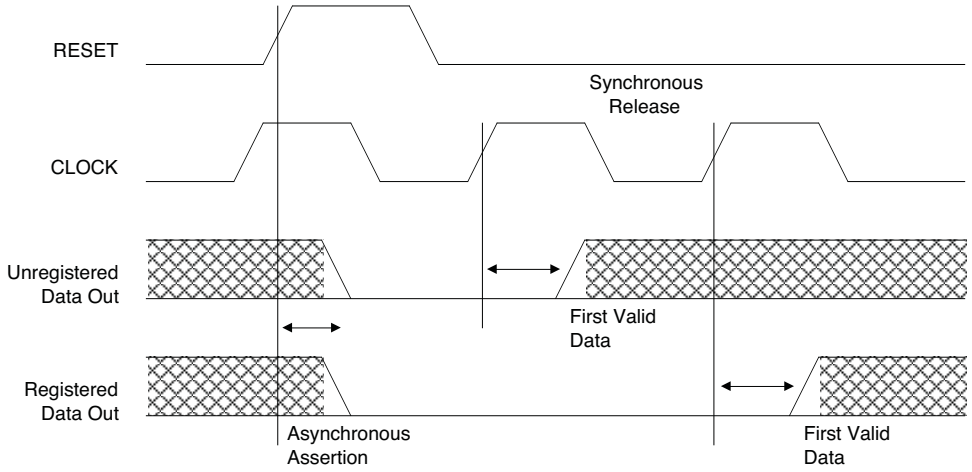
Clock Enable control is always provided for Input Data and Address signals. When Output Registers are enabled, separate **Output Clock Enables** can be selected.

Users can specify the use of **Byte Enables**. Byte Enables can be used to mask the input data so that only specific bytes of memory are overwritten. The unwritten bytes retain the previously written data.

The **Reset Mode** of the memory can be specified by the user for both assertion and release. For the **synchronous reset**, the clock should be there and reset signal should satisfy setup/hold time requirements for both asserting and deasserting edges. The write function is automatically disabled during a synchronous reset because the CS registers are reset, but the write is not disabled during an asynchronous reset operation.

The asynchronous reset can be programmed to be released (de-asserted) synchronously. As shown in Figure 12-6, when de-asserted synchronously, the first clock edge after reset will release the internal reset to all registers that have asynchronous reset, i.e. the data output registers, the FIFO counters and the FIFO flag registers.

Figure 12-6. Asynchronous Reset with Synchronous Release



Memory may be initialized at configuration to all 1's or all 0's. To maximize the number of UFM bits, initialize the EBRs to an all 0's pattern. Initializing to an all 0's pattern does not use up UFM bits. Users can also initialize their memory with the contents specified in the **Memory File**. It is optional to provide this file for RAM; however for ROM, the **Memory File** is required. These files can be of Binary, Hex or Addressed Hex format. The details of these formats are discussed in the Initialization File section of this document.

Traditionally, the initialization **Memory File** is static and is stored in the device configuration bitstream. Alternatively, the MachXO2 architecture allows the memory initialization data to be stored in UFM where it may be accessed and/or dynamically modified by the user. To enable this feature, select **Allow Update of initialization file stored in UFM**. For details of this feature, please refer to TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

At this point, users can click the **Generate** button to generate the module they have customized. A VHDL or Verilog netlist is then generated and placed in the specified location. Users can incorporate this netlist in their designs. In addition, an instantiation template file (*_tmpl.v or .vhd), a Lattice Parameter file (*.lpc), a testbench template file (tb_*_tmpl.v or .vhd), and two log files (*_generate.log, *.srp) are generated. Finally, a schematic symbol file (*.sym) is created if Design Entry type is Schematic/VHDL or Schematic/Verilog.

Once the module is generated, user can either instantiate the *.lpc or the Verilog-HDL/ VHDL file in top-level module of their design.

ECC in Memory Modules

IPexpress allows users to implement Error Check Codes in the EBR-based memory modules. There is a checkbox to enable ECC in the configuration tab for the module.

If you choose to use ECC, you will have a 2-bit error signal and the error codes are as below:

- Error[1:0] = "00" – Indicates there is no error.
- Error[1:0] = "01" – Indicates there was a 1-bit error which was fixed.

- Error[1:0] = “10” – Indicates there was a 2-bit error which cannot be corrected.
- Error[1:0] = “11” – Not used.

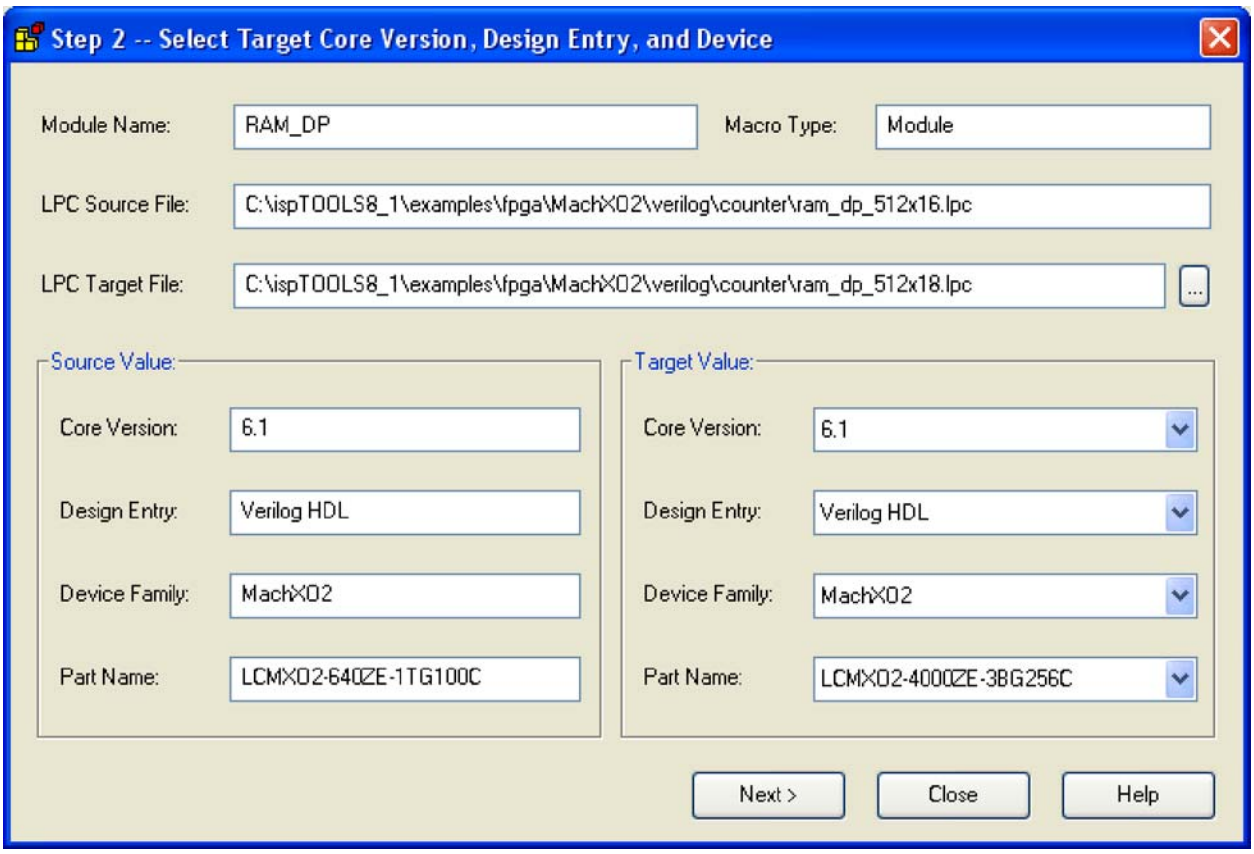
IP Regeneration/Modification

Sometimes it is useful to regenerate or modify a previously generated module. By regenerating a customized module or IP you can modify any of its settings including: device type, design entry method, and any of the options specific to the module. You can also update older modules or IP to the latest version. From the IPexpress main window, choose **Tools > Regenerate IP/Module**.

In the **Select a Parameter File** dialog box, choose the Lattice Parameter Configuration (.lpc) file of the module or IP you wish to regenerate, and click **Open**.

This opens a dialog box as shown in Figure 12-7.

Figure 12-7. Example Regenerating/Modifying IP



The **Select Target Core Version, Design Entry, and Device** dialog box shows the current settings for the module or IP in the **Source Value** box. Make your new settings in the **Target Value** box.

If you want to generate a new set of files in a new location, set the location in the LPC **Target File** box. The base of the .lpc file name will be the base of all the new file names. The LPC **Target File** must end with a .lpc extension.

Click **Next**, and proceed with module customization as before.

The various memory modules, both EBR and Distributed, are discussed in detail in the following sections.

Utilizing PMI

Parameterizable Module Instantiation (PMI) allows experienced users to skip the graphical interface and utilize the configurable memory modules on-the-fly from the ispLEVER Project Navigator.

The necessary parameters and control signals can be set in either Verilog or VHDL. The top-level design includes the defined memory parameters and declared signals. The interface can then automatically generate the black box during synthesis and ispLEVER can generate the netlist on-the-fly. Lattice memories are the same as industry standard memories, so you can get the parameters for each module from any memory-related guide, which is available through the on-line help system.

PMI modules are instantiated the same way other modules are in your HDL. The process is similar to the process for IPexpress with the addition of setting parameters to customize the module. The ispLEVER software provides a template for the Verilog or VHDL instantiation command that specifies the customized module's ports and parameters. Refer to the ispLEVER online help section "Instantiating a PMI Module" for further information.

Memory Module Inference

Finally, memories may be instantiated within Verilog or VHDL modules through inference. The HDL constructs for memory inferencing is synthesis vendor dependant. Refer to the documentation provided by the synthesis engine vendor for correct inference constructs and attribute settings.

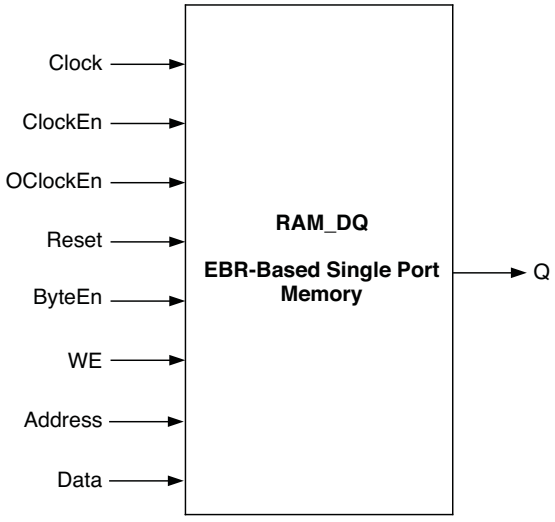
IPexpress Memory Modules

Single Port RAM (RAM_DQ) – EBR Based

The EBR blocks in the MachXO2 devices can be configured as Single Port RAM (RAM_DQ). IPexpress allows users to generate the Verilog-HDL or VHDL netlist for the memory size, as per design requirements.

IPexpress generates the memory module as shown in Figure 12-8.

Figure 12-8. Single Port Memory Module Generated by IPexpress



Since the device has a number of EBR blocks, the generated module makes use of these EBR blocks, or primitives, and cascades them to create the memory sizes specified by the user in the IPexpress GUI. For memory sizes smaller than an EBR block, the module will be created in one EBR block. For memory sizes larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

In Single Port RAM mode, the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for the Single Port Memory are listed in Table 12-1.

Table12-1. EBR-Based Single Port Memory Port Definitions

Port Name in the Generated Module	Description	Active State
Clock	Clock	Rising Clock Edge
ClockEn[1]	Clock Enable	Active High
*OClockEn[2]	Output Clock Enable	Active High
Reset[3]	Reset	Active High
*ByteEn[4]	Byte Enable	Active High
WE	Write Enable	Active High
Address	Address Bus	—
Data	Data In	—
Q	Data Out	—
*ERROR	Error Check Code	Active High

*Denotes optional port

1. **ClockEn** is used as clock enable for all the input registers.
2. **OClockEn** can be used as clock enable for the optional output registers. This allows the full pipeline of data to be output, including the last word.
3. **Reset** resets only the optional output registers of the RAM. It does not reset the input registers or the contents of memory.
4. **ByteEn** can be used to mask the input data so that only specific bytes of memory are overwritten.

The Single Port RAM (RAM_DQ) can be configured in **NORMAL**, **READ BEFORE WRITE** or **WRITE THROUGH** modes. Each of these modes affects what data comes out of the port Q of the memory during the write operation followed by the read operation at the same memory location.

IPexpress implements the MachXO2 Single Port RAM (RAM_DQ) using an appropriately configured DP8KC primitive.

Figures 12-9 through 12-14 show the internal timing waveforms for the Single Port RAM (RAM_DQ).

Figure 12-9. Single Port RAM Timing Waveform – NORMAL Mode, Without Output Registers

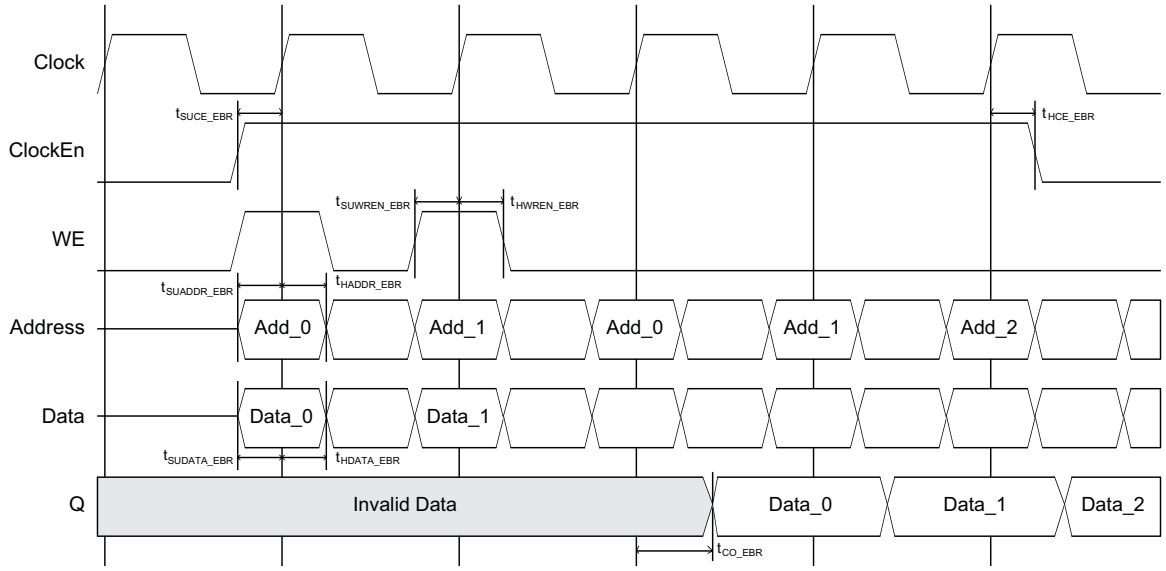


Figure 12-10. Single Port RAM Timing Waveform – NORMAL Mode, With Output Registers

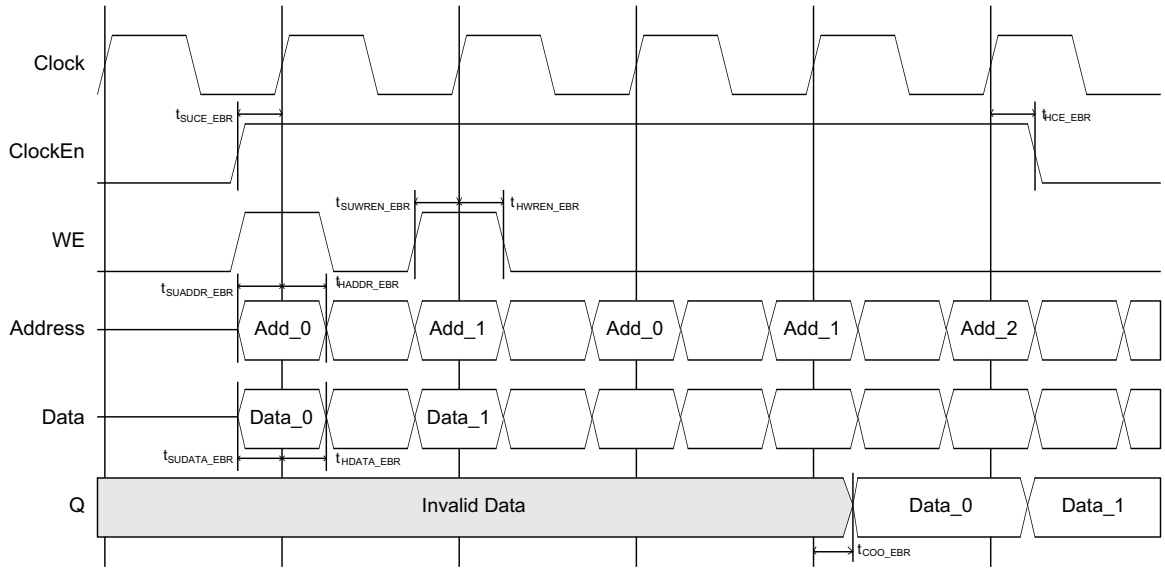


Figure 12-11. Single Port RAM Timing Waveform – READ BEFORE WRITE Mode, Without Output Registers

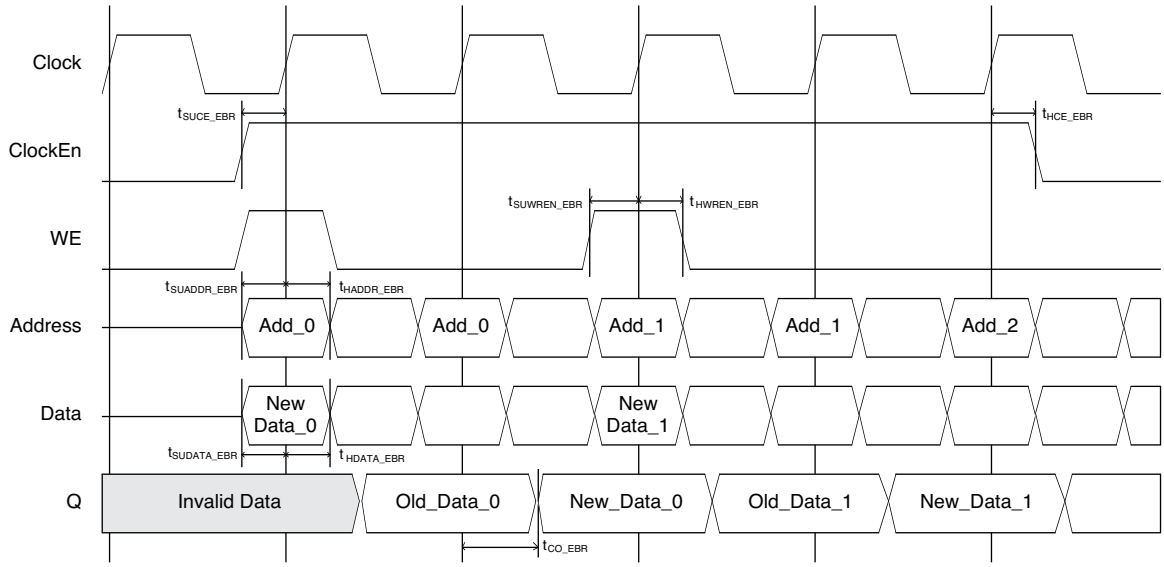


Figure 12-12. Single Port RAM Timing Waveform – READ BEFORE WRITE Mode, With Output Registers

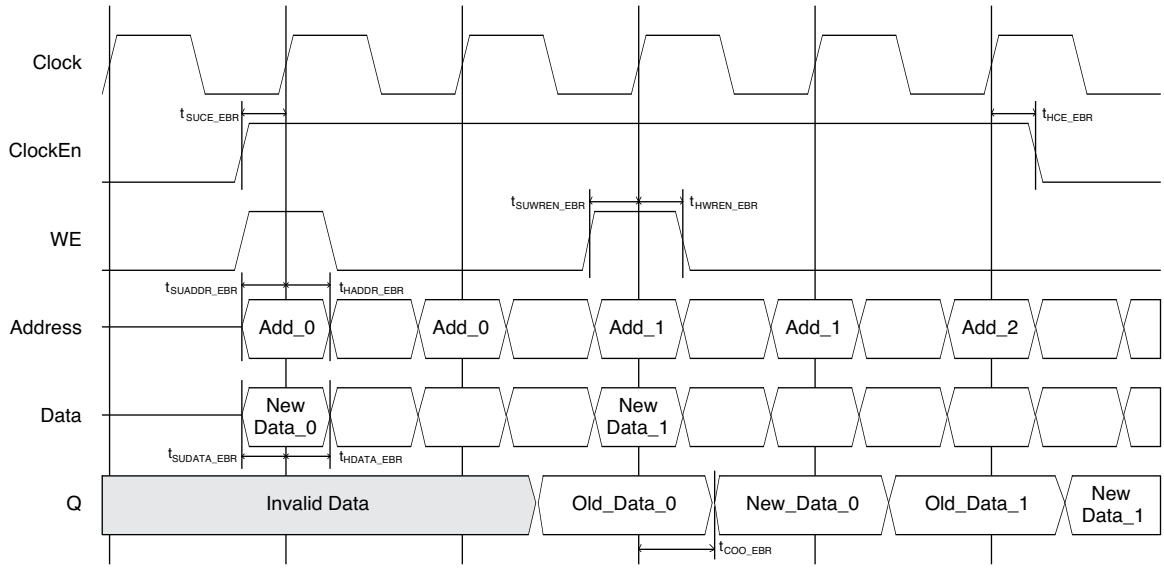


Figure 12-13. Single Port RAM Timing Waveform – WRITE THROUGH Mode, Without Output Registers

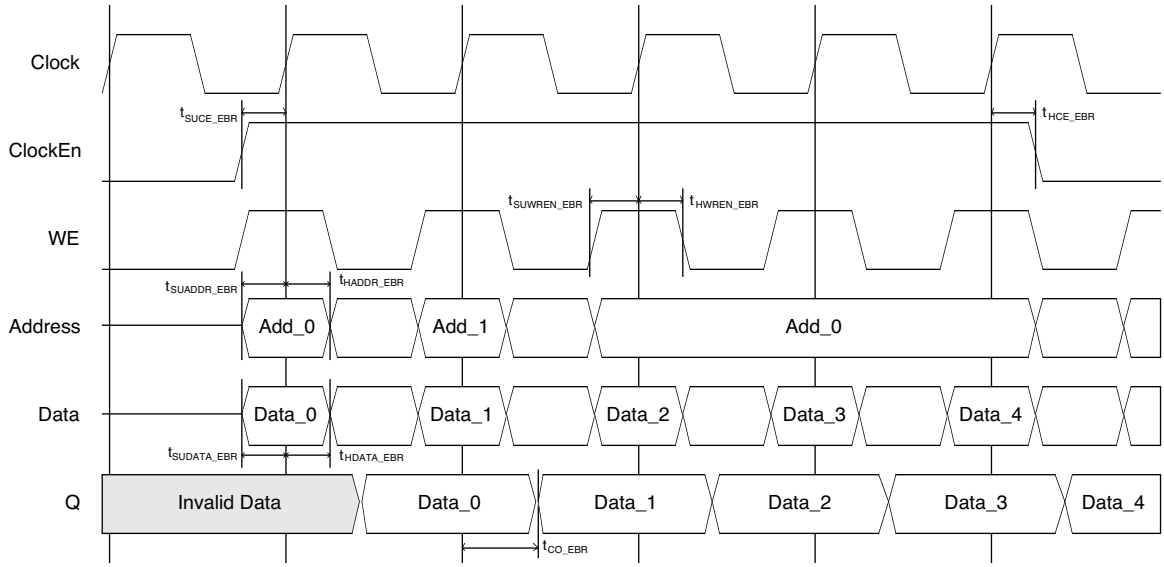
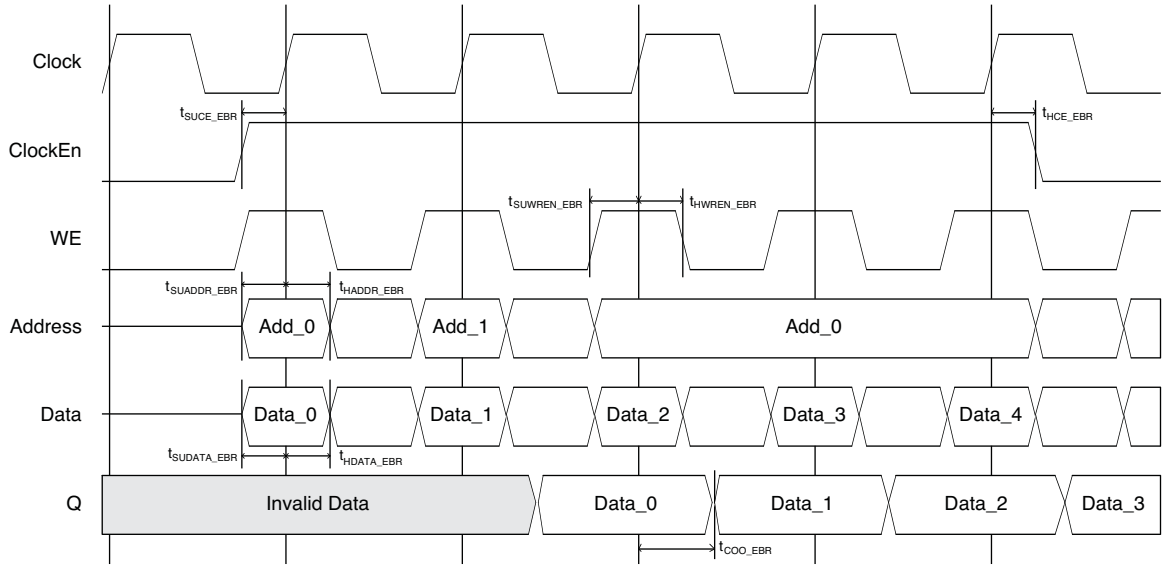


Figure 12-14. Single Port RAM Timing Waveform – WRITE THROUGH Mode, With Output Registers

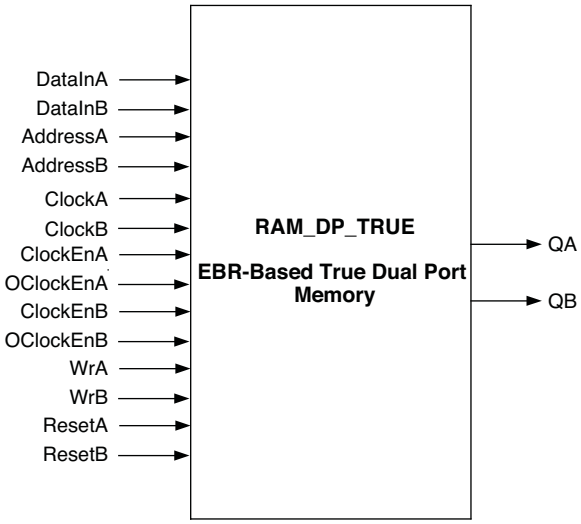


Dual Port RAM (RAM_DP_TRUE) – EBR Based

The EBR blocks in MachXO2 devices can be configured as True-Dual Port RAM (RAM_DP_TRUE). IPexpress allows users to generate the Verilog-HDL or VHDL netlists for various memory sizes depending on design requirements.

IPexpress generates the memory module as shown in Figure 12-15.

Figure 12-15. True Dual Port Memory Module Generated by IPexpress



Since the device has a number of EBR blocks, the generated module makes use of these EBR blocks, or primitives, and cascades them to create the memory sizes specified by the user in the IPexpress GUI. For memory sizes smaller than an EBR block, the module will be created in one EBR block. For memory sizes larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

In True Dual Port RAM mode, the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for True Dual Port Memory are in Table 12-2.

Table12-2. EBR-Based True Dual Port Memory Port Definitions

Port Name in the Generated Module	Description	Active State
DataInA, DataInB	Input Data port A and port B	—
AddressA, AddressB	Address Bus port A and port B	—
ClockA, ClockB	Clock for PortA and PortB	Rising Clock Edge
ClockEnA, ClockEnB[1]	Clock Enables for Port CLKA and CLKB	Active High
*OClockEnA, *OClockEnB[2]	Output Clock Enables for PortA and PortB	Active High
WrA, WrB	Write enable port A and port B	Active High
ResetA, ResetB[3]	Reset for PortA and PortB	Active High
QA, QB	Output Data port A and port B	—
*ByteEnA, *ByteEnB[4]	Byte Enable port A and port B	Active High
*ERROR	Error Check Code	Active High

*Denotes optional port

1. **ClockEnA/B** are used as clock enable for all the input registers.
2. **OClockEnA/B** can be used as clock enable for the optional output registers. This allows the full pipeline of data to be output, including the last word.
3. **Reset** resets only the optional output registers of the RAM. It does not reset the input registers or the contents of memory.
4. **ByteEnA/B** can be used to mask the input data so that only specific bytes of memory are overwritten.

The True Dual Port RAM (RAM_DP_TRUE) can be configured as **NORMAL**, **READ BEFORE WRITE** or **WRITE THROUGH** modes. Each of these modes affects what data comes out of the port Q of the memory during the write operation followed by the read operation at the same memory location.

IPexpress implements the MachXO2 True Dual Port RAM (RAM_DP_TRUE) using the DP8KC primitive.

Figures 12-16 through 12-21 show the internal timing waveforms for the True Dual Port RAM (RAM_DP_TRUE).

Figure 12-16. True Dual Port RAM Timing Waveform – NORMAL Mode, without Output Registers

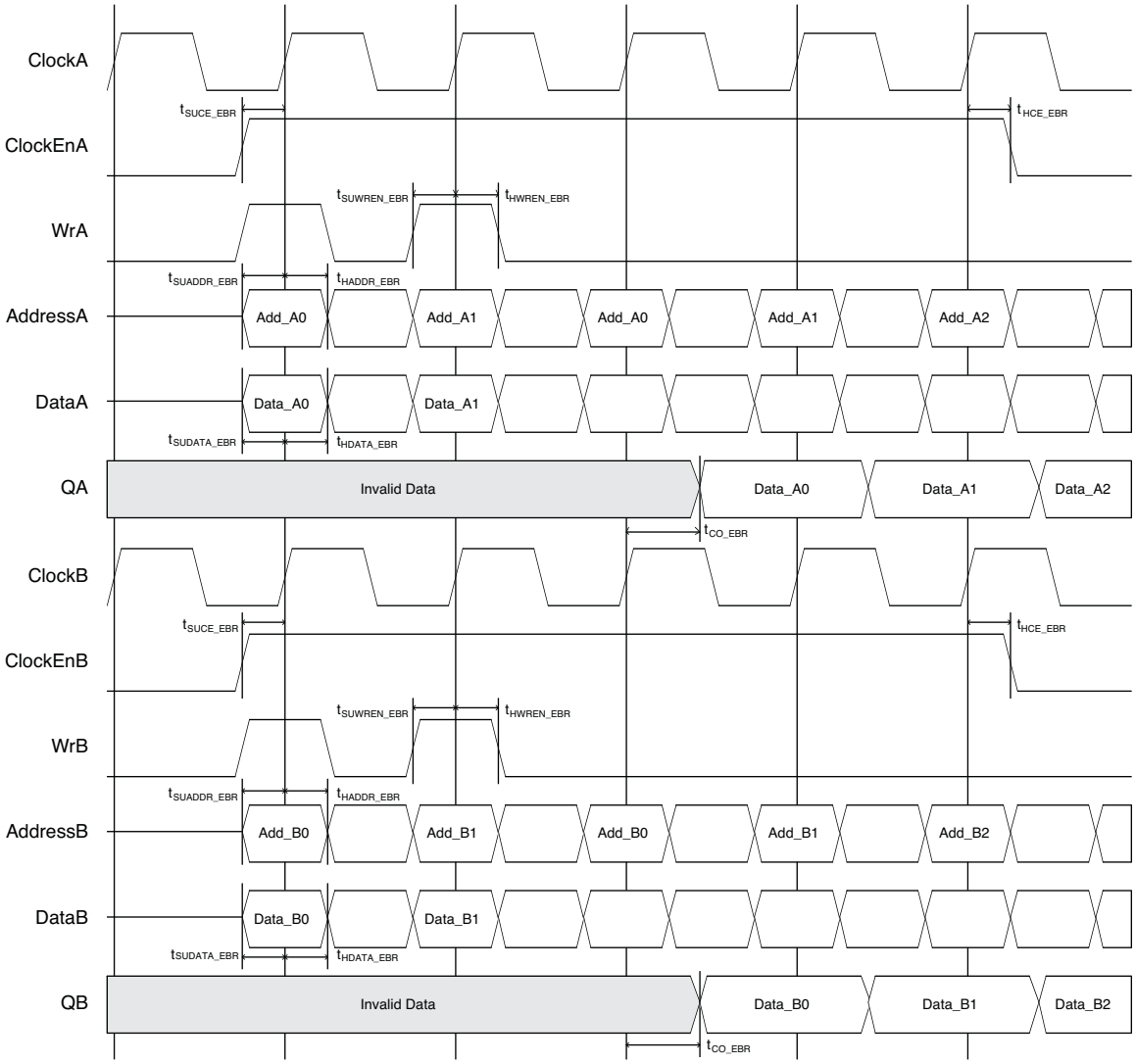


Figure 12-17. True Dual Port RAM Timing Waveform – NORMAL Mode with Output Registers

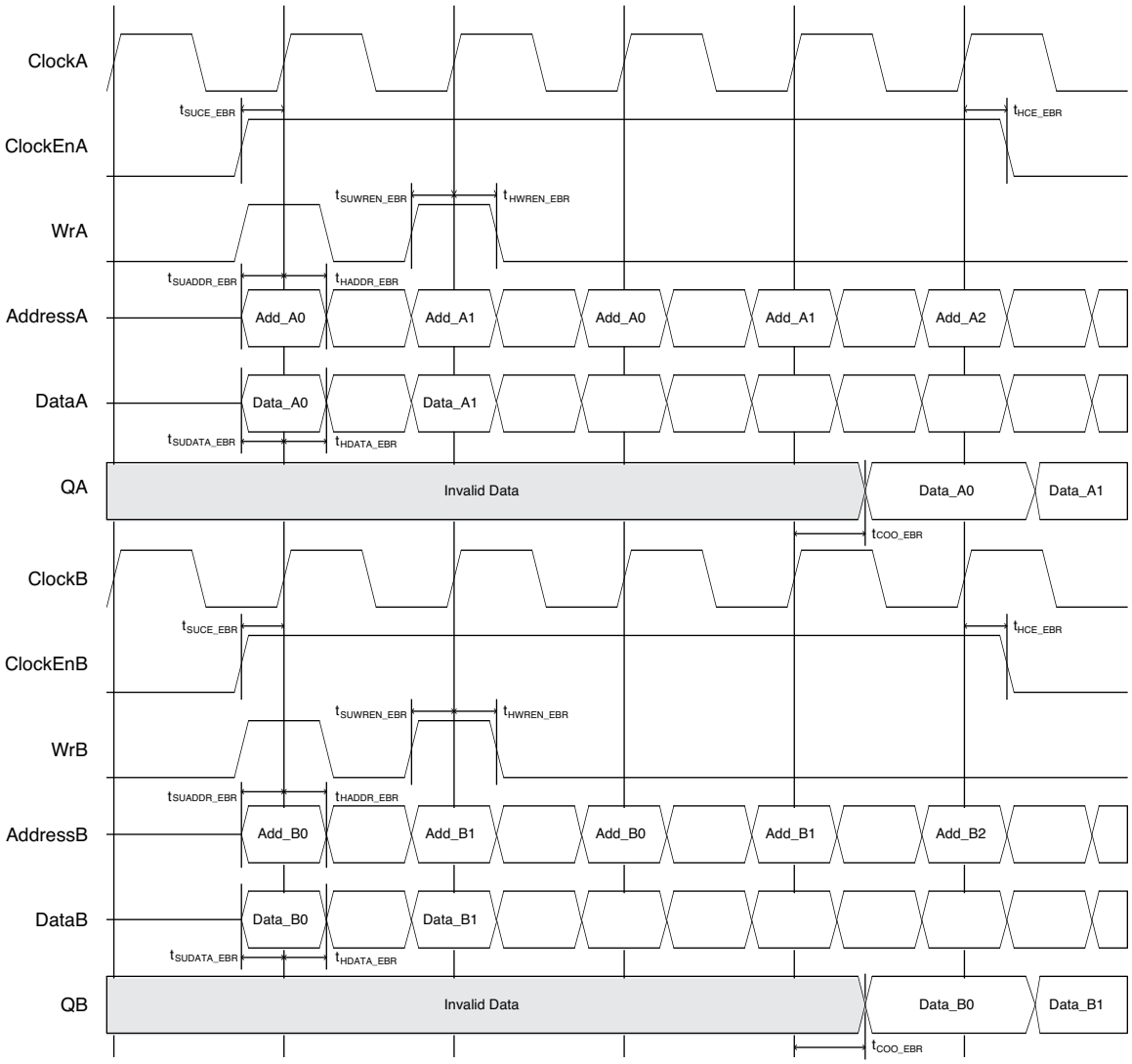


Figure 12-18. True Dual Port RAM Timing Waveform – READ BEFORE WRITE Mode, without Output Registers

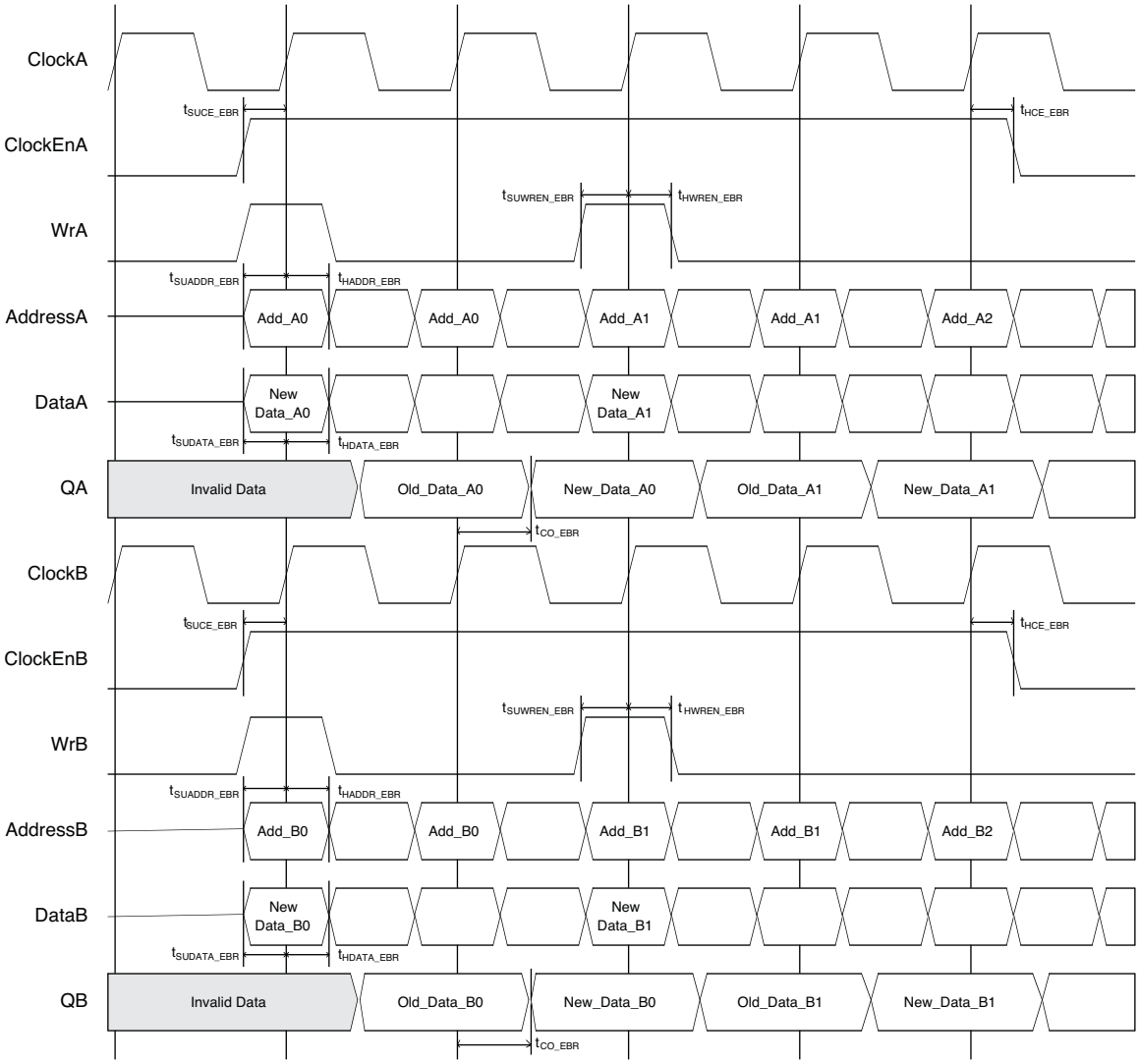


Figure 12-19. True Dual Port RAM Timing Waveform – READ BEFORE WRITE Mode, with Output Registers

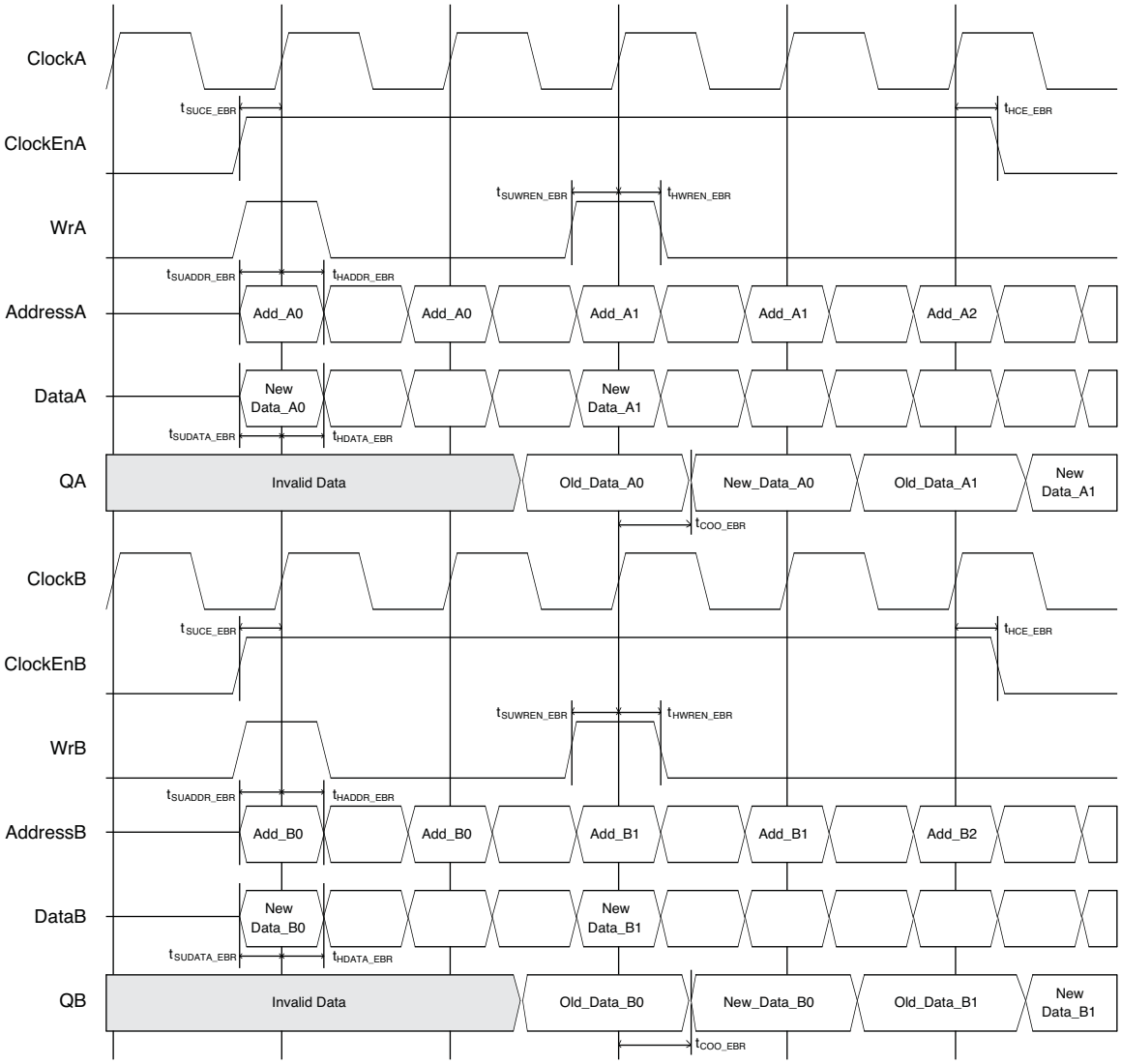


Figure 12-20. True Dual Port RAM Timing Waveform – WRITE THROUGH Mode, without Output Registers

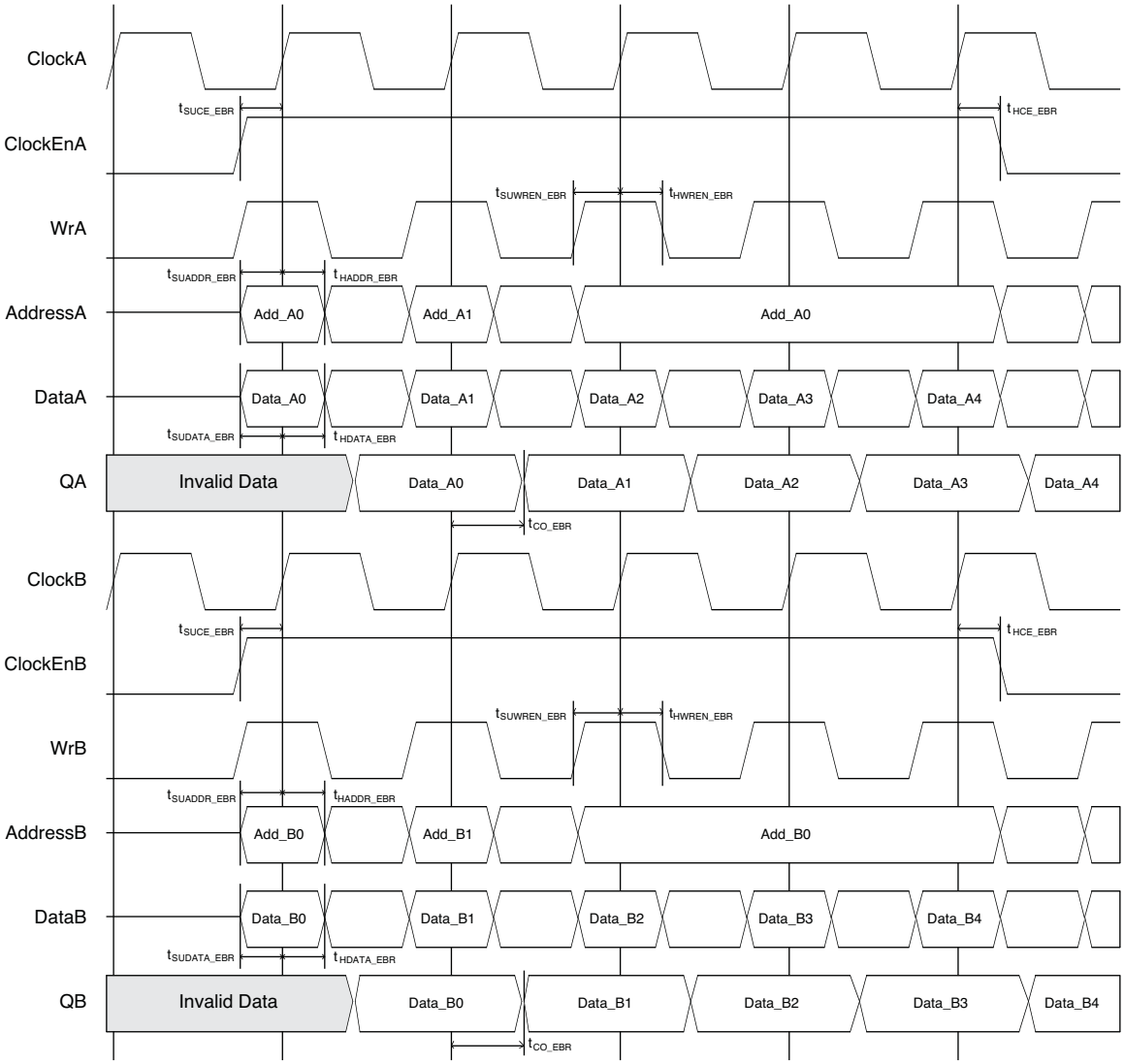
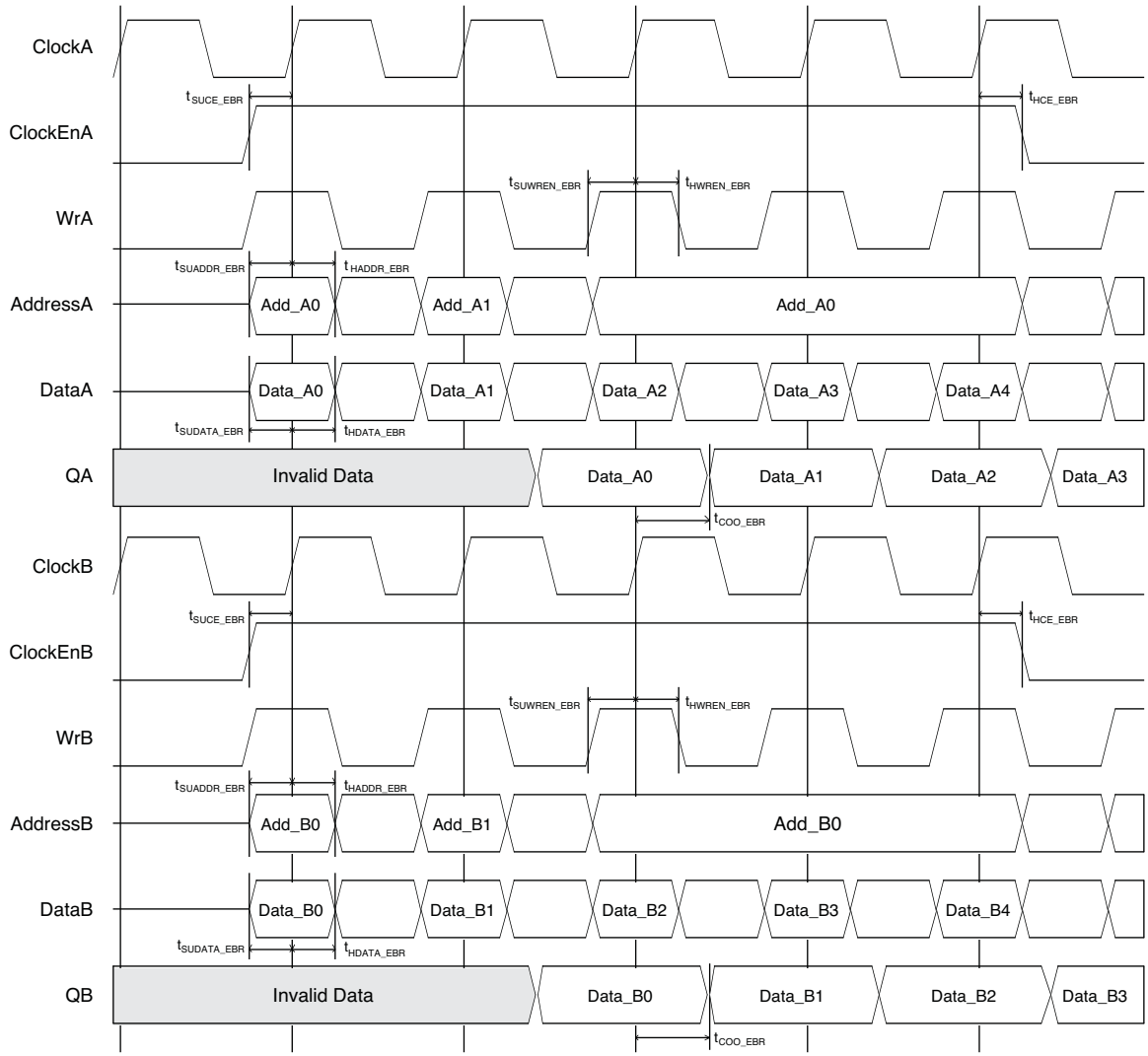


Figure 12-21. True Dual Port RAM Timing Waveform – WRITE THROUGH Mode, with Output Registers

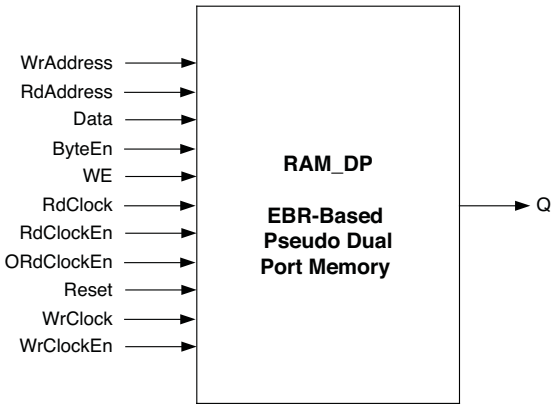


Pseudo Dual Port RAM (RAM_DP) – EBR Based

The EBR blocks in the MachXO2 devices can be configured as Pseudo-Dual Port RAM (RAM_DP). IPexpress allows users to generate the Verilog-HDL or VHDL netlists for various memory sizes depending on design requirements.

IPexpress generates the memory module as shown in Figure 12-22.

Figure 12-22. Pseudo Dual Port Memory Module Generated by IPexpress



Since the device has a number of EBR blocks, the generated module makes use of these EBR blocks, or primitives, and cascades them to create the memory sizes specified by the user in the IPexpress GUI. For memory sizes smaller than an EBR block, the module will be created in one EBR block. For memory sizes larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

In Pseudo Dual Port RAM mode, the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for the Pseudo Dual Port Memory are listed in Table 12-3.

Table12-3. EBR-Based Pseudo-Dual Port Memory Port Definitions

Port Name in the Generated Module	Description	Active State
WrAddress	Write Address	—
RdAddress	Read Address	—
Data	Write Data	—
*ByteEn[1]	Byte Enable	Active High
WE	Write Enable	Active High
RdClock	Read Clock	Rising Edge
RdClockEn[2]	Read Clock Enable	Active High
*ORdClockEn[3]	Read Output Clock Enable	Active High
Reset[4]	Reset	Active High
WrClock	Write Clock	Rising Edge
WrClockEn[2]	Write Clock Enable	Active High
Q	Read Data	—
*ERROR	Error Check Code	Active High

*Denotes optional port

1. **ByteEn** can be used to mask the input data so that only specific bytes of memory are overwritten.
2. **RdClockEn**/**WrClockEn** are used as clock enable for all the input registers.
3. **ORdClockEn** can be used as clock enable for the optional output registers. This allows the full pipeline of data to be output, including the last word.
4. **Reset** resets only the optional output registers of the RAM. It does not reset the input registers or the contents of memory.

IPexpress implements the MachXO2 Pseudo Dual Port RAM (RAM_DP) using the PDPW8KC primitive, or the DP8KC primitive in narrow data port (9 bits or less) configurations.

Figures 12-23 and 12-24 show the internal timing waveforms for the Pseudo Dual Port RAM (RAM_DP).

Figure 12-23. Pseudo-Dual Port RAM Timing Diagram – Without Output Registers

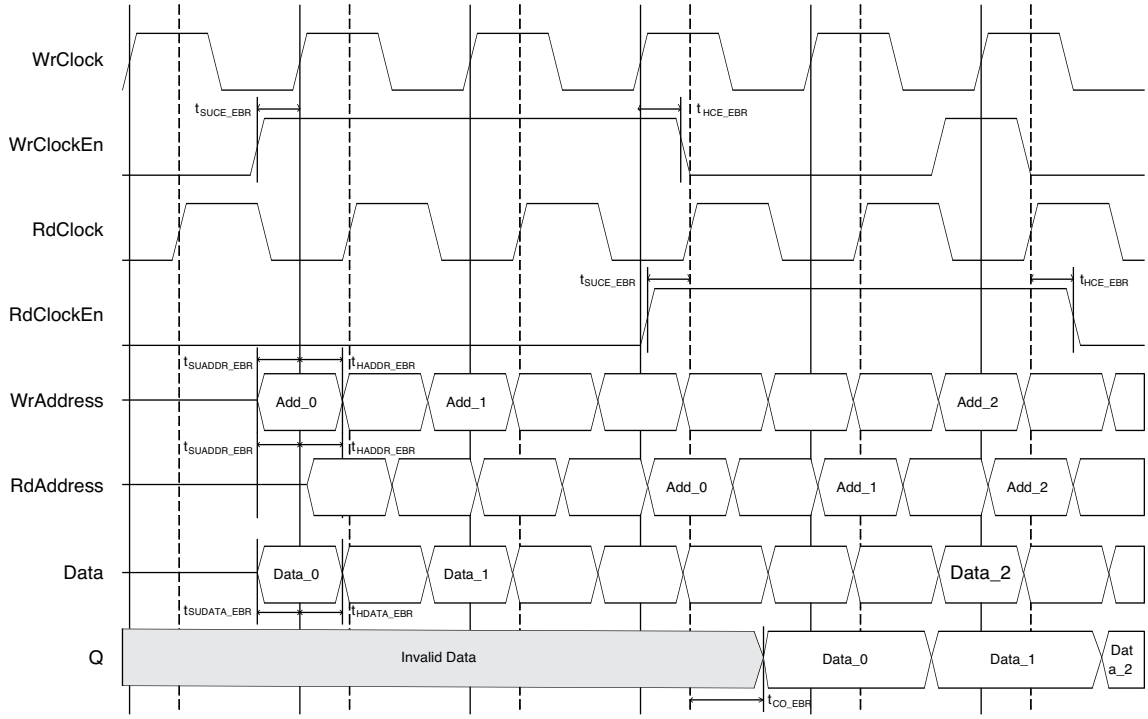
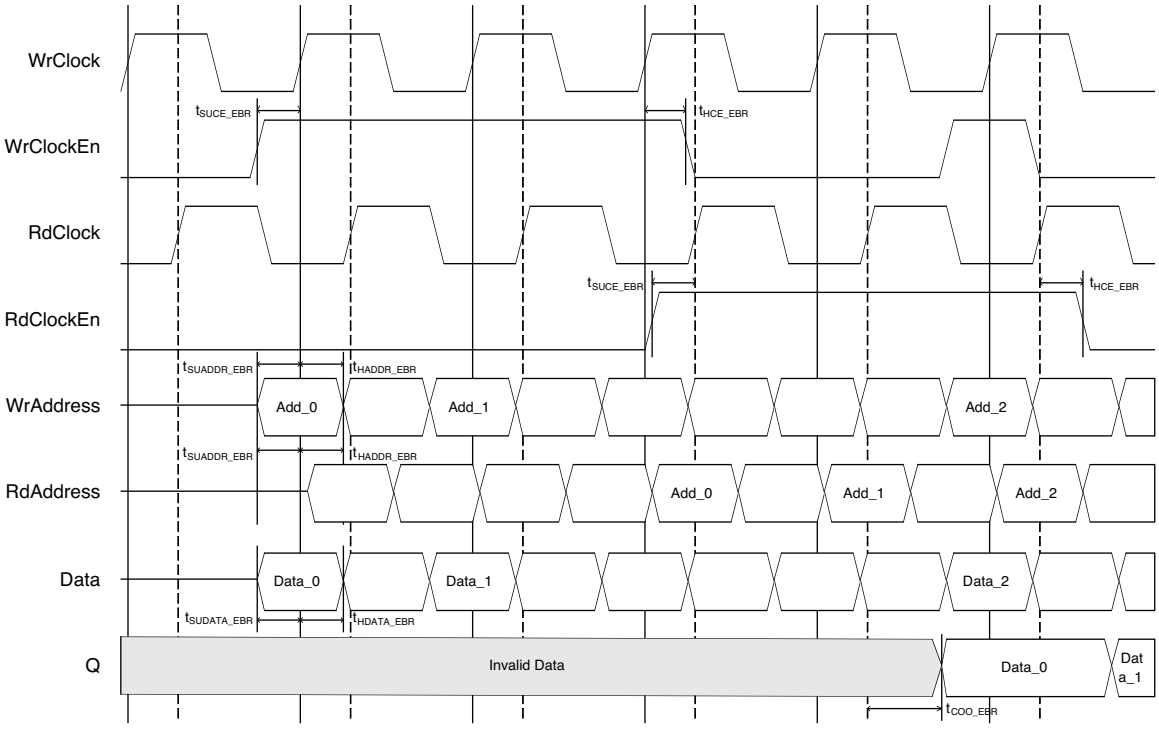


Figure 12-24. Pseudo-Dual Port RAM Timing Diagram – With Output Registers

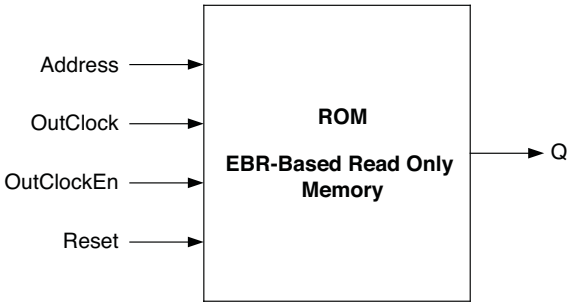


Read Only Memory (ROM) – EBR Based

The EBR blocks in the MachXO2 devices can be configured as Read Only Memory (ROM). IPexpress allows users to generate the Verilog-HDL or VHDL netlist for various memory sizes depending on design requirements. Users are required to provide the ROM memory content in the form of an initialization file.

IPexpress generates the memory module as shown in Figure 12-25.

Figure 12-25. ROM – Read Only Memory Module Generated by IPexpress



Since the device has a number of EBR blocks, the generated module makes use of these EBR blocks, or primitives, and cascades them to create the memory sizes specified by the user in the IPexpress GUI. For memory sizes smaller than an EBR block, the module will be created in one EBR block. For memory sizes larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

The various ports and their definitions for the ROM are listed in Table 12-4.

Table12-4. EBR-Based ROM Port Definitions

Port Name in Generated Module	Description	Active State
Address	Read Address	—
OutClock	Clock	Rising Clock Edge
OutClockEn[1]	Clock Enable	Active High
Reset[2]	Reset	Active High
Q	Read Data	—
*ERROR	Error Check Code	Active High

*Denotes optional port

1. **OutClockEn** can be used as clock enable for the optional output registers.
2. **Reset** resets only the optional output registers of the ROM. It does not reset the contents of the memory.

While generating the ROM using IPexpress, the user must provide the initialization file to pre-initialize the contents of the ROM. These files are the *.mem files and they can be of Binary, Hex or the Addressed Hex formats. The initialization files are discussed in detail in the Initializing Memory section of this document.

IPexpress implements the MachXO2 Read Only Memory (ROM) using an appropriately configured DP8KC primitive with write-enables tied low.

Figures 12-26 and 12-27 show the internal timing waveforms for the Read Only Memory (ROM).

Figure 12-26. ROM Timing Waveform – Without Output Registers

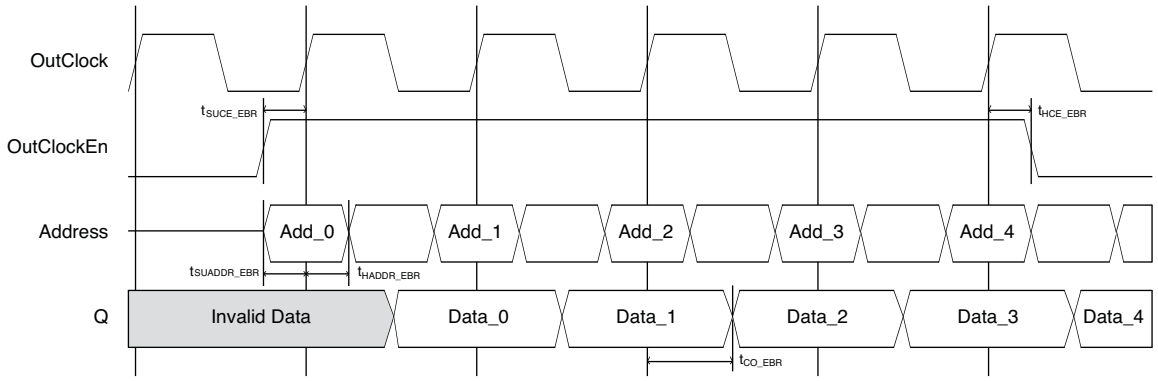
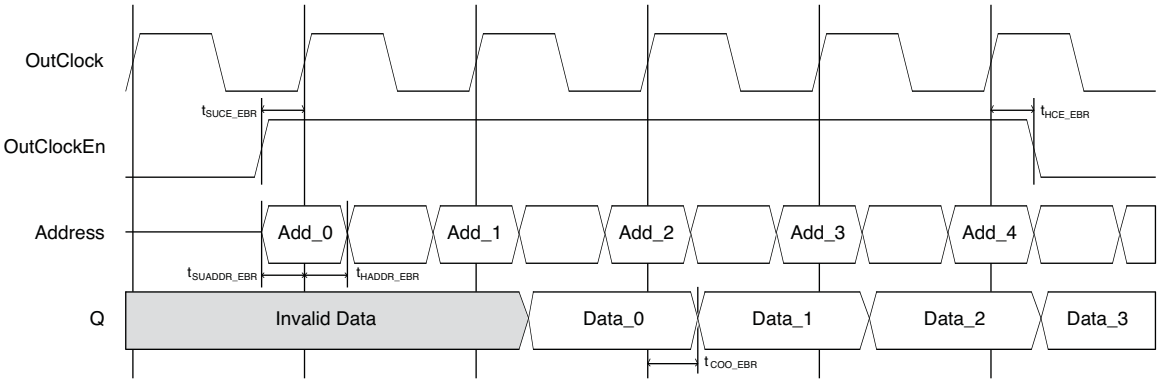


Figure 12-27. ROM Timing Waveform – With Output Registers

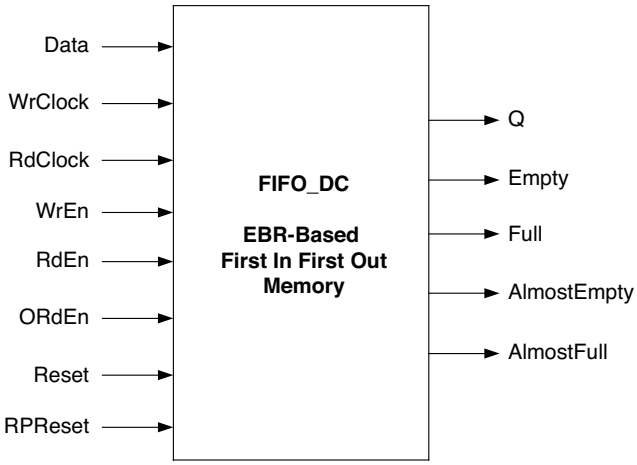


First In First Out (FIFO_DC) – EBR Based

The EBR blocks in MachXO devices can be configured as Dual-Clock First-In First-Out Memory (FIFO_DC). IPexpress allows users to generate the Verilog-HDL or VHDL netlist for various memory sizes depending on design requirements.

IPexpress generates the FIFO_DC memory module as shown in Figure 12-28.

Figure 12-28. FIFO Module Generated by IPexpress



Since the device has a number of EBR blocks, the generated module makes use of these EBR blocks, or primitives, and cascades them to create the memory sizes specified by the user in the IPexpress GUI. For memory sizes smaller than an EBR block, the module will be created in one EBR block. For memory sizes larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

In FIFO_DC mode, the input data is registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for the FIFO_DC are listed in Table 12-5.

Table12-5. EBR-Based FIFO_DC Memory Port Definitions

Port Name in Generated Module	Description	Active State
WrClock	Write Port Clock	Rising Clock Edge
RdClock	Read Port Clock	Rising Clock Edge
WrEn	Write Enable	Active High
RdEn	Read Enable	Active High
*ORdEn[1]	Output Read Enable	Active High
Reset[2]	Reset	Active High
RPReset[3]	Read Pointer Reset	Active High
Q	Data Output	—
Empty	Empty Flag	Active High
Full	Full Flag	Active High
AlmostEmpty	Almost Empty Flag	Active High
AlmostFull	Almost Full Flag	Active High
*ERROR	Error Check Code	Active High

*Denotes optional port

1. **ORdEn** can be used as clock enable for the optional output registers. This allows the full pipeline of data to be output, including the last word.
2. **Reset** resets only the optional output registers, pointer circuitry and flags of the FIFO. It does not reset the input registers or the contents of memory.
3. **RPReset** resets only the read pointer. See additional discussion below.

IPexpress implements the MachXO2 Dual-Clock First-In First-Out Memory (FIFO_DC) using the FIFO8KB primitive.

FIFO_DC Flags

The FIFO_DC have four flags available: Empty, Almost Empty, Almost Full and Full. Almost Empty and Almost Full flags have a programmable range.

The program ranges for the four FIFO_DC flags are specified in Table 6.

Table12-6. FIFO_DC Flag Settings

Module Flag Name	Description	Programming Range
Full	Full flag setting	1 to (2^N - 1)
AlmostFull	Almost full setting	1 to (FULL -1)
AlmostEmpty	Almost empty setting	1 to (FULL -1)
Empty	Empty setting	0

The value of Empty is fixed at 0. When coming out of reset, the active high flags Empty and Almost Empty are set to high, since they are true.

The user should specify the absolute value of the address at which the Almost Empty and Almost Full flags will go true. For example, if the Almost Full flag is required to go true at the address location 500 for a FIFO of depth 512, the user should specify a value of 500 in IPexpress.

The Empty and Almost Empty flags are always registered to the read clock and the Full and Almost Full flags are always registered to the write clock.

At reset both the write and read counters are pointing to address zero. After reset is de-asserted data can be written into the FIFO_DC to the address pointed to by the write counter at the positive edge of the write clock when the write enable is asserted

Similarly, data can be read from the FIFO_DC from the address pointed to by the read counter at the positive edge of the read clock when read enable is asserted.

Read Pointer Reset (RPReset) is used to facilitate a retransmit operation and is more commonly used in “packetized” communications. Asserting RPReset causes the internal read pointer to be reset to zero. It is typically used in conjunction with the assertion of Reset prior to each new ‘packet’ which resets both read and write pointers to zero. In this application, the user must keep careful track of when a packet is written into or read from the FIFO_DC. To avoid the possible corruption of memory, RPReset should not be asserted until the prior read cycle is complete (i.e. RdEn deasserted for one clock period). Upon the deassertion of RPReset, the Empty and Almost Empty flags assume their correct state after one read clock cycle – this is a regular condition known as boundary cycle latency.

The data output of the FIFO_DC can be registered or non-registered through a selection in IPexpress. The output registers are enabled by read enable.

FIFO_DC Dual and Dynamic Threshold Options

The optional Almost Full and Almost Empty flag thresholds may be individually set for single (default) or dual threshold operation. In addition, the thresholds may be static at configuration (default) or dynamically set through optional ports. The implementation of Dual or Dynamic thresholds automatically creates supporting LUT-based logic.

Table12-7. EBR-Based FIFO_DC Optional Dynamic Threshold Port Definitions

Port Name in Generated Module	Description
AmEmptyThresh	Almost Empty Single Threshold
AmFullThresh	Almost Full Single Threshold
AmEmptySetThresh	Almost Empty Set Threshold
AmEmptyClrThresh	Almost Empty Clear Threshold
AmFullSetThresh	Almost Full Set Threshold
AmFullClrThresh	Almost Full Clear Threshold

FIFO_DC Operation

If the output registers are not enabled it will take two clock cycles to read the first word out. The register for the flag logic causes this extra clock latency. In the architecture of the emulated FIFO_DC, the internal read enables for reading the data out is controlled not only by the read enable provided by the user but also the empty flag. When the data is written into the FIFO, an internal empty flag is registered using write clock that is enabled by write enable (WrEn). Another clock latency is added due to the clock domain transfer from write clock to read clock using another register which is clocked by read clock that is enabled by read enable.

Internally, the output of this register is inverted and then ANDed with the user-provided read enable that becomes the internal read enable to the RAM_DP which is at the core of the FIFO_DC.

Thus, the first read data takes two clock cycles to propagate through. During the first data out, read enable goes high for one clock cycle, empty flag is de-asserted and is not propagated through the second register enabled by the read enable. The first clock cycle brings the Empty Low and the second clock cycle brings the internal read enable high (RdEn and !EF) and then the data is read out by the second clock cycle. Similarly, the first write data after the full flag has a similar latency.

If the user has enabled the output registers, the output registers will cause an extra clock delay during the first data out as they are clocked by the read clock and enabled by the read enable.

1. First RdEn and Clock Cycle to propagate the EF internally.
2. Second RdEn and Clock Cycle to generate internal Read Enable into the DPRAM.
3. Third RdEn and Clock Cycle to get the data out of the output registers.

Figures 12-29 and 12-30 show the internal timing waveforms for the Dual Clock FIFO (FIFO_DC).

Figure 12-29. FIFO_DC Without Output Registers (Non-Pipelined)

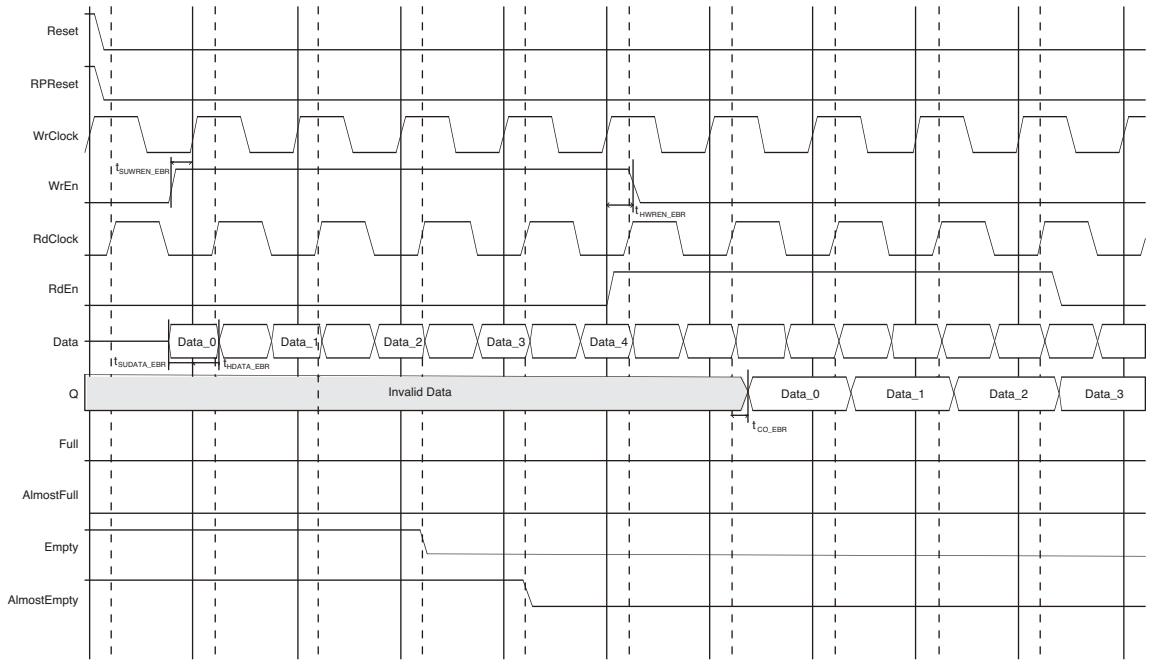
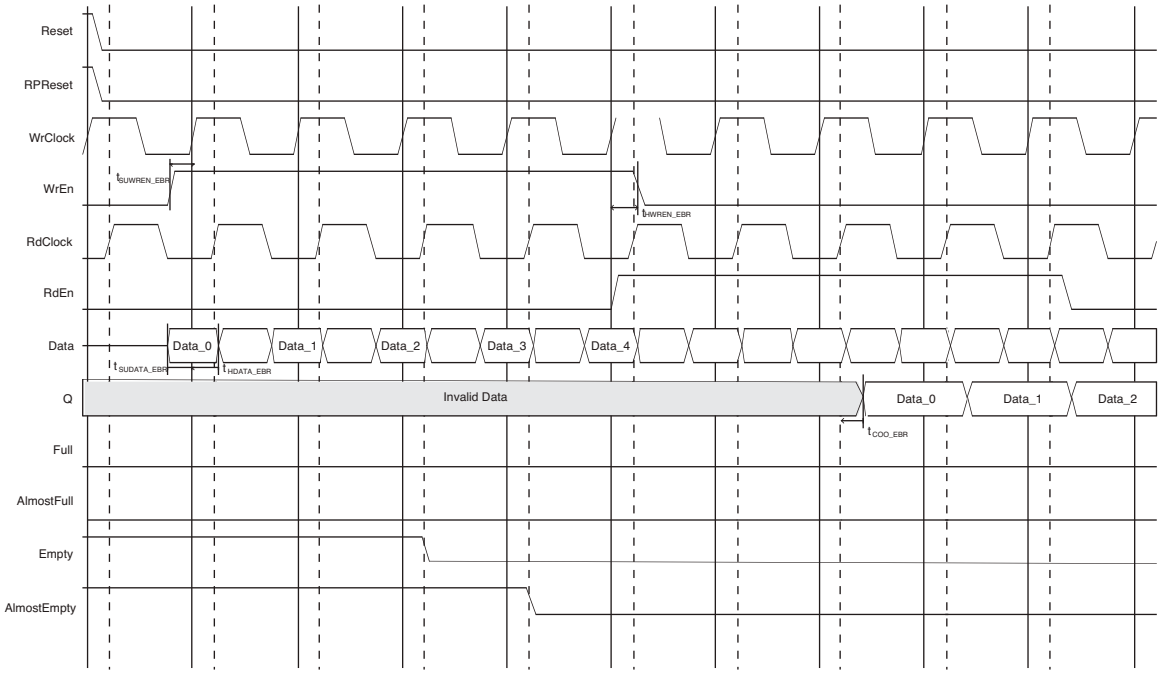


Figure 12-30. FIFO_DC With Output Registers (Pipelined)

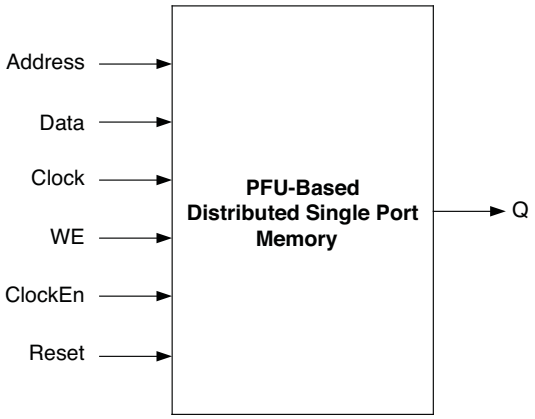


Distributed Single Port RAM (Distributed_SPRAM) – PFU Based

PFU-based Distributed Single Port RAM is created using the 4-input LUT (Look-Up Table) available in the PFU. These LUTs can be cascaded to create a larger Distributed Memory sizes.

Figure 12-31 shows the Distributed Single Port RAM module as generated by IPexpress.

Figure 12-31. Distributed Single Port RAM Module Generated by IPexpress



The generated module makes use of the 4-input LUTs available in the PFU. Additional decode logic is generated by utilizing the resources available in the PFU.

The various ports and their definitions for the Memory are as per Table 12-8.

Table12-8. PFU-Based Distributed Single Port RAM Port Definitions

Port Name in Generated Module	Description	Active State
Address	Address	—
Data	Data In	—
Clock	Clock	Rising Clock Edge
WE	Write Enable	Active High
ClockEn	Clock Enable	Active High
Reset[1]	Reset	Active High
Q	Data Out	—

1. **Reset** is available only when Output Registers are enabled.

Figures 12-32 and 12-33 show the internal timing waveforms for the Distributed Single Port RAM (Distributed_SPRAM).

Figure 12-32. PFU-Based Distributed Single Port RAM Timing Waveform – Without Output Registers

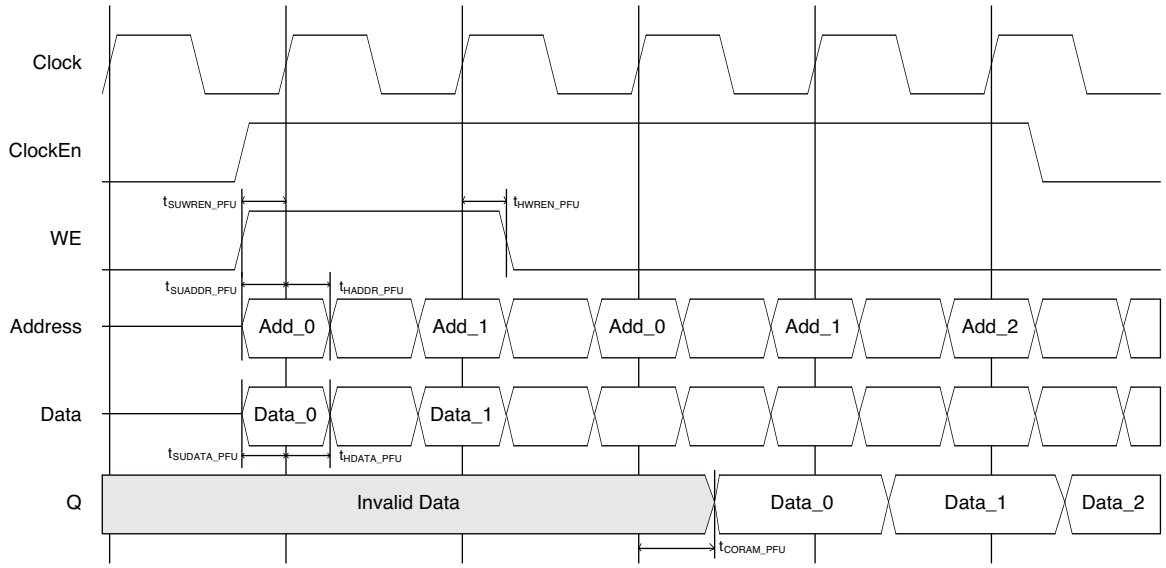
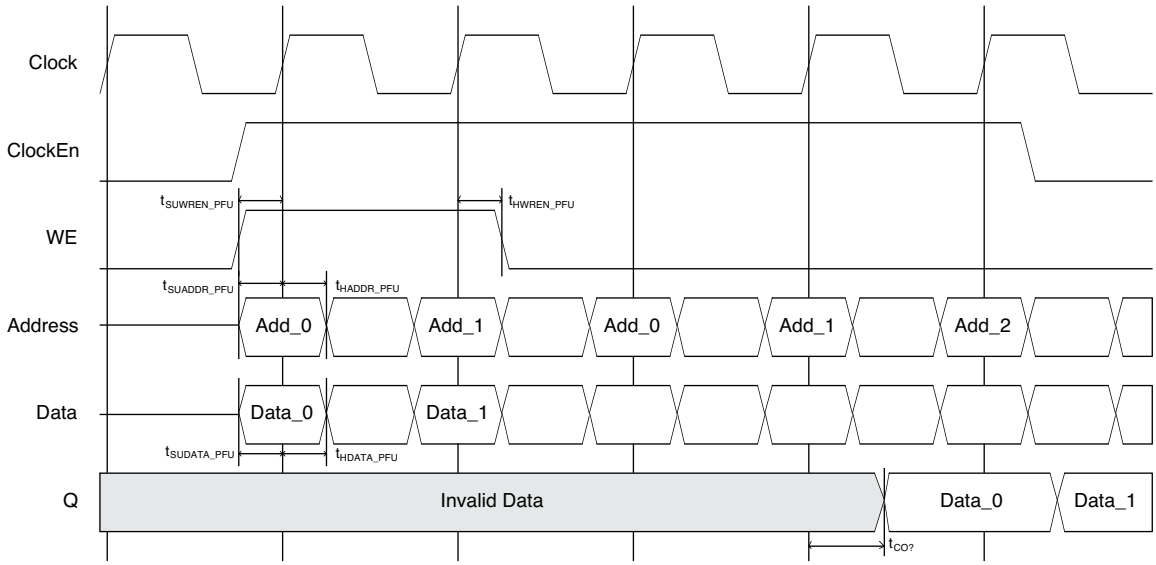


Figure 12-33. PFU- Based Distributed Single Port RAM Timing Waveform – With Output Registers

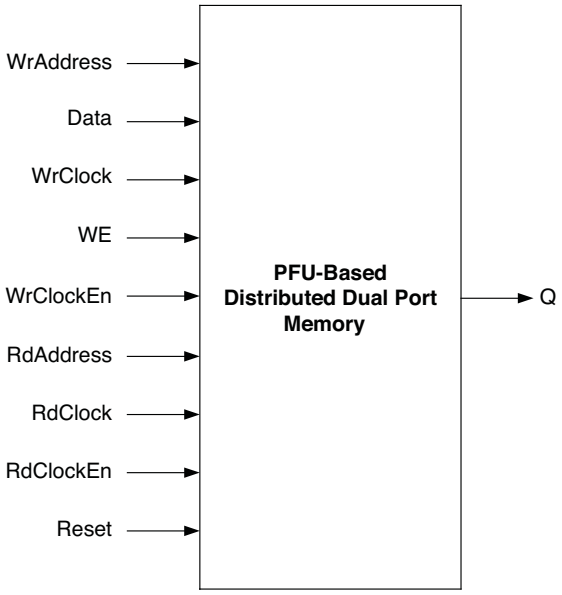


Distributed Dual Port RAM (Distributed_DPRAM) – PFU Based

PFU-based Distributed Dual Port RAM is created using the 4-input LUT (Look-Up Table) available in the PFU. These LUTs can be cascaded to create larger Distributed Memory sizes.

Figure 12-34 shows the Distributed Dual Port RAM module as generated by IPexpress.

Figure 12-34. Distributed Dual Port RAM Module Generated by IPexpress



The generated module makes use of the 4-input LUTs available in the PFU. Additional decode logic is generated by utilizing the resources available in the PFU.

The various ports and their definitions are listed in Table 12-9.

Table12-9. PFU-Based Distributed Dual Port RAM Port Definitions

Port Name in Generated Module	Description	Active State
WrAddress	Write Address	—
Data	Data Input	—
WrClock	Write Clock	Rising Clock Edge
WE	Write Enable	Active High
WrClockEn	Write Clock Enable	Active High
RdAddress	Read Address	—
*RdClock	Read Clock	Rising Clock Edge
*RdClockEn	Read Clock Enable	Active High
Reset*	Reset	Active High
Q	Data Out	—

*Denotes optional port.

The optional ports Read Clock (RdClock) and Read Clock Enable (RdClockEn) are not available in the hardware primitive. These are generated by IPexpress when the user wants to enable the output registers in the IPexpress configuration.

Figures 12-35 and 12-36 show the internal timing waveforms for the Distributed Dual Port RAM (Distributed_DPRAM).

Figure 12-35. PFU-Based Distributed Dual Port RAM Timing Waveform – Without Output Registers

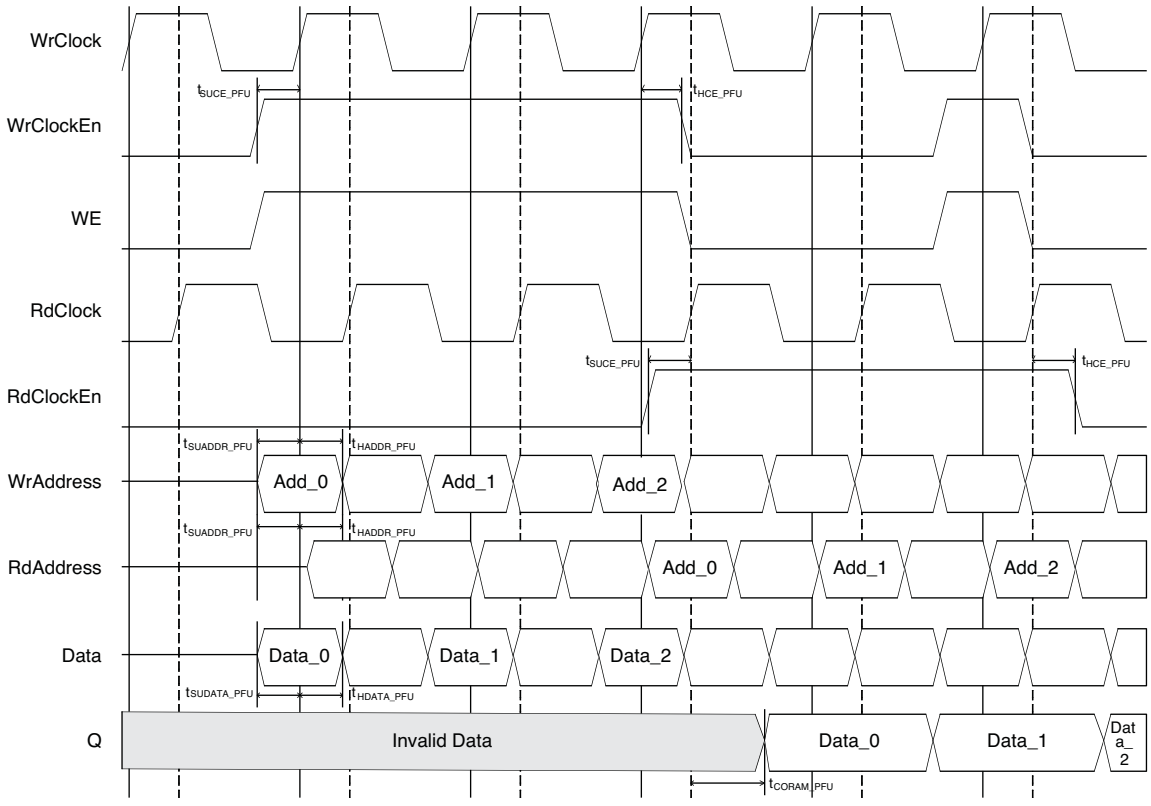
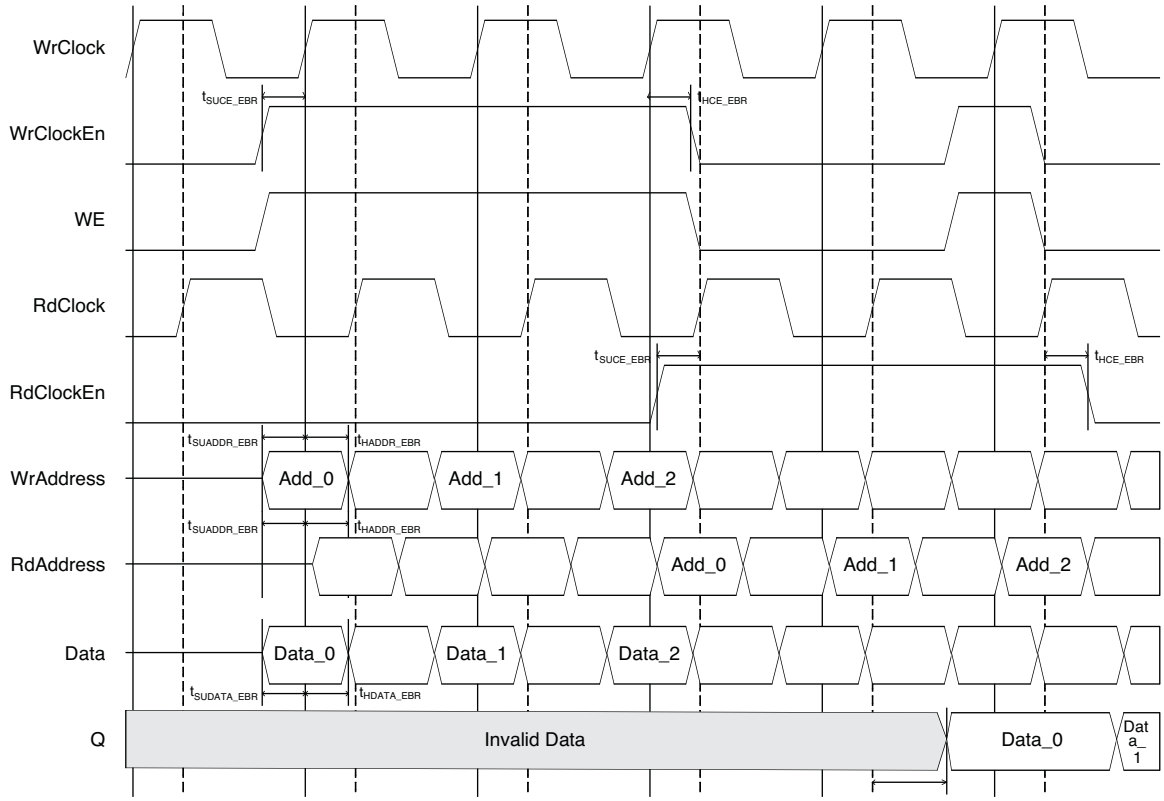


Figure 12-36. PFU-Based Distributed Dual Port RAM Timing Waveform – With Output Registers

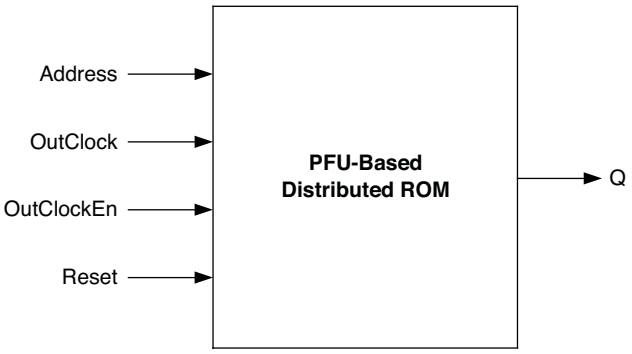


Distributed ROM (Distributed_ROM) – PFU-Based

PFU-based Distributed ROM is created using the 4-input LUT (Look-Up Table) available in the PFU. These LUTs can be cascaded to create a larger Distributed Memory sizes.

Figure 12-37 shows the Distributed ROM module as generated by IPexpress.

Figure 12-37. Distributed ROM Generated by IPexpress



The generated module makes use of the 4-input LUTs available in the PFU. Additional decode logic is generated by utilizing the resources available in the PFU.

The various ports and their definitions are listed in Table 12-10.

Table12-10. PFU-Based Distributed ROM Port Definitions

Port Name in Generated Module	Description	Active State
Address	Address	—
OutClock*	Out Clock	Rising Clock Edge
OutClockEn*	Out Clock Enable	Active High
Reset*	Reset	Active High
Q	Data Out	—

*Denotes optional port.

The optional ports Out Clock (OutClock) and Out Clock Enable (OutClockEn) are not available in the hardware primitive. These are generated by the IPexpress when the user wants to enable the output registers in the IPexpress configuration.

Figures 12-38 and 12-39 show the internal timing waveforms for the Distributed ROM.

Figure 12-38. PFU-Based ROM Timing Waveform – Without Output Registers

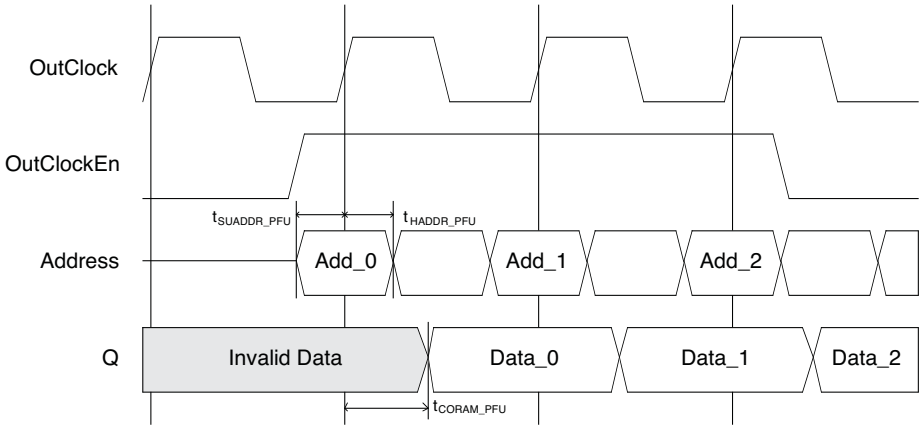
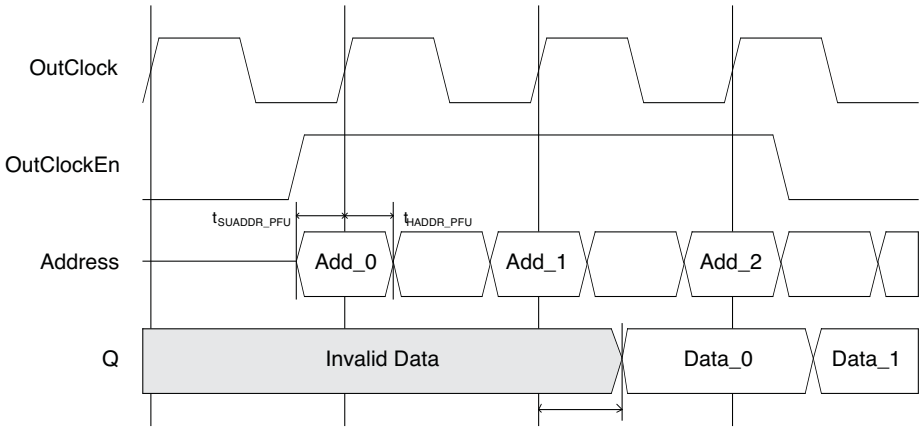


Figure 12-39. PFU-Based ROM Timing Waveform – With Output Registers

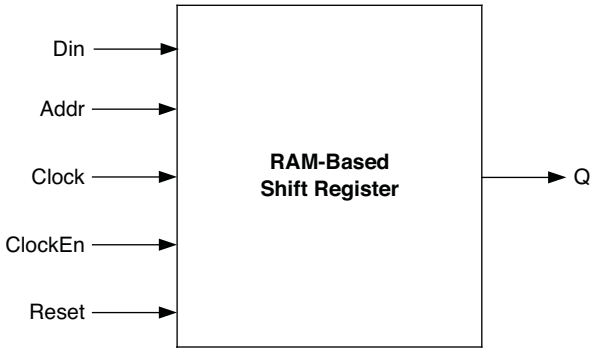


RAM-Based Shift Register

The Distributed SPRAM blocks in the MachXO2 devices, in combination with LUT-based logic, can be configured as a RAM-based Shift Register. IPexpress allows users to generate the Verilog-HDL or VHDL netlist for the Shift Register length, as per design requirements.

IPexpress generates the Shift Register module as shown in Figure 12-40.

Figure 12-40. RAM-Based Shift Register Generated by IPexpress



The generated module makes use of the 4-input LUTs available in the PFU. Additional logic is generated by utilizing the resources available in the PFU.

The various ports and their definitions are listed in Table 12-11.

Table12-11. RAM-Based Shift Register Port Definitions

Port Name in Generated Module	Description	Active State
Din	Data In	—
*Addr	Address	—
Clock	Clock	Rising Clock Edge
ClockEn	Clock Enable	Active High
Reset	Reset	Active High
Q	Data Out	—

*Denotes optional port.

The optional Addr port is available only when Variable Length type is selected. It is generated by IPexpress when the user wants to enable the Variable Length operation in the IPexpress configuration. Figures 12-41 and 12-42 show the internal timing waveforms for the RAM-Based Shift Register.

Figure 12-41. RAM-Based Shift Register Timing Waveform – Without Output Registers (Shift = 2)

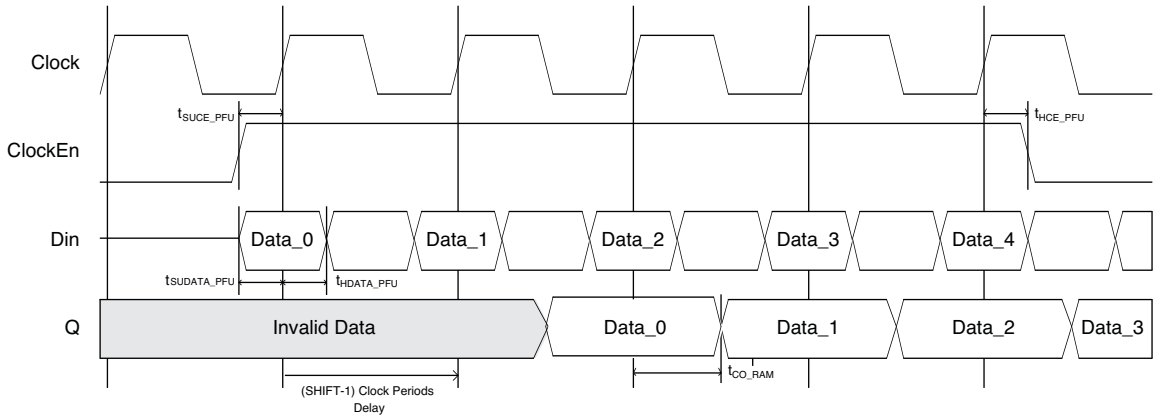
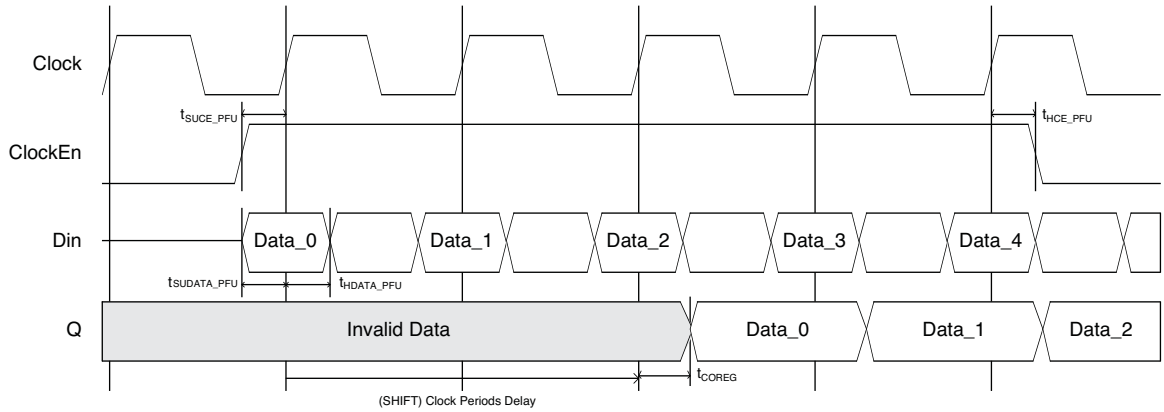


Figure 12-42. RAM-Based Shift Register Timing Waveform – With Output Registers (Shift = 2)



MachXO2 Primitives

Single Port RAM (SP8KC) – EBR Based

The Single Port RAM primitive is shown below.

Figure 12-43. Single Port RAM (SP8KC)

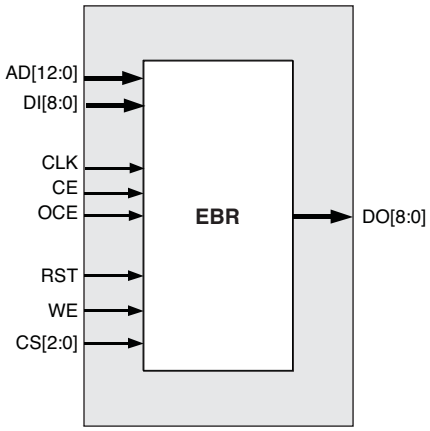


Table12-12. EBR-Based Single Port Memory Port Definitions

Port Name in the EBR Block Primitive (SP8KC)	Description	Active State
AD	Address Bus	—
DI	Data In	—
CLK	Clock	Rising Clock Edge
CE	Clock Enable	Active High
OCE	Output Clock Enable	Active High
RST	Reset	Active High
WE	Write Enable	Active High
CS[2:0]	Chip Select	—
DO	Data Out	—

Each SP8KC primitive consists of 9,216 bits of RAM. The possible values for address depth and data width for the SP8KC primitive are listed in Table 12-13.

Table12-13. Single Port Memory Sizes for 9K Memories in MachXO2

Single Port Memory Size	Input Data	Output Data	Address [MSB:LSB]
8K x 1	DI	DO	AD[12:0]
4K x 2	DI[1:0]	DO[1:0]	AD[12:1]
2K x 4	DI[3:0]	DO[3:0]	AD[12:2]
1K x 9	DI[8:0]	DO[8:0]	AD[12:3]

Table 12-14 shows the various attributes available for the SP8KC. Some of these attributes are user selectable through the IPexpress GUI. For detailed attribute definitions, refer to Appendix A.

Table12-14. Single Port RAM Attributes for MachXO2 (SP8KC)

Attribute	Description	Values	Default Value	User Selectable through IPexpress
DATA_WIDTH	Data Word Width	1, 2, 4, 9	9	Yes
REGMODE	Register Mode (Pipelining)	NOREG, OUTREG	NOREG	Yes
RESETMODE	Selects Reset Type	ASYNC, SYNC	SYNC	Yes
CSDECODE	Chip Select Decode	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111 0b000	0b000	No
WRITEMODE	Read/Write Behavior	NORMAL, WRITE-THROUGH, READBEFORE-WRITE	NORMAL	Yes
GSR	Enable Global Set Reset	ENABLED, DISABLED	DISABLED	No
INITVAL_00 .. INITVAL_1F	Initialization Value	0x00000000000000000000000000000000000000000000000000000000000000000000000000000000 .... 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF (80- character hex strings)	0x00000000000000000000000000000000000000000000000000000000000000000000000000000000	No
ASYNC_RESET_RELEASE	Reset Release	ASYNC, SYNC	SYNC	Yes
INIT_DATA	Init Values Status	STATIC, DYNAMIC	STATIC	Yes

True Dual Port RAM (DP8KC) – EBR-Based

The True Dual Port RAM primitive is shown below.

Figure 12-44. True Dual-Port RAM (DP8KC)

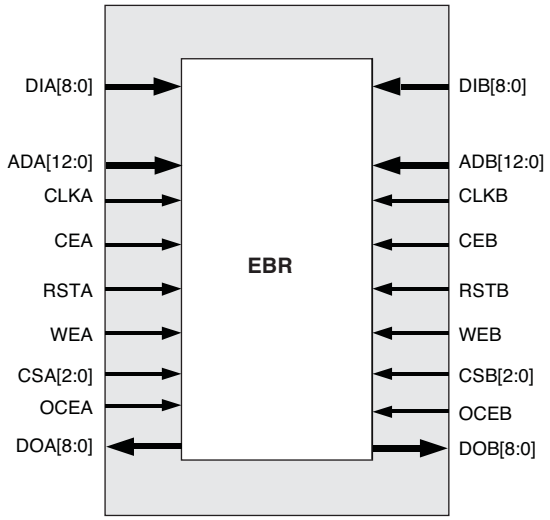


Table12-15. EBR-Based True Dual Port Memory Port Definitions

Port Name in the EBR Block Primitive (DP8KC)	Description	Active State
DIA, DIB	Input Data port A and port B	—
ADA, ADB	Address Bus port A and port B	—
CLKA, CLKB	Clock for PortA and PortB	Rising Clock Edge
CEA, CEB	Clock Enables for Port CLKA and CLKB	Active High
RSTA, RSTB	Reset for PortA and PortB	Active High
WEA, WEB	Write enable port A and port B	Active High
CSA[2:0], CSB[2:0]	Chip Selects for each port	—
OCEA, OCEB	Output Clock Enables for PortA and PortB	Active High
DOA, DOB	Output Data port A and port B	—

Each DP8KC primitive consists of 9,216 bits of RAM. The possible values for address depth and data width for the DP8KC primitive are listed in Table 12-16.

Table12-16. Dual Port Memory Sizes for 9K Memory in MachXO2

Dual Port Memory Size	Input Data Port A	Input Data Port B	Output Data Port A	Output Data Port B	Address Port A [MSB:LSB]	Address Port B [MSB:LSB]
8K x 1	DIA	DIB	DOA	DOB	ADA[12:0]	ADB[12:0]
4K x 2	DIA[1:0]	DIB[1:0]	DOA[1:0]	DOB[1:0]	ADA[12:1]	ADB[12:1]
2K x 4	DIA[3:0]	DIB[3:0]	DOA[3:0]	DOB[3:0]	ADA[12:2]	ADB[12:2]
1K x 9	DIA[8:0]	DIB[8:0]	DOA[8:0]	DOB[8:0]	ADA[12:3]	ADB[12:3]

Table 12-17 shows the various attributes available for the True Dual Port Memory (RAM_DP_TRUE). Some of these attributes are user-selectable through the IPexpress GUI. For detailed attribute definitions, refer to the Appendix A.

Table12-17. Dual Port RAM Attributes for MachXO2 (DP8KC)

Attribute	Description	Values	Default Value	User Selectable through IPexpress
DATA_WIDTH_A	Data Word Width Port A	1, 2, 4, 9	9	Yes
DATA_WIDTH_B	Data Word Width Port B	1, 2, 4, 9	9	Yes
REGMODE_A	Register Mode (Pipelining) for Port A	NOREG, OUTREG	NOREG	Yes
REGMODE_B	Register Mode (Pipelining) for Port B	NOREG, OUTREG	NOREG	Yes
RESETMODE	Selects the Reset type	ASYNC, SYNC	SYNC	Yes
CSDECODE_A	Chip Select Decode for Port A	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	No
CSDECODE_B	Chip Select Decode for Port B	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	No
WRITEMODE_A	Read / Write Mode for Port A	NORMAL, WRITE-THROUGH, READBEFORE-WRITE	NORMAL	Yes
WRITEMODE_B	Read / Write Mode for Port B	NORMAL, WRITE-THROUGH, READBEFORE-WRITE	NORMAL	Yes
GSR	Enables Global Set Reset	ENABLE, DISABLE	DISABLED	No
INITVAL_00 .. INITVAL_1F	Initialization Value	0x00000000000000000000000000000000000000000000000000000000000000000000000000000000 .... 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF (80-character hex strings)	0x00000000000000000000000000000000000000000000000000000000000000000000000000000000	No
ASYNC_RESET_RELEASE	Reset Release	ASYNC, SYNC	SYNC	Yes
INIT_DATA	Init Values Status	STATIC, DYNAMIC	STATIC	Yes

Pseudo Dual Port RAM (PDPW8KC) – EBR-Based

The Pseudo Dual Port RAM primitive is shown below.

Figure 12-45. Pseudo Dual-Port RAM (PDPW8KC)

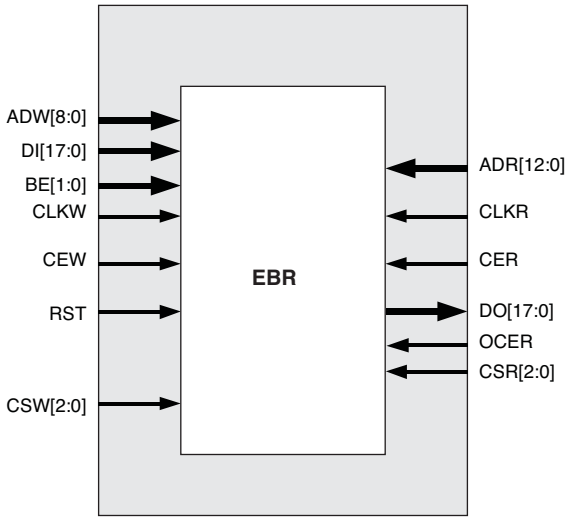


Table12-18. EBR-Based Pseudo-Dual Port Memory Port Definitions

Port Name in the EBR Block Primitive (PDPW8KC)	Description	Active State
ADW	Write Address	—
DI	Write Data	—
BE	Byte Enable	Active High
CLKW	Write Clock	Rising Edge
CEW	Write Clock Enable	Active High
RST	Reset	Active High
CSW	Write Chip Select	—
ADR	Read Address	—
CLKR	Read Clock	Rising Edge
CER	Read Clock Enable	Active High
DO	Read Data	—
OCER	Read Output Clock Enable	Active High
CSR	Read Chip Select	—

Each PDPW8KC primitive consists of 9,216 bits of RAM. The possible values for address depth and data width for the PDPW8KC primitive are listed in Table 12-19.

Table12-19. Pseudo-Dual Port Memory Sizes for 9K Memory in MachXO2

Pseudo-Dual Read Port Memory Size	Write Data Port	Read Data Port	Read Address Port [MSB:LSB]	Write Address Port [MSB:LSB]
8K x 1	DI[17:0]	DO	ADR[12:0]	ADW[8:0]
4K x 2	DI[17:0]	DO[1:0]	ADR[12:1]	ADW[8:0]
2K x 4	DI[17:0]	DO[3:0]	ADR[12:2]	ADW[8:0]
1K x 9	DI[17:0]	DO[8:0]	ADR[12:3]	ADW[8:0]
512 x 18	DI[17:0]	DO[17:0]*	ADR[12:4]	ADW[8:0]

Note: High and low bytes are swapped with regard to DI word.

Table 12-20 shows the various attributes available for the Pseudo Dual Port Memory (RAM_DP). Some of these attributes are user selectable through the IPexpress GUI. For detailed attribute definitions, refer to Appendix A.

Table12-20. Pseudo-Dual Port RAM Attributes for MachXO2 (PDPW8KC)

Attribute	Description	Values	Default Value	User Selectable through IPexpress
DATA_WIDTH_W	Write Data Word Width	18	18	Yes
DATA_WIDTH_R	Read Data Word Width	1, 2, 4, 9, 18	9	Yes
REGMODE	Register Mode (Pipelining)	NOREG, OUTREG	NOREG	Yes
RESETMODE	Selects the Reset type	ASYNC, SYNC	SYNC	Yes
CSDECODE_W	Chip Select Decode for Write	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	No
CSDECODE_R	Chip Select Decode for Read	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	No
GSR	Enables Global Set Reset	ENABLE, DISABLE	DISABLED	No
INITVAL_00 .. INITVAL_1F	Initialization Value	0x000000000000000000000000000000000000000000000000000000000000000000000000000000 .... 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFF (80- character hex strings)	0x000000000000000000000000000000000000000000000000000000000000000000000000000000	No
ASYNC_RESET_RELEASE	Reset Release	ASYNC, SYNC	SYNC	Yes
INIT_DATA	Init Values Status	STATIC, DYNAMIC	STATIC	Yes

Dual-Clock FIFO (FIFO8KB) – EBR Based

The Dual-Clock FIFO RAM primitive is shown below.

Figure 12-46. FIFO_DC Primitive (FIFO8KB)

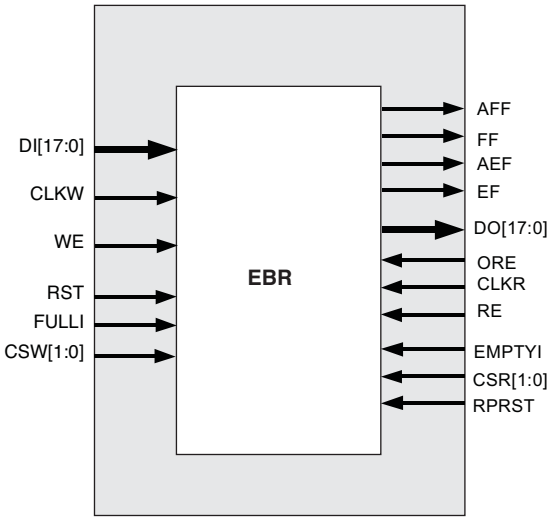


Table12-21. EBR-Based FIFO_DC Memory Port Definitions

Port Name in Primitive (FIFO8KB)	Description	Active State
DI	Data Input	—
CLKW	Write Port Clock	Rising Clock Edge
WE	Write Enable	Active High
FULLI	Write inhibit	Active High
CSW	Write Chip Select	Active High
AFF	Almost Full Flag	Active High
FF	Full Flag	Active High
AEF	Almost Empty	Active High
EF	Empty Flag	Active High
DO	Data Output	—
ORE	Output Read Enable	Active High
CLKR	Read Port Clock	Rising Clock Edge
RE	Read Enable	Active High
EMPTYI	Read inhibit	Active High
CSR	Read Chip Select	Active High
RPRST	Read Pointer Reset	Active High

Each FIFO8KB primitive consists of 9,216 bits of RAM. The possible values for address depth and data width for the FIFO8KB primitive are listed in Table 12-22.

Table12-22. MachXO2 FIFO_DC Data Widths Sizes

FIFO Size	Input Data	Output Data
8K x 1	DI	DO
4K x 2	DI[1:0]	DO[1:0]
2K x 4	DI[3:0]	DO[3:0]
1K x 9	DI[8:0]	DO[8:0]
512 x 18	DI[17:0]	DO[17:0]

Table 12-23 shows the various attributes available for the FIFO_DC. Some of these attributes are user-selectable through the IPexpress GUI. For detailed attribute definitions, refer to Appendix A.

Table12-23. FIFO_DC Attributes for MachXO2 (FIFO8KB)

Attribute	Description	Values	Default Value	User Selectable through IPexpress
DATA_WIDTH_W	Data Width Write Mode	1, 2, 4, 9, 18	18	YES
DATA_WIDTH_R	Data Width Read Mode	1, 2, 4, 9, 18	18	YES
REGMODE	Register Mode	NOREG, OUTREG	NOREG	YES
RESETMODE	Select Reset Type	ASYNC, SYNC	ASYNC	YES
CSDECODE_W	Chip Select Decode for Write Mode	0b00, 0b01, 0b10, 0b11	0b00	NO
CSDECODE_R	Chip Select Decode for Read Mode	0b00, 0b01, 0b10, 0b11	0b00	NO
GSR	Enable Global Set Reset	ENABLED, DISABLED	DISABLED	NO
AEPOINTER	Almost Empty Pointer	0b00000000000000, ..., 0b01111111111111	—	YES
AFPOINTER	Almost Full Pointer	0b00000000000000, ..., 0b01111111111111	—	YES
FULLPOINTER	Full Pointer	0b00000000000000, ..., 0b10000000000000	—	YES
FULLPOINTER1	Full Pointer minus 1	0b00000000000000, ..., 0b01111111111111	—	NO
AFPOINTER1	Almost Full Pointer minus 1	0b00000000000000, ..., 0b01111111111110	—	NO
AEPOINTER1	Almost Empty Pointer plus 1	0b00000000000000, ..., 0b10000000000000	—	NO
ASYNC_RESET_RELEASE	Reset Release	ASYNC, SYNC	SYNC	Yes

FIFO_DC Flags

The FIFO_DC have four flags available: Empty, Almost Empty, Almost Full and Full. Almost Empty, Almost Full and Full flags have a programmable range.

The program ranges for the four FIFO_DC flags are specified in Table 12-24.

Table12-24. FIFO_DC Flag Settings

Module Flag Name	FIFO Attribute Name	Description	Programming Range	Program Bits
Full	FULLPOINTER	Full setting	1 to (2N - 1)	14
	FULLPOINTER1	Full – 1	1 to (FULL-1)	14
AlmostFull	AFPOINTER	Almost full setting	1 to (FULL -1)	14
	AFPOINTER1	Almost full – 1	1 to (FULL -1)	14
	AEPOINTER1	Almost empty + 1	1 to (FULL -1)	14
AlmostEmpty	AEPOINTER	Almost empty setting	1 to (FULL -1)	14
Empty	—	Empty setting	0	—

The value of Empty is fixed at 0. When coming out of reset, the active high flags Empty and Almost Empty are set to high, since they are true.

Careful attention is required to set the Pointer attributes to match the desired behavior. Refer to Appendix B, Setting FIFO_DC Pointer Attributes.

The Empty and Almost Empty flags are always registered to the read clock and the Full and Almost Full flags are always registered to the write clock.

Distributed SPRAM (SPR16X4C) – PFU Based

The PFU based distributed single port RAM primitive is shown below.

Figure 12-47. Distributed_SPRAM Primitive (SPR16X4C)

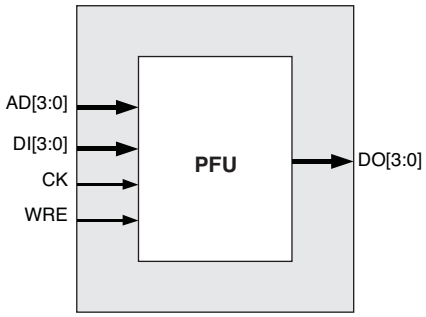


Table12-25. PFU based Distributed Single Port RAM Port Definitions

Port Name in the PFU Primitive	Description	Active State
AD[3:0]	Address	—
DI[3:0]	Data In	—
CK	Clock	Rising Clock Edge
WRE	Write Enable	Active High
DO[3:0]	Data Out	—

Distributed DPRAM (DPR16X4C) – PFU Based

The PFU based distributed Pseudo Dual-port RAM primitive is below.

Figure 12-48. Distributed DPRAM Primitive (DPR16X4C)

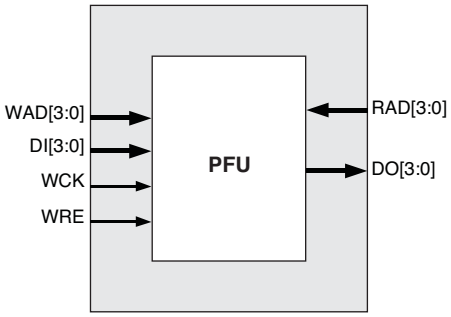


Table12-26. PFU based Distributed Dual-Port RAM Port Definitions

Port Name in the EBR Block Primitive	Description	Active State
WAD[3:0]	Write Address	—
DI[3:0]	Data Input	—
WCK	Write Clock	Rising Clock Edge
WRE	Write Enable	Active High
RAD[3:0]	Read Address	—
DO[3:0]	Data Out	—

Distributed ROM (ROMnnnX1A) – PFU Based

The PFU based distributed ROM primitives are shown below.

Figure 12-49. Distributed_ROM Primitive (ROM16X1A)

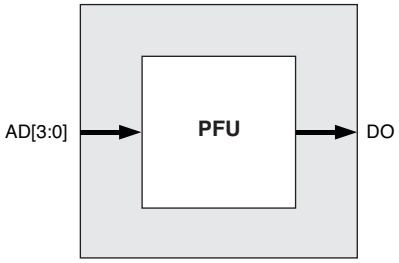


Figure 12-50. Distributed_ROM Primitive (ROM32X1A)

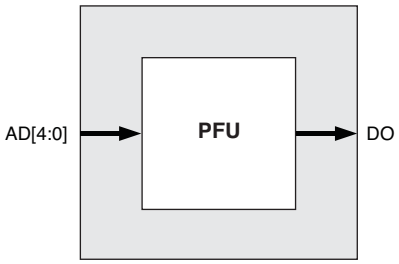


Figure 12-51. Distributed_ROM Primitive (ROM64X1A)

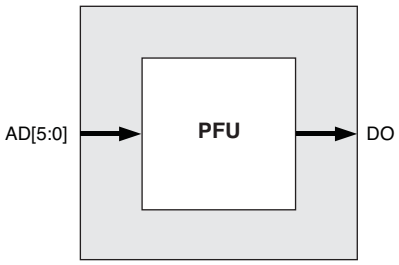


Figure 12-52. Distributed_ROM Primitive (ROM128X1A)

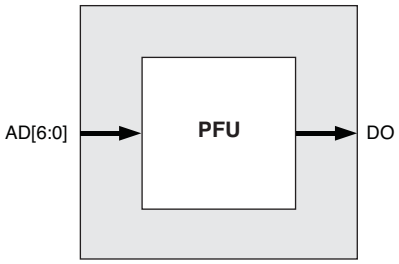


Figure 12-53. Distributed_ROM Primitive (ROM256X1A)

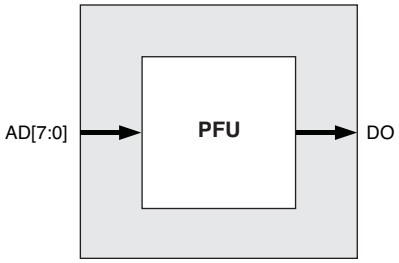


Table12-27. PFU-Based Distributed ROM Port Definitions

Port Name in the PFU Block Primitive	Description
AD[n:0]	Address
DO	Data Out

Initializing Memory

In the EBR based ROM or RAM memory modes and the PFU-based ROM memory mode, it is possible to specify the power-on state of each bit in the memory array. Each bit in the memory array can have one of two values: 0 or 1.

Initialization File Format

The initialization file is an ASCII file, which can be created or edited using any ASCII editor. IPexpress supports three different types of memory file formats:

1. Binary file
2. Hex File
3. Addressed Hex

The file name for the memory initialization file is *.mem (<file_name>.mem). Each row depicts the value to be stored in a particular memory location and the number of characters (or the number of columns) represents the number of bits for each address (or the width of the memory module).

The initialization file is primarily used for configuring the ROMs. The EBR in RAM can also use the initialization file to preload the memory contents.

Binary File
The file is a text file of 0's and 1's. The rows indicate the number of words and columns indicate the width of the memory.

```
Memory Size 20x32
00100000010000000010000001000000
00000001000000010000000100000001
00000010000000100000001000000010
00000011000000110000001100000011
00000100000001000000010000000100
00000101000001010000010100000101
00000110000001100000011000000110
00000111000001110000011100000111
00001000010010000000100001001000
00001001010010010000100101001001
00001010010010100000101001001010
00001011010010110000101101001011
00001100000011000000110000001100
00001101001011010000110100101101
00001110001111100000111000111110
00001111001111110000111100111111
00010000000100000001000000010000
00010001000100010001000100010001
00010010000100100001001000010010
00010011000100110001001100010011
```

Hex File
The Hex file is a text file of hex characters arranged in a similar row-column arrangement. The number of rows in the file is same as the number of address locations, with each row indicating the content of the memory location.

```
Memory Size 8x16
A001
0B03
1004
CE06
0007
040A
0017
02A4
```

Addressed Hex
Addressed Hex consists of lines of address and data. Each line starts with an address, followed by a colon, and any number of data. The format of memfile is address: data data data data ... where address and data are hexadecimal numbers.

```
–A0 : 03 F3 3E 4F
–B2 : 3B 9F
```

The first line puts 03 at address A0, F3 at address A1, 3E at address A2,and 4F at address A3. The second line puts 3B at address B2 and 9F at address B3.

There is no limitation on the values of address and data. The value range is automatically checked based on the values of addr_width and data_width. If there is an error in an address or data value, an error message is printed. Users need not specify data at all address locations. If data is not specified at certain address, the data at that location is initialized to 0. IPexpress makes memory initialization possible both through the synthesis and simulation flows.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
January 2011	01.1	Removed footnotes from the following figures: Top View of the MachXO2-1200 Device and Top View of the MachXO2-4000 Device.
February 2011	01.2	Updated document with new corporate logo.
		Document status changed from Advance to Final.

Appendix A. Attribute Definitions

DATA_WIDTH

Data width is associated with the RAM and FIFO elements. The DATA_WIDTH attribute defines the number of bits in each word. It takes the values defined in the RAM size tables in each memory module.

REGMODE

REGMODE, or the Register mode attribute, is used to enable pipelining in the memory. This attribute is associated with the RAM and FIFO elements. The REGMODE attribute takes the NOREG or OUTREG mode parameter that disables and enables the output pipeline registers.

RESETMODE

The RESETMODE attribute allows users to select the mode of reset in the memory. This attribute is associated with the block RAM elements. RESETMODE takes two parameters: SYNC and ASYNC. SYNC means that the memory reset is synchronized with the clock. ASYNC means that the memory reset is asynchronous to clock.

CSDECODE

CSDECODE, or the Chip Select Decode attributes, are associated to block RAM elements. Chip Select (CS) is a useful port when multiple cascaded EBR blocks are required by the memory. The CS signal forms the MSB for the address when multiple EBR blocks are cascaded. CS is a 3-bit bus, so it can cascade eight memories easily. CSDECODE takes the following parameters: “000”, “001”, “010”, “011”, “100”, “101”, “110”, and “111”. CSDECODE values determine the decoding value of CS[2:0]. CSDECODE_W is chip select decode for write and CSDECODE_R is chip select decode for read for Pseudo Dual Port RAM. CSDECODE_A and CSDECODE_B are used for true dual port RAM elements and refer to the A and B ports.

WRITEMODE

The WRITEMODE attribute is associated with the block RAM elements. It takes the NORMAL, WRITETHROUGH, and READBEFOREWRITE mode parameters.

In NORMAL mode, the output data does not change or get updated during the write operation. This mode is supported for all data widths.

In WRITETHROUGH mode, the output data is updated with the input data during the write cycle. This mode is supported for all data widths.

In READBEFOREWRITE mode, the output data port is updated with the existing data stored in the write address, during a write cycle. This mode is supported for x9 and x18 data widths.

WRITEMODE_A and WRITEMODE_B are used for dual port RAM elements and refer to the A and B ports in case of a True Dual Port RAM.

For all modes of the True Dual Port module, simultaneous read access from one port and write access from the other port to the same memory address is not recommended. The read data may be unknown in this situation. Also, simultaneous write access to the same address from both ports is not recommended. When this occurs, the data stored in the address becomes undetermined when one port tries to write a 'H' and the other tries to write a 'L'.

It is recommended that users implement control logic to identify this situation if it occurs and then either:

1. Implement status signals to flag the read data as possibly invalid, or
2. Implement control logic to prevent the simultaneous access from both ports.

GSR

GSR, the Global Set/ Reset attribute, is used to enable or disable the global set/reset for the RAM element.

ASYNC_RESET_RELEASE

When RESETMODE is set to ASYNC, the ASYNC_RESET_RELEASE attribute allows users to select how the reset is de-asserted/released: When set to SYNC, the reset is de-asserted synchronously to the clock. When set to ASYNC, the memory reset is released asynchronously (without relation to the clock).

INIT_DATA

The INIT_DATA attribute allows the user to specify how EBR initialization values are stored and accessed. When set to STATIC, the EBR initialization values are compressed by the software and stored in a variable location in UFM (User Flash Memory). When set to DYNAMIC, the initialization values are not compressed, and stored in a user-accessible, fixed location in UFM.

Appendix B. Setting FIFO_DC Pointer Attributes

The FIFO_DC uses pointer attributes to control the Full, Almost Full and Almost Empty flags.

The values for the pointer attributes are set according to the following table and equations:

Table12-28. Pointer Attribute Setting Equations

Flag	Trip Value	Attribute	Port Width	Equation
Full	ff	FULLPOINTER	wrw[1]	[(ff - 1) * wrw] + 1
		FULLPOINTER1		[(ff - 2) * wrw] + 1
Almost Full	aff	AFPOINTER	wrw[1]	[(aff - 1) * wrw] + 1
		AFPOINTER1		[(aff - 2) * wrw] + 1
Almost Empty	aef	AEPOINTER	rdw[1]	[(aef) * rdw] + rdw - 1
		AEFOINTER1		[(aef + 1) * rdw] + rdw - 1

1. Set Write Port Width (wrw) and Read Port Width (rdw) per Table 12-29.

Table12-29. Port Width Values

Attribute: Data_width_w, Data_width_r	Port Width: wrw, rdw
1	1
2	2
4	4
9	8
18	16

The user should specify the absolute value of the address at which the Almost Empty and Almost Full flags will go true. For example, if the Almost Full flag is required to go true at the address location 500 for a FIFO of depth 512, the user should specify aff = 500.

Worked Example:

Write Data Width: 18 => use wrw=16

Read Data Width: 4 => use rdw = 4

Full: ff =16 ((16) 18-bit words)

Almost Full: aff = 14

Almost Empty: aef = 8 ((8) 4-bit words, which corresponds to (2) 16-bit writes)

Empty: 0 (always)

Calculated Values:

FULLPOINTER = [(ff - 1) * wrw] + 1
= [(16 – 1) * 16] + 1
= (15 * 16) + 1
= 241
=> 14’ b00_0000_1111_0001

FULLPOINTER1 = [(ff - 2) * wrw] + 1
= [(16 – 2) * 16] + 1
= (14 * 16) + 1
= 225
=> 14’ b00_0000_1110_0001

AFPOINTER = [(aff - 1) * wrw] + 1
= [(14 – 1) * 16] + 1
= (13 * 16) + 1
= 209
=> 14’ b00_0000_1101_0001

AFPOINTER1 = [(aff - 2) * wrw] + 1
= [(14 – 2) * 16] + 1
= (12 * 16) + 1
= 193
=> 14’ b00_0000_1100_0001

AEPOINTER = [(aef) * rdw] + rdw - 1
= (8 * 4) + 4 - 1
= (8 * 4) + 3
= 35
=> 14’ b00_0000_0010_0011

AEFOINTER1 = [(aef + 1) * rdw] + rdw - 1
= [(8+1) * 4] + 4 - 1
= (9 * 4) + 3
= 39
=> 14’ b00_0000_0010_0111

MachXO2 sysCLOCK PLL Design and Usage Guide

September 2012 Technical Note TN1199

Introduction

MachXO2™ devices support a variety of I/O interfaces such as display interfaces (7:1 LVDS) and memory interfaces (LPDDR, DDR, DDR2). In order to support applications which use these interfaces, the MachXO2 device architecture has been designed to include advanced clocking features that are typically found in higher density FPGAs. These features provide designers the ability to synthesize clocks, minimize clock skew, improve performance and manage power consumption.

This technical note describes the clock resources available in the MachXO2 devices. Details are provided for primary clocks, edge clocks, clock dividers, sysCLOCK™ PLLs, DCC elements, the secondary high fan-out nets, and the internal oscillator available in the MachXO2 device.

The number of PLLs, edge clocks, and clock dividers for each MachXO2 device are listed in Table 13-1.

Table 13-1. Number of PLLs, Edge Clocks, and Clock Dividers

Parameter	Description	XO2-256	XO2-640	XO2-640U	XO2-1200	XO2-1200U	XO2-2000	XO2-2000U	XO2-4000	XO2-7000
Number of PLLs	General purpose PLLs	0	0	1	1	1	1	2	2	2
Number of edge clocks	Edge clocks for high-speed applications	0	0	4	4	4	4	4	4	4
Number of clock dividers	Clock dividers for DDR applications	0	0	4	4	4	4	4	4	4

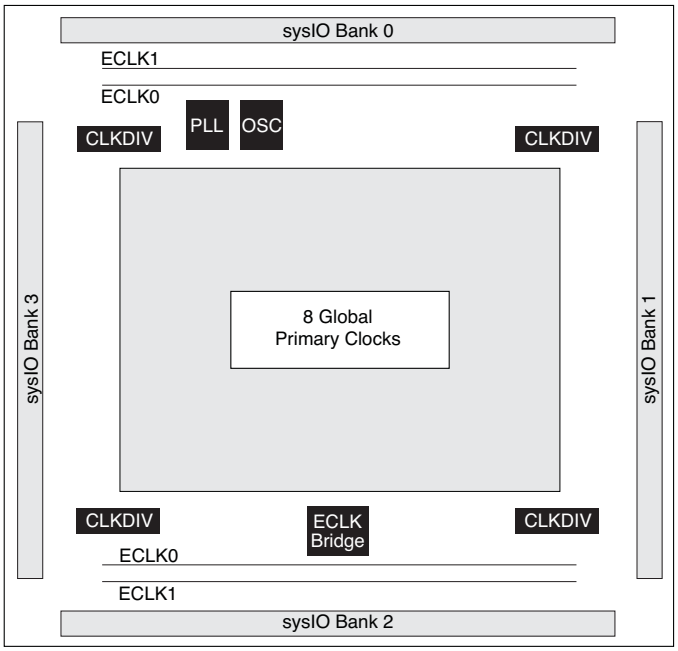
Clock/Control Distribution Network

MachXO2 devices provide global clock distribution in the form of eight global primary clocks and eight secondary high fan-out nets. Two edge clocks are provided on the top and bottom sides of the MachXO2-640U, MachXO2-1200/U and higher density devices. Other clock sources include clock input pins, internal nodes, PLLs, clock dividers, and the internal oscillator.

MachXO2 Top Level View

A top level view of the major clocking resources for the MachXO2-1200 device is shown in Figure 13-1.

Figure 13-1. MachXO2 Clocking Structure (MachXO2-1200)



Primary Clocks

The MachXO2 device has eight global primary clocks. The primary clock networks provide a low skew clock distribution path across the chip for high fan-out signals. Two of the primary clocks are equipped with a Dynamic Clock Mux (DCMA) feature that provides the ability to switch between two different clock sources.

The sources of the primary clocks are:

- Dedicated clock pins
- PLL outputs
- CLKDIV outputs
- Internal nodes

Dynamic Clock Mux (DCMA)

The MachXO2 devices have two Dynamic Clock Muxes (DCMA) that allow a design to dynamically switch between two independent primary clock signals. The output of the DCMA is to the primary clock distribution network. The inputs to the DCMA can be any of the clock sources available to the primary clock network.

The DCMA is a simple clock buffer with a multiplexer function. There is no synchronization of the clock signals when switching occurs so a glitch could occur.

DCMA Primitive Definition

The DCMA primitive can be instantiated in the source code of a design as defined in this section. Figure 13-2 and Table 13-2 show the DCMA definitions.

Figure 13-2. DCMA Primitive Symbol

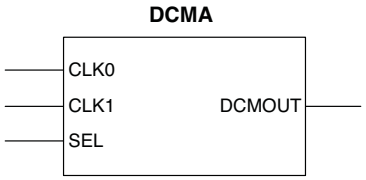


Table 13-2. DCMA Primitive Port Definition

Port Name	I/O	Description
CLK0	I	Clock input port zero – this the default
CLK1	I	Clock input port one
SEL	I	Select port - SEL=0 for CLK0 - SEL=1 for CLK1
DCMOUT	O	Clock output port

DCMA Declaration in VHDL Source Code

Library Instantiation

```
library machxo2;
use machxo2.all;
```

Component Declaration

```
component DCMA
port  CLK0     :     in std_logic;
      CLK1     :     in std_logic;
      SEL      :     in std_logic;
      DCMOUT   :     out std_logic);
end component;
```

DCMA Instantiation

```
I1: DCMA
port map (CLK0 => CLK0,
CLK1 => CLK1,
SEL => SEL,
DCMOUT => DCMOUT);
```

DCMA Usage with Verilog Source Code

Component Declaration

```
module DCMA (CLK0, CLK1, SEL, DCMOUT);

input CLK0;
input CLK1;
input SEL;
output DCMOUT;

endmodule
```

DCMA Instantiation

```
DCMA I1 (.CLK0 (CLK0),
         .CLK1 (CLK1),
         .SEL (SEL),
         .DCMOUT (DCMOUT));
```

Dynamic Clock Control (DCCA)

The MachXO2 devices have a dynamic clock control feature that is available for each of the primary clock networks. The Dynamic Clock Control (DCCA) allows each primary clock to be disabled from core logic if desired. Doing so disables a clock and its associated logic in the design when is it not needed and thus saves power.

DCCA Primitive Definition

The DCCA primitive can be instantiated in the source code of a design as defined in this section. Figure 13-3 and Table 13-3 show the DCMA definitions.

Figure 13-3. DCCA Primitive Symbol

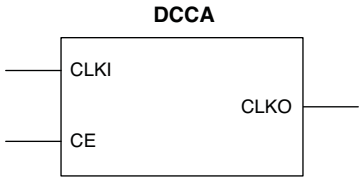


Table 13-3. DCCA Primitive Port Definition

Port Name	I/O	Description
CLKI	I	Clock input port
CE	I	Clock enable port - CE = 0 – disabled - CE = 1 – enabled
CLKO	O	Clock output port

DCCA Declaration in VHDL Source Code

Library Instantiation

```
library machxo2;
use machxo2.all;
```

Component Declaration

```
component DCCA
 port(CLKI     :     in std_logic;
      CE       :     in std_logic;
      CLKO     :     out std_logic);
end component;
```

DCCA Instantiation

```
I1: DCCA
 port map  (  CLKI =>  CLKI,
              CE   =>  CE,
              CLKO =>  CLKO);
end component;
```

DCCA Usage with Verilog Source Code

Component Declaration

```
module DCCA (CLKI, CE, CLKO);

   input CLKI;
   input CE;
   output CLKO;

endmodule
```

DCCA Instantiation

```
DCCA I1  (.CLKI (CLKI),
          .CE   (CE),
          .CLKO (CLKO));
```

Edge Clocks

There are two edge clock resources on the top and bottom sides of the MachXO2-640U, MachXO2-1200/U and higher density devices. These clocks, which have low injection time and skew, are used to clock I/O registers. Edge clock resources are designed for high-speed I/O interfaces with high fan-out capability. Refer to Appendix B for detailed information on the ECLK locations and connectivity.

The sources of edge clocks are:

- Dedicated clock pins
- PLL outputs
- Internal nodes

Edge Clock Bridge

The MachXO2-640U, MachXO2-1200/U and higher density devices also have an edge clock bridge that is used to enhance communication of ECLKs across the device. The bridge allows an input on the bottom of the device to drive the edge clock on the top edge of the device with minimal skew. Edge clock sources can either go through the edge clock bridge to connect to the edge clock or can be directly connected using the shortest path.

The Edge Clock Bridge is primarily intended for use with high-speed data interfaces such as DDR or 7:1 LVDS Video. For more information on the use of the Edge Clock Bridge please see TN1203, Implementing High-Speed Interfaces with MachXO2 Devices.

In the edge clock bridge there is a clock select mux that allows a design to switch between two different clock sources for each edge clock. This clock select mux is modeled using the ECLKBRIDGECS primitive. A block diagram of the edge clock bridge is shown in Appendix B.

ECLKBRIDGECS Primitive Definition

The ECLKBRIDGECS primitive can be instantiated in the source code of a design as defined in this section. A design can have up to two instantiations of ECLKBRIDGECS primitives if desired. Figure 13-4 and Table 13-4 show the ECLKBRIDGECS definitions.

Figure 13-4. ECLKBRIDGECS Primitive Symbol

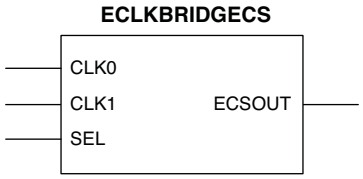


Table 13-4. ECLKBRIDGECS Primitive Port Definition

Port Name	I/O	Description
CLK0	I	Clock Input port zero – this the default.
CLK1	I	Clock Input port one
SEL	I	Select port - SEL = 0 for CLK0 - SEL=1 for CLK1
ECSOUT	O	Clock output port

ECLKBRIDGECS Declaration in VHDL Source Code

Library Instantiation

```
library machxo2;
use machxo2.all;
```

Component Declaration

```
component ECLKBRIDGECS
 port (  CLK0     :     in std_logic;
         CLK1     :     in std_logic;
         SEL      :     in std_logic;
         ECSOUT   :     out std_logic);
end component;
```

ECLKBRIDGECS Instantiation

```
I1: ECLKBRIDGECS
 port map (  CLK0    =>  CLK0,
             CLK1    =>  CLK1,
             SEL     =>  SEL,
             ECSOUT  =>  ECSOUT);
```

ECLKBRIDGECS Usage with Verilog Source Code

Component Declaration

```
module ECLKBRIDGECS (CLK0, CLK1, SEL, ECSOUT);

   input CLK0;
   input CLK1;
   input SEL;
   output ECSOUT;

endmodule
```

ECLKBRIDGECS Instantiation

```
ECLKBRIDGECS I1 ( .CLK0 (CLK0),
                  .CLK1 (CLK1),
                  .SEL   (SEL),
                  .ECSOUT (ECSOUT));
```

Edge Clock Synchronization (ECLKSYNCA)

MachXO2-640U, MachXO2-1200/U and higher density devices also have a dynamic edge clock synchronization control (ECLKSYNCA). This feature allows each edge clock to be disabled from core logic if desired. Designers can use this feature to synchronize the edge clock to an event or external signal if desired. Designers can also use this feature to design applications in which a clock and its associated logic can be dynamically disabled to save power. For the “R1” version of the MachXO2 devices ECLKSYNC may have a glitch in the output under certain conditions, leading to possible loss of synchronization. The “R1” versions of the MachXO2 devices have an “R1” suffix at the end of the part number (e.g., LCMXO2-1200ZE-1TG144CR1). For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

ECLKSYNCA Primitive Definition

The ECLKSYNCA primitive can be instantiated in the source code of a design as defined in this section. Figure 13-5 and Table 13-3 show the ECLKSYNCA definitions.

Figure 13-5. ECLKSYNCA Primitive Symbol

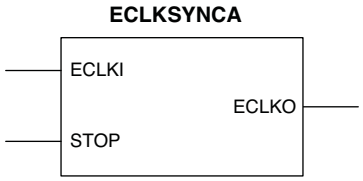


Table 13-5. ECLKSYNCA Primitive Port Definition

Port Name	I/O	Description
ECLKI	I	Clock input port
STOP	I	Control signal to stop edge clock - STOP=0 Clock is active - STOP=1 Clock is off
ECLKO	O	Clock output port

ECLKSYNCA Declaration in VHDL Source Code

Library Instantiation

```
library machxo2;
use machxo2.all;
```

Component Declaration

```
component ECLKSYNCA
port     ( ECLKI   :in std_logic;
           STOP    :in std_logic;
           ECLKO   :out std_logic);
end component;
```

ECLKSYNCA Instantiation

```
I1: ECLKSYNCA
port map (  ECLKI => ECLKI,
            STOP => STOP,
            ECLKO => ECLKO);
```

ECLKSYNCA Usage with Verilog Source Code

Component Declaration

```
module ECLKSYNCA (ECLKI,  STOP, ECLKO);

input ECLKI;
input STOP;
output ECLKO;

endmodule
```

ECLKSYNCA Instantiation

```
ECLKSYNCA I1    (.ECLKI (ECLKI),
                 .STOP (STOP),
                 .ECLKO (ECLKO));
```

Secondary High Fan-out Nets

MachXO2 devices have eight secondary high fan-out nets that can be used for clock, control, or high fan-out data signals. These secondary nets are a flexible global clocking resource with low injection delay and lower skew than the general routing resources. The secondary nets are primarily for global high fan-out control signals such as Clock Enables (CE), Local Set/Reset (LSR), and Reset (RST) applications. The eight secondary nets can have four independent control or clock inputs and four independent data inputs.

The sources of the secondary nets are:

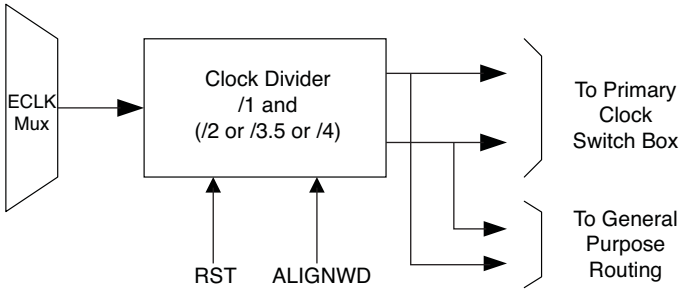
- Dedicated clock pins
- Internal nodes

Clock Dividers (CLKDIVC)

There are four clock dividers available in the MachXO2-640U, MachXO2-1200/U and higher density devices. The MachXO2-256 and MachXO2-640 devices do not have clock dividers. The clock divider provides two outputs. One is the same frequency as the input clock and the other is the input clock divided by either 2, 3.5, or 4. Both of the outputs have matched input-to-output delay. The input to the clock divider is the output from the edge clock mux. The outputs of the clock divider drive the primary clock network and are also available for general purpose routing or secondary clocks.

A block diagram of the clock divider is shown in Figure 13-6.

Figure 13-6. MachXO2 Clock Divider



CLKDIVC Primitive Definition

The CLKDIVC primitive can be instantiated in the source code of a design as defined in this section. Figure 13-7 and Tables 13-6 and 13-7 show the CLKDIVC definitions.

Figure 13-7. CLKDIVC Primitive Symbol

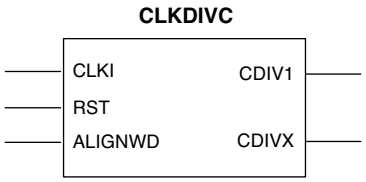


Table 13-6. CLKDIVC Primitive Port Definition

Port Name	I/O	Description
CLKI	I	Clock input
RST	I	Reset input - asynchronously forces all outputs low - RST = 0 Clock output outputs are active - RST = 1 Clock output outputs are OFF
ALIGNWD	I	Signal is used for word alignment. - ALIGNWD = 0 when not used See TN1203, Implementing High-Speed Interfaces with MachXO2 Devices for more information.
CDIV1	O	Divide by 1 output port
CDIVX	O	Divide by 2, 3.5 or 4 output port

Table 13-7. CLKDIVC Primitive Attribute Definition

Name	Description	Value	Default
GSR	GSR Enable	ENABLED, DISABLED	DISABLED
DIV	CLK Divider	2.0, 3.5 or 4.0	2.0

The ALIGNWD input is intended for use with high-speed data interfaces such as DDR or 7:1 LVDS video. For more information on the use of ALIGNWD please see TN1203, Implementing High-Speed Interfaces with MachXO2 Devices.

CLKDIVC Declaration in VHDL Source Code

Library Instantiation

```
library machxo2;
use machxo2.all;
```

Component and Attribute Declaration

```
component CLKDIVC
generic (DIV : string;
         GSR : string);
port  (RST     : in  std_logic;
      CLKI     : in  std_logic;
      ALIGNWD  : in  std_logic;
      CDIV1    : out std_logic;
      CDIVX    : out std_logic);
end component;
```

CLKDIVC Instantiation

```
I1: CLKDIVC
generic map ( DIV => "2.0",
              GSR => "DISABLED")
port map (    RST => RST,
              CLKI => CLKI,
              ALIGNWD => ALIGNWD,
              CDIV1 => CDIV1,
              CDIVX = > CDIVX);
```

CLKDIVC Usage with Verilog Source Code

Component and Attribute Declaration

```
module CLKDIVC (RST, CLKI, ALIGNWD, CDIV1, CDIVX);

parameter DIV = "2.0";        // "2.0", "3.5", "4.0"
parameter GSR = "DISABLED";   // "ENABLED", "DISABLED"

input RST;
input CLKI;
input ALIGNWD;
output CDIV1;
output CDIVX;

endmodule
```

CLKDIVC Instantiation

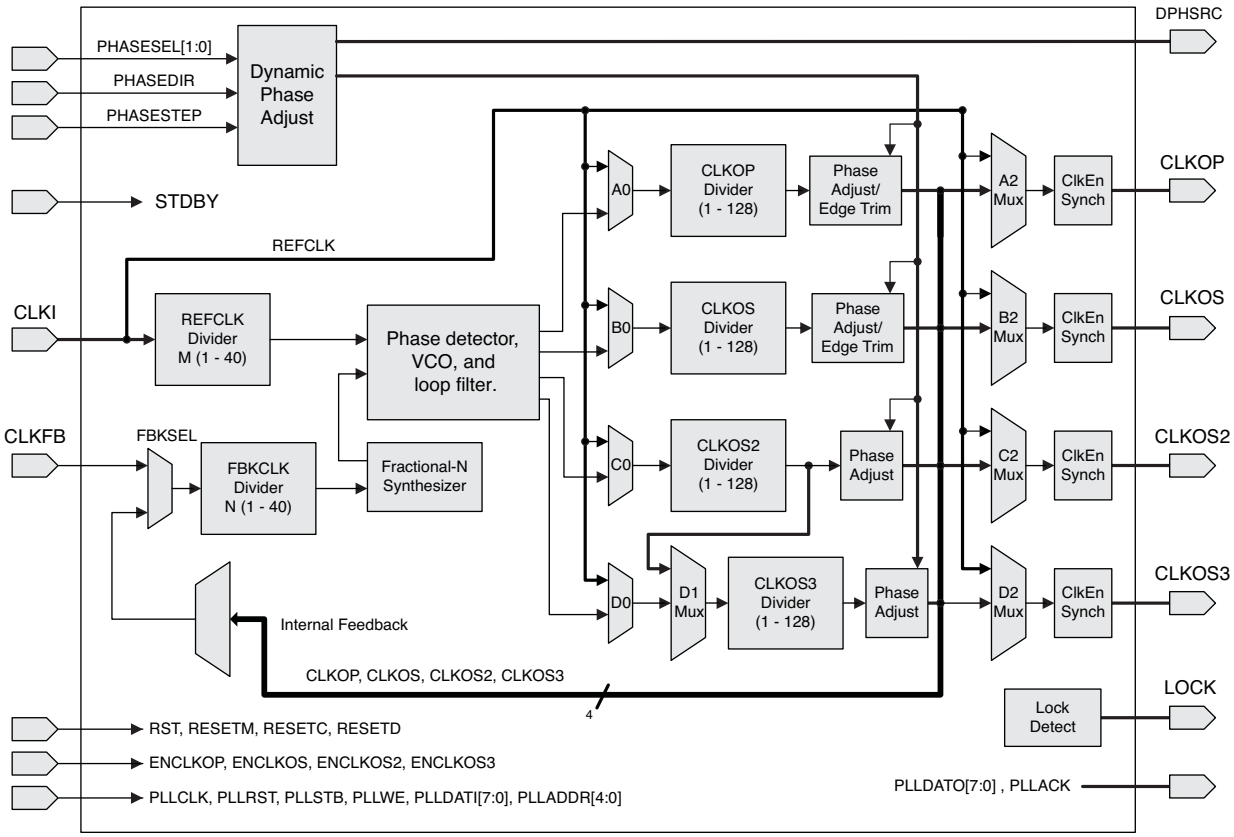
```
defparam I1.DIV = "2.0";
defparam I1.GSR = "DISABLED";

CLKDIVC I1 (   .RST (RST),
               .CLKI(CLKI),
               .ALIGNWD (ALIGNWD),
               .CDIV1 (CDIV1),
               .CDIVX (CDIVX));
```

sysCLOCK PLL

The MachXO2 PLL provides features such as clock injection delay removal, frequency synthesis, and phase adjustment. Figure 13-8 shows a block diagram of the MachXO2 PLL.

Figure 13-8. MachXO2 PLL Block Diagram



Functional Description

PLL Divider Blocks

Input Clock (CLKI) Divider: The CLKI divider is used to control the input clock frequency into the PLL block. The divider setting directly corresponds to the divisor of the output clock. The input must be within the input frequency range specified in the MachXO2 Family Data Sheet. The output of the input divider must also be within the phase detector frequency range specified in the data sheet.

Feedback Loop (CLKFB) Divider: The CLKFB divider is used to divide the feedback signal. Effectively, this multiplies the output clock because the divided feedback must speed up to match the input frequency into the PLL block. The PLL block increases the output frequency until the divided feedback frequency equals the input frequency. The output of the feedback divider must be within the phase detector frequency range specified in the MachXO2 Family Data Sheet.

Output Clock Dividers (CLKOP, CLKOS, CLKOS2, CLKOS3): The output clock dividers allow the VCO frequency to be scaled up to the 400-800 MHz range which minimizes jitter. Each of the output dividers is independent of the other dividers and each uses the VCO as the source by default. Each of the output dividers can be set to a value of 1 to 128. The CLKOS2 and CLKOS3 dividers can be cascaded together to produce a lower frequency output if desired.

Phase Adjustment (Static Mode): The CLKOP, CLKOS, CLKOS2, and CLKOS3 outputs can be phase adjusted relative to the input clock. The phase adjustments can be done in 45° steps. The clock output selected as the feedback cannot use the static phase adjustment feature.

Phase Adjustment (Dynamic Mode): The phase adjustments can also be controlled in a dynamic mode using the PHASESEL, PHASEDIR, and PHASESTEP ports. The clock output selected as the feedback cannot use the dynamic phase adjustment feature. Please see the Dynamic Phase Adjustment section of this document for more details.

Edge Trim Adjustment (Static Mode): The CLKOP and CLKOS ports can be finely tuned with an edge trim adjustment feature.

PLL Features

Standby Mode

The MachXO2 PLL contains a Standby mode that allows the PLL to be placed into a standby state to save power when not needed in the design. The PLL can be powered down completely or just partially depending on the needs of the design.

Fractional-N synthesis

The MachXO2 PLL contains a fractional-N synthesis feature which allows the user to generate an output clock which is a non-integer multiple of the input frequency. The user is allowed to enter a value between 0 and 65535 for the fractional-N divider. This value is then divided by 65536 and the result is added to the feedback divider. A MASH Delta-Sigma modulation technique is used such that the average effective feedback divide value is equal to this value. Fractional-N synthesis can be used to create a closer PPM match to the target frequency.

WISHBONE Ports

The MachXO2 PLL contains a WISHBONE port feature which allows the PLL settings to be dynamically changed from the user logic. When using this feature the EFB block must also be instantiated in the design to allow access to the WISHBONE ports. The WISHBONE ports of the PLL must be connected to the WISHBONE ports of the EFB block for proper simulation and operation. The use of the WISHBONE ports is described in detail in Appendix D.

PLL Inputs and Outputs

CLKI Input

The CLKI signal is the reference clock for the PLL. It must conform to the specifications in the data sheet in order for the PLL to operate correctly. The CLKI signal can come from a dedicated dual-purpose I/O pin, from any I/O pin, or from routing. The dedicated dual-purpose I/O pin provides a low skew input path and is the recommended source for the PLL. The reference clock will be divided by the input (M) divider to create one input to the phase detector of the PLL.

CLKFB Input

The CLKFB signal is the feedback signal to the PLL. The feedback signal is used by the PLL to determine if the output clock needs adjustment to maintain the correct frequency, phase, or other characteristic. The CLKFB signal can come from the primary clock net, from a dedicated dual-purpose I/O pin, directly from an output clock divider, or from routing. By using external feedback designers can compensate for board-level clock alignment. The feedback clock signal will be divided by the feedback (N) divider to create an input to the phase detector of the PLL. A bypassed PLL output cannot be used as the feedback signal.

RST Input

The PLL reset occurs under two conditions. At power-up an internal power-up reset signal from the configuration block resets the PLL. The user-controlled PLL reset signal RST can be provided as a part of the PLL module. The RST signal can be driven by an internally-generated reset function or by an I/O pin. This RST signal resets the PLL core (VCO, phase detector, and charge pump) and the output dividers which causes the outputs to be grounded, even in bypass mode.

After the RST signal is de-asserted the PLL will start the lock-in process and will take t_{LOCK} time to complete the PLL LOCK. Figure 13-9 shows the timing diagram of the RST input. The RST signal is active high. The RST signal is optional.

The RST input does NOT reset the input divider (M-divider). The reason for not resetting the M-divider is that there may be a clock used externally that is a synchronized to the reference clock. In this case there is a state relationship between the external clock and the M-divided clock (which the PLL is synchronized to). This relationship needs to be preserved by the user when resetting the PLL. In this condition, RST will be used to reset the PLL without resetting the M-divider.

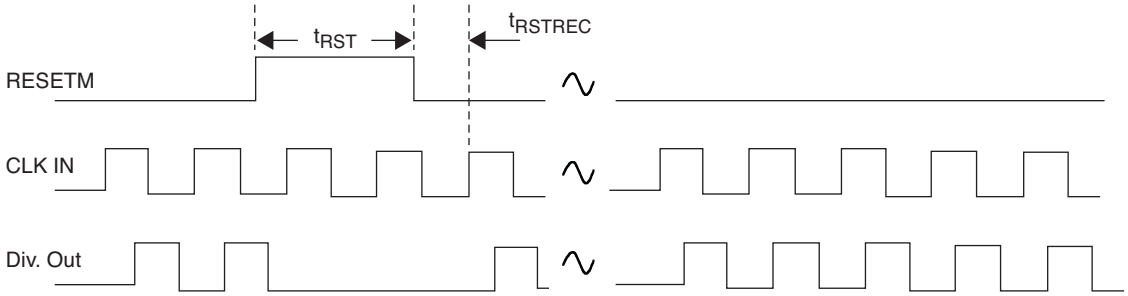
RESETM Input

The user-controlled PLL reset signal RESETM can be provided as a part of the PLL module. The RESETM signal can be driven by an internally-generated reset function or by an I/O pin. The RESETM signal resets the PLL core (similar to RST) and the all the dividers, including the M-divider. This causes the outputs to be grounded, including when the PLL is in bypass mode.

After the RESETM signal is de-asserted the PLL will start the lock-in process and will take t_{LOCK} time to complete the PLL LOCK. Figure 13-9 shows the timing diagram of the RESETM input. The RESETM signal is active high. The RESETM signal is optional.

If the user wishes to synchronize the PLL output to an external clock source the RESETM signal can be used to reset the PLL.

Figure 13-9. RST and RESETM Timing Diagram



RESETC Input

The user-controlled PLL reset signal RESETC can be provided as a part of the PLL module. The RESETC signal can be driven by an internally-generated reset function or by an I/O pin. This RESETC signal resets only the CLKOS2 output divider. This causes the CLKOS2 output to be grounded unless the output is in the bypass mode. If this output is in bypass mode as a clock divider it will be reset by the RESETC signal. The RESETC signal can be used to synchronize the CLKOS2 output to an external clock signal.

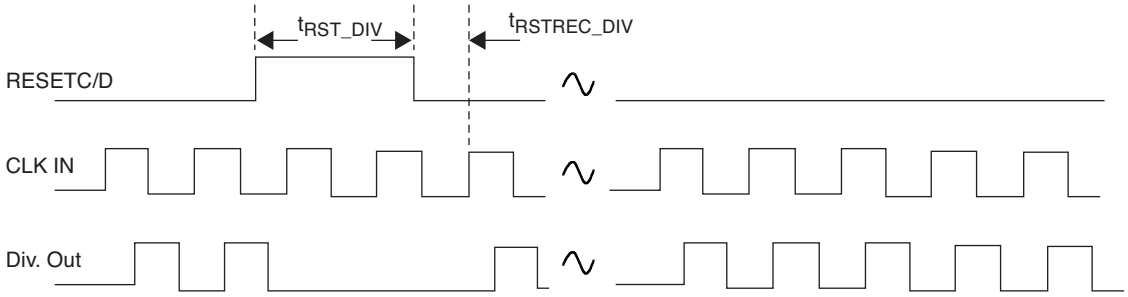
After the RESETC signal is de-asserted there is a time delay of t_{RSTREC_DIV} time before the next clock edge will toggle the CLKOS2 output divider. Figure 13-10 shows the timing diagram of the RESETC input. The RESETC signal will not affect the PLL loop unless the CLKOS2 output is used in the feedback path. If the CLKOS2 output is used in the feedback path it is recommended to use the RST or RESETM signal to reset the PLL rather than RESETC. The RESETC signal is active high. The RESETC signal is optional.

RESETD Input

The user-controlled PLL reset signal RESETD can be provided as a part of the PLL module. The RESETD signal can be driven by an internally-generated reset function or by an I/O pin. This RESETD signal resets only the CLKOS3 output divider. This causes the CLKOS3 output to be grounded unless the output is in the bypass mode. If this output is in bypass mode as a clock divider it will be reset by the RESETD signal. The RESETD signal can be used to synchronize the CLKOS3 output to an external clock signal.

After the RESETD signal is de-asserted there is a time delay of t_{RSTREC_DIV} time before the next clock edge will toggle the CLKOS3 output divider. Figure 13-10 shows the timing diagram of the RESETD input. The RESETD signal will not affect the PLL loop unless the CLKOS3 output is used in the feedback path. If the CLKOS3 output is used in the feedback path it is recommended to use the RST or RESETM signal to reset the PLL rather than RESETD. The RESETD signal is active high. The RESETD signal is optional.

Figure 13-10. RESETC and RESETD Timing Diagram



ENCLKOP Input
The ENCLKOP signal is used to enable and disable the CLKOP output from a user signal. This enables designers to save power by stopping the CLKOP output when it is not used. Additionally this signal also allows the designer to synchronize CLKOP with another signal in the design. The ENCLKOP signal is optional and will only be available if the user has selected the clock enable ports option in IPexpress™. If the ENCLKOP signal is not requested the CLKOP output will be active at all times (when the PLL is instantiated) unless the PLL is placed into the standby mode. The ENCLKOP signal is active high.

ENCLKOS Input
The ENCLKOS signal is used to enable and disable the CLKOS output from a user signal. This enables designers to save power by stopping the CLKOS output when it is not used. Additionally this signal also allows the designer to synchronize CLKOS with another signal in the design. The ENCLKOS signal is optional and will only be available when the PLL is configured with the CLKOS output and the Clock Enable ports options in IPexpress. If the PLL is configured with the CLKOS output enabled and the ENCLKOS signal is not requested the CLKOS output will always be active unless the PLL is placed into the standby mode. The ENCLKOS signal is active high.

ENCLKOS2 Input
The ENCLKOS2 signal is used to enable and disable the CLKOS2 output from a user signal. This enables designers to save power by stopping the CLKOS2 output when it is not used. Additionally this signal also allows the designer to synchronize CLKOS2 with another signal in the design. The ENCLKOS2 signal is optional and will only be available when the PLL is configured with the CLKOS2 output and the Clock Enable ports options in IPexpress. If the PLL is configured with the CLKOS2 output enabled and the ENCLKOS2 signal is not requested the CLKOS2 output will always be active unless the PLL is placed into the standby mode. The ENCLKOS2 signal is active high.

ENCLKOS3 Input
The ENCLKOS3 signal is used to enable and disable the CLKOS2 output from a user signal. This enables designers to save power by stopping the CLKOS3 output when it is not used. Additionally this signal also allows the designer to synchronize CLKOS3 with another signal in the design. The ENCLKOS3 signal is optional and will only be available when the PLL is configured with the CLKOS3 output and the Clock Enable ports options in IPexpress. If the ENCLKOS3 signal is not requested the CLKOS3 output will always be active unless the PLL is placed into the standby mode. The ENCLKOS3 signal is active high.

STDBY Input
The STDBY signal is used to put the PLL into a low power standby mode when it is not required. The STDBY port can be connected to the power controller so that the PLL will enter the low power state when device is driven to the Standby mode. Alternatively the STDBY port can be driven by user logic independent of the standby mode. The

STDBY signal is optional and will only be available if the user has selected the Standby ports option in IPexpress. The STDBY signal is active high.

PHASESEL Input
The PHASESEL[1:0] input is used to specify which PLL output port will be affected by the dynamic phase adjustment ports. The settings available are shown in the Dynamic Phase Adjustment section of this document. The PHASESEL signal must be stable before the PHASESTEP signal is toggled. The PHASESEL signal is optional and will only be available if the user has selected the Dynamic Phase ports option in IPexpress.

PHASEDIR Input
The PHASEDIR input is used to specify which direction the dynamic phase shift will occur, advanced (leading) or delayed (lagging). When PHASEDIR = 0 then the phase shift will be delayed from the current clock by one step. When PHASEDIR = 1 then the phase shift will be advanced from the current clock by one step. The PHASEDIR signal must be stable before the PHASESTEP signal is toggled.

The PHASEDIR signal is optional and will only be available if the user has selected the Dynamic Phase ports option in IPexpress.

PHASESTEP Input
The PHASESTEP signal is used to initiate the dynamic phase adjustment for the clock output port and in the direction specified by the PHASESEL and PHASEDIR inputs respectively. The PHASESTEP signal is optional and will only be available if the user has selected the Dynamic Phase ports option in IPexpress.

CLKOP Output
CLKOP is the main clock output of the sysCLOCK PLL. This signal is always available by default and can be routed to the primary clock network of the chip. The CLKOP output can also be routed to top/bottom edge clocks. The CLKOP output can be phase-shifted either statically or dynamically and can also be used with the duty trim adjustment feature. The CLKOP signal output can either come from the CLKOP output divider or can bypass the PLL. When CLKOP is in the bypass mode the output divider can either be bypassed or used in the circuit.

CLKOS Output
The secondary clock output of the sysCLOCK PLL is the CLKOS signal. This signal is available when selected by the user and can be routed to the primary clock network of the device. The CLKOS output can also be routed to top and bottom edge clocks. The CLKOS output can be phase-shifted either statically or dynamically and can also be used with the duty trim adjustment feature. The CLKOS signal output can either come from the CLKOS output divider or can bypass the PLL. When CLKOS is in the bypass mode the output divider can either be bypassed or used in the circuit. The CLKOS signal is optional.

CLKOS2 Output
The CLKOS2 signal is another secondary clock output that is available in the sysCLOCK PLL. This signal is available when selected by the user and can be routed to the primary clock network of the chip. The CLKOS2 output cannot be routed to top and bottom edge clocks. The CLKOS2 output can be phase-shifted either statically or dynamically but does not have the duty trim adjustment feature. The CLKOS2 signal output can either come from the CLKOS2 output divider or can bypass the PLL. When CLKOS2 is in the bypass mode the output divider can either be bypassed or used in the circuit. The CLKOS2 signal is optional.

CLKOS3 Output
The CLKOS3 signal is another secondary clock output that is available in the sysCLOCK PLL. This signal is available when selected by the user and can be routed to the primary clock network of the chip. The CLKOS3 output cannot be routed to top/bottom edge clocks. The CLKOS3 output can be phase-shifted either statically or dynamically but does not have the duty trim adjustment feature. The CLKOS3 signal output can either come from the CLKOS3 output divider or can bypass the PLL. When CLKOS3 is in the bypass mode the output divider can either be bypassed or used in the circuit. The CLKOS3 signal is optional.

The CLKOS3 output also supports lower frequency outputs that require an output divider value larger than 128. This is accomplished by cascading the CLKOS2 and CLKOS3 output dividers. When used in this application the CLKOS2 output cannot be used as an independent clock output. A cascaded clock output cannot be used for the feedback signal of the PLL.

DPHSRC Output
The DPHSRC output is used to indicate whether the dynamic phase ports or the WISHBONE registers are being used for control of the dynamic phase adjustment feature. The dynamic phase ports are the PHASESEL, PHASEDIR, and PHASESTEP ports. The DPHSRC signal is optional and will be available if the user has selected the Dynamic Phase ports option in IPexpress. If the user has not selected the Dynamic Phase ports option the WISHBONE registers will be used to set the dynamic phase adjustment feature by default.

LOCK Output
The LOCK output provides information about the status of the PLL. After the device is powered up and the input clock is valid, the PLL will achieve lock within the specified lock time. Once lock is achieved, the PLL LOCK signal will be asserted. The LOCK can either be in the Normal Lock mode or the Sticky Lock mode. In the Normal Lock mode, the LOCK signal is asserted when the PLL determines it has achieved lock and de-asserted if a loss of lock is detected. In Sticky Lock mode, once the LOCK signal is asserted it will stay asserted until the PLL reset is asserted or until the PLL is powered down. It is recommended to assert PLL RST to re-synchronize the PLL to the reference clock when the PLL loses lock. The LOCK signal is available to the FPGA routing to implement the generation of the RST signal if requested by the designer. The LOCK signal is optional and will be available if the user has selected the Provide PLL Lock signal option in IPexpress.

For the “R1” version of the MachXO2 devices, the PLL Lock signal will glitch high when coming out of standby. This glitch lasts for about 10µsec before returning low. The “R1” versions of the MachXO2 devices have an “R1” suffix at the end of the part number (e.g., LCMXO2-1200ZE-1TG144CR1). For more details on the R1 to Standard migration, refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

WISHBONE Ports
The WISHBONE parts are listed in Appendix D along with the description of how to use them. The WISHBONE ports are optional.

PLL Attributes

The PLL utilizes several attributes that allow the configuration of the PLL through source constraints and a preference file. The following section details these attributes and their usage.

FIN
The input frequency can be any value within the specified frequency range based upon the divider settings.

CLKI_DIV, CLKFB_DIV, CLKOP_DIV, CLKOS_DIV, CLKOS2_DIV, CLKOS3_DIV
These dividers determine the output frequencies of each of the output clocks. The user is not allowed to input an invalid combination when using IPexpress. Valid combinations are determined by the input frequency, the dividers, and the PLL specifications.

The CLKOP_DIV value is calculated to maximize the FVCO within the specified range based upon the FIN and CLKOP_FREQ in conjunction with the CLKI_DIV and CLKFB_DIV values. This applies when the CLKOP output is

used for the feedback signal. If another output is used for the feedback signal then the corresponding output divider shall be calculated in this manner.

The output signals that are not used for the feedback signal will use an output divider value based upon the VCO frequency and desired output frequency. The possible divider values for all these dividers are 1 to 128, though in some cases the full range is not allowed since it would violate the PLL specifications.

FREQUENCY_PIN_CLKI, FREQUENCY_PIN_CLKOP, FREQUENCY_PIN_CLKOS, FREQUENCY_PIN_CLOS2, FREQUENCY_PIN_CLKOS3
These input and output clock frequencies determine the divider values.

Frequency Tolerance – CLKOP, CLKOS, CLKOS2, CLKOS3
When the desired output frequency is not achievable, users may enter the frequency tolerance of the clock output.

MachXO2 PLL Primitive Definition

The PLL primitive can be instantiated in the source code of a design as defined in this section. Figure 13-11 and Table 13-8 show the EHXPLLJ definitions.

Figure 13-11. PLL Primitive Symbol

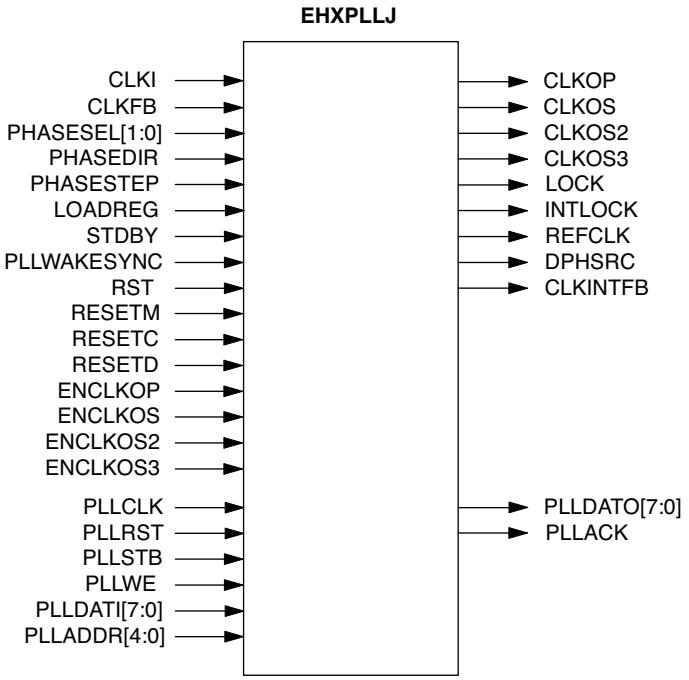


Table 13-8. PLL Primitive Port Definition

Port Name	I/O	Description
CLKI	I	Input clock to PLL
CLKFB	I	Feedback clock
PHASESEL[1:0]	I	Select which output is affected by Dynamic Phase adjustment ports
PHASEDIR	I	Dynamic Phase adjustment direction.
PHASESTEP	I	Dynamic Phase step – toggle shifts VCO phase adjust by one step
LOADREG	I	Dynamic Phase Load – toggle loads divider phase adjustment values into PLL
CLKOP	O	Primary PLL output clock (with phase shift adjustment)
CLKOS	O	Secondary PLL output clock (with phase shift adjust)
CLKOS2	O	Secondary PLL output clock2 (with phase shift adjust)

Table 13-8. PLL Primitive Port Definition (Continued)

Port Name	I/O	Description
CLKOS3	O	Secondary PLL output clock3 (with phase shift adjust)
LOCK	O	PLL LOCK, asynchronous signal. Active high indicates PLL is locked to input and feedback signals.
INTLOCK	O	PLL internal LOCK, asynchronous signal. Active high indicates PLL lock using internal feedback.[1]
REFCLK	O	Output of reference clock mux
DPHSRC	O	Dynamic phase source – ports or WISHBONE is active
STDBY	I	Standby signal to power down the PLL
PLLWAKESYNC	I	PLL wake-up sync. Enable PLL to switch from internal to user feedback path when the PLL wakes up.[1]
RST	I	PLL Reset without resetting the M-divider. Active high reset.
RESETM	I	PLL Reset - includes resetting the M-divider. Active high reset.
RESETC	I	Reset for CLKOS2 output divider only. Active high reset.
RESETD	I	Reset for CLKOS3 output divider only. Active high reset.
ENCLKOP	I	Clock Enable for CLKOP output
ENCLKOS	I	Clock Enable for CLKOS output - only available if CLKOS port is active
ENCLKOS2	I	Clock Enable for CLKOS2 output - only available if CLKOS2 port is active
ENCLKOS3	I	Clock Enable for CLKOS3 output - only available if CLKOS3 port is active
PLLCLK	I	PLL data bus clock input signal
PLLRST	I	PLL data bus reset. This resets only the data bus not any register values.
PLLSTB	I	PLL data bus strobe signal
PLLWE	I	PLL data bus write enable signal
PLLADDR [4:0]	I	PLL data bus address
PLLDATI [7:0]	I	PLL data bus data input
PLLDATO [7:0]	O	PLL data bus data output
PLLACK	O	PLL data bus acknowledge signal

1. The PLLWAKWSYNC and INTLOCK primitive ports are not brought out to the module level when IPexpress is used to generate the PLL. The ports are tied off in the module. Testing indicated that using these ports did not have a significant benefit.

Dynamic Phase Adjustment

The MachXO2 PLL supports dynamic phase adjustments through either the dynamic phase adjust ports or the WISHBONE interface using the following method. The WISHBONE interface is covered in more detail in Appendix D.

To use the dynamic phase adjustment feature the PHASESEL[1:0], PHASEDIR, PHASESTEP ports/signals are used. The DPHSRC port is also available and can be used to confirm that the correct signal source, the primitive ports or WISHBONE signals, has been selected prior to implementing the phase adjustment. The default setting when the dynamic phase ports are selected is to use the primitive ports for dynamic phase adjustments. The source for the dynamic phase adjustments can also be changed from the WISHBONE interface if desired using the MC1_DYN_SOURCE WISHBONE register. If the user does not select the dynamic phase ports from the GUI interface then the WISHBONE signals will be used for dynamic phase adjustments.

All four output clocks, CLKOP, CLKOS, CLKOS2, and CLKOS3, have the dynamic phase adjustment feature but only one output clock can be adjusted at a time. Table 13-9 shows the output clock selection settings available using the PHASESEL[1:0] signal. The PHASESEL signal must be stable before the PHASESTEP signal is toggled.

Table 13-9. PHASESEL Signal Settings Definitions

PHASESEL[1:0]	PLL Output Shifted
00	CLKOS
01	CLKOS2
10	CLKOS3
11	CLKOP

The selected output clock phase will either be advanced or delayed depending upon the value of the PHASEDIR port or signal. Table 13-10 shows the PHASEDIR settings available. The PHASEDIR signal must be stable before the PHASESTEP signal is toggled.

Table 13-10. PHASEDIR Signal Settings Definitions

PHASEDIR	Direction
00	Delayed (lagging)
01	Advanced (leading)

Once the PHASESEL and PHASEDIR have been set the phase adjustment is made by toggling the PHASESTEP signal. Each pulse of the PHASESTEP signal will generate a phase shift of one step. The PHASESTEP signal pulse must be initiated from a logic zero value and the phase shift will be initiated on the negative edge of the PHASESTEP signal. The step size is specified in the equation below.

Step size = 45° / Output Divider

If the phase shift desired is larger that 1 step the PHASESTEP signal can be pulsed several times to generate the desired phase shift. One step size is the smallest phase shift that can be generated by the PLL. The dynamic phase adjustment results in a glitch-free adjustment when delaying the output clock but glitches may result when advancing the output clock.

Frequency Calculation

The PLL can be used to synthesize a clock frequency that is needed in a design when the user's board does not have the necessary frequency source. The synthesized frequency can be calculated using the equations listed below.

$$f_{OUT} = f_{IN} * N/M \quad (1)$$

$$f_{VCO} = f_{OUT} * V \quad (2)$$

$$f_{PFD} = f_{IN} / M = f_{FB} / N \quad (3)$$

Where:

f_{OUT} is the output frequency.

f_{IN} is the input frequency.

f_{VCO} is the VCO frequency.

f_{PFD} is the PFD (Phase detector) Frequency.

f_{FB} is the Feedback signal Frequency.

N is the feedback divider (integer value shown in the IPexpress GUI).

M is the input divider (integer value shown in the IPexpress GUI).

V is the output divider (integer value shown in the IPexpress GUI).

These equations hold true for the clock output signal that is used for the feedback source to the PLL. Once the VCO frequency has been calculated from these equations, it can be used to calculate the remaining output clock signals using equation (2) above.

The equations listed above are valid provided that the divider value used for the output and feedback paths are equivalent. If they are not then the equation (1) becomes more complex because the two dividers must be included.

Fractional-N Synthesis Operation

The MachXO2 sysCLOCK PLLs support high resolution (16-bit) fractional-N synthesis. Fractional-N frequency synthesis allows the user to generate an output clock which is a non-integer multiple of the input frequency. The Fractional-N synthesis option is enabled in the IPexpress GUI by checking the Enable box under the Fractional-N Divider heading and then entering a number between 0 and 65535 into the adjacent box. The value which is entered in to the box will be divided by 65536 to form the fractional part of the feedback divider (also called the N divider) value. The effective feedback divider value is given by the equation:

$$Neff = N + (F/65536) \quad (4)$$

Where:

N is the integer Feedback divider (shown in the IPexpress GUI).

F is the value entered into the Fractional-N synthesis box described above.

The output frequency is given by the equation:

$$f_{OUT} = (f_{IN}/M) * Neff \quad (5)$$

Where:

f_{OUT} is the output frequency.

f_{IN} is the input frequency.

M is the input divider (shown in the IPexpress GUI).

The Fractional-N synthesis works by using a delta-sigma technique to approximate the fractional value that was entered by the user. Therefore, using the Fractional-N synthesis option will result in higher jitter of the PLL VCO and output clocks compared to using an integer value for the feedback divider. It is recommended that Fractional-N synthesis only be used if the N/M divider ratio is 4 or larger to prevent impacting the PLL jitter performance excessively. Fractional N jitter numbers can be found in the MachXO2 Family Data Sheet.

Low Power Features

The MachXO2 PLL contains several features that enable designers to minimize the power consumption of a design. These include dynamic clock enable and support for the standby mode.

Dynamic Clock Enable

The dynamic clock enable feature allows designers to turn off selected output clocks during periods when they are not used in the design. To support this feature, each output clock has an independent output enable signal that can be selected. The output enable signals are ENCLKOP, ENCLKOS, ENCLKOS2, and ENCLKOS3. When the Clock Enable Ports option is selected in the IPexpress GUI the output enable signal will be brought out to the top level ports of the PLL module for the CLKOP port and any other ports that are enabled in the IPexpress GUI.

If an output is not enabled in the IPexpress GUI, the ports for that selected output signal will not be present in the module and that output will be inactive.

Standby Mode

In order to minimize power consumption, the PLL can be shut down when it is not required by the application. The PLL can then be restarted when it is needed again and, after a short delay to allow the PLL to lock to the feedback signal, the output clocks will be reactivated. To support this mode the Standby Ports option is selected in the IPexpress GUI. This will cause the STDBY signal to be brought out to the top level of the PLL module. Placing the PLL into the Standby mode powers down the PLL and will cause all the outputs to be disabled.

The PLL will enter the Standby mode when the STDBY signal is driven high and the outputs will be driven low. The STDBY port can be connected to the power controller so that the PLL will enter the low power state when device is driven to the Standby mode. Alternatively the STDBY port can be driven by user logic independent of the Standby mode.

The PLL will wake-up from the Standby mode when the STDBY signal is driven low. When waking up from Standby mode the PLL will automatically lock to the external feedback signal that was originally selected prior to entering Standby mode. The PLL will lock to the external feedback signal after a maximum time delay of t_{LOCK}. When the PLL achieves lock to the external feedback signal the LOCK signal will be asserted high to indicate that it has locked.

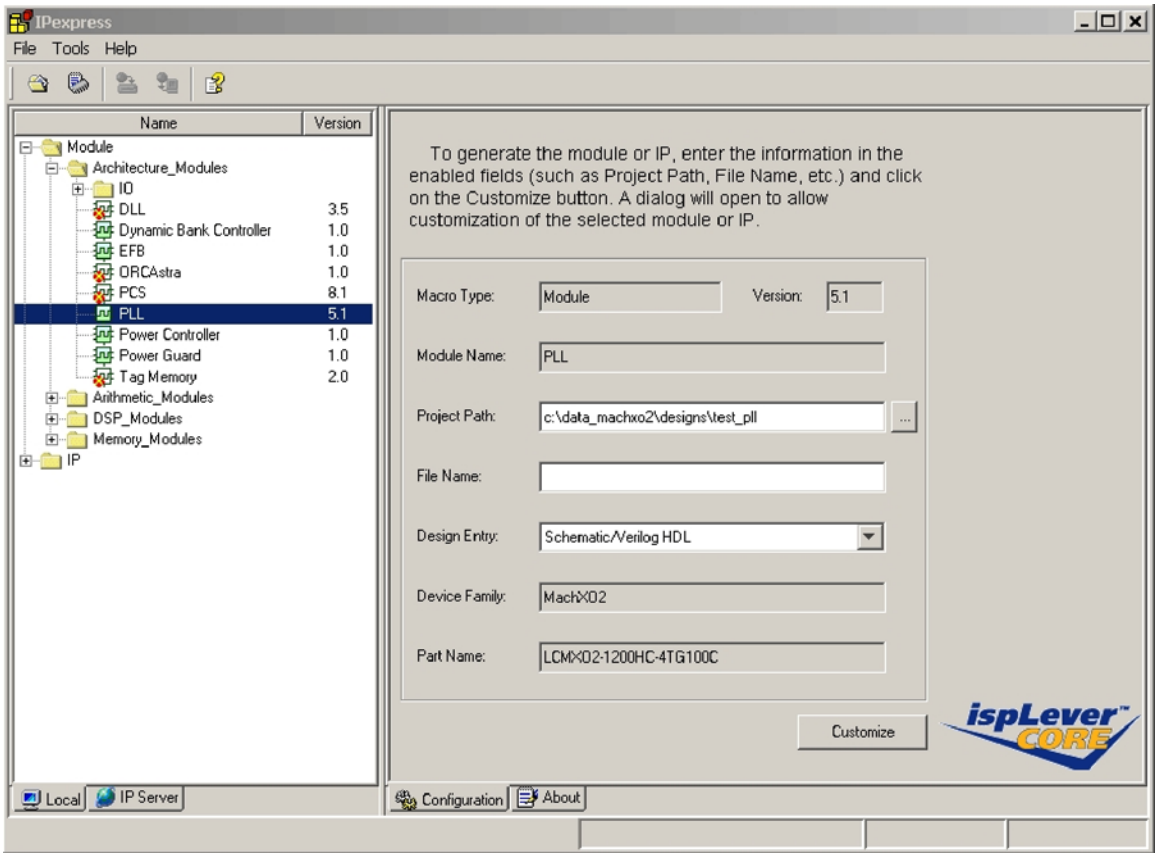
Configuring the PLL Using IPexpress

IPexpress is used to create and configure a PLL. Designers can select the parameters for the PLL using the graphical user interface. This process results in an HDL model that is used in the simulation and synthesis flow.

Figure 13-12 shows the main window when the PLL is selected in IPexpress from ispLEVER. For an example of the equivalent screen in Lattice Diamond®, see Figure 13-24 in Appendix E. When IPexpress is opened from within the ispLEVER Project Navigator or from Diamond, the project settings are automatically filled in for the you. The only entry required when using ispLEVER is the file name. When using Diamond, the file name and module output type (VHDL or Verilog) must be entered.

If IPexpress is opened as a stand-alone tool then it is necessary to supply the additional parameters shown on this screen. After entering the module name of choice, clicking on the **Customize** button will open the Configuration tab window as shown in Figure 13-13.

Figure 13-12. IPexpress Main Window for PLL Module



Configuration Tab

The configuration tab lists all user-accessible attributes with default values set. Upon completion, clicking on the **Generate** button will generate the source.

Configuration Modes

There are two modes that can be used to configure the PLL in the Configuration Tab: Frequency Mode and Divider Mode.

- **Frequency Mode**: In this mode the user enters the input and output clock frequencies and IPexpress calculates the divider settings. After input and output frequencies are entered, clicking the **Calculate** button will display the divider values and actual frequencies.

  If the output frequency entered is not achievable the nearest frequency will be displayed in the “Actual” text box and an error message will be displayed. The user can also enter a tolerance value in percent. When the Calculate button is pressed the calculation will be considered accurate if the result in within the entered tolerance range.

  If an entered value is out of range it will be displayed in red and an error message will be displayed after the Calculate button is used.

- **Divider Mode**: In this mode the user sets the input frequency and the divider settings. Users will choose the CLKOP divider value to maximize the frequency of the VCO within the acceptable range as specified in the MachXO2 Family Data Sheet.

If the combination of entered values will result in an invalid PLL configuration the user will be prompted by a text box to change the value with a suggestion for the value that is out of range.

Figure 13-13. MachXO2 PLL Configuration Tab

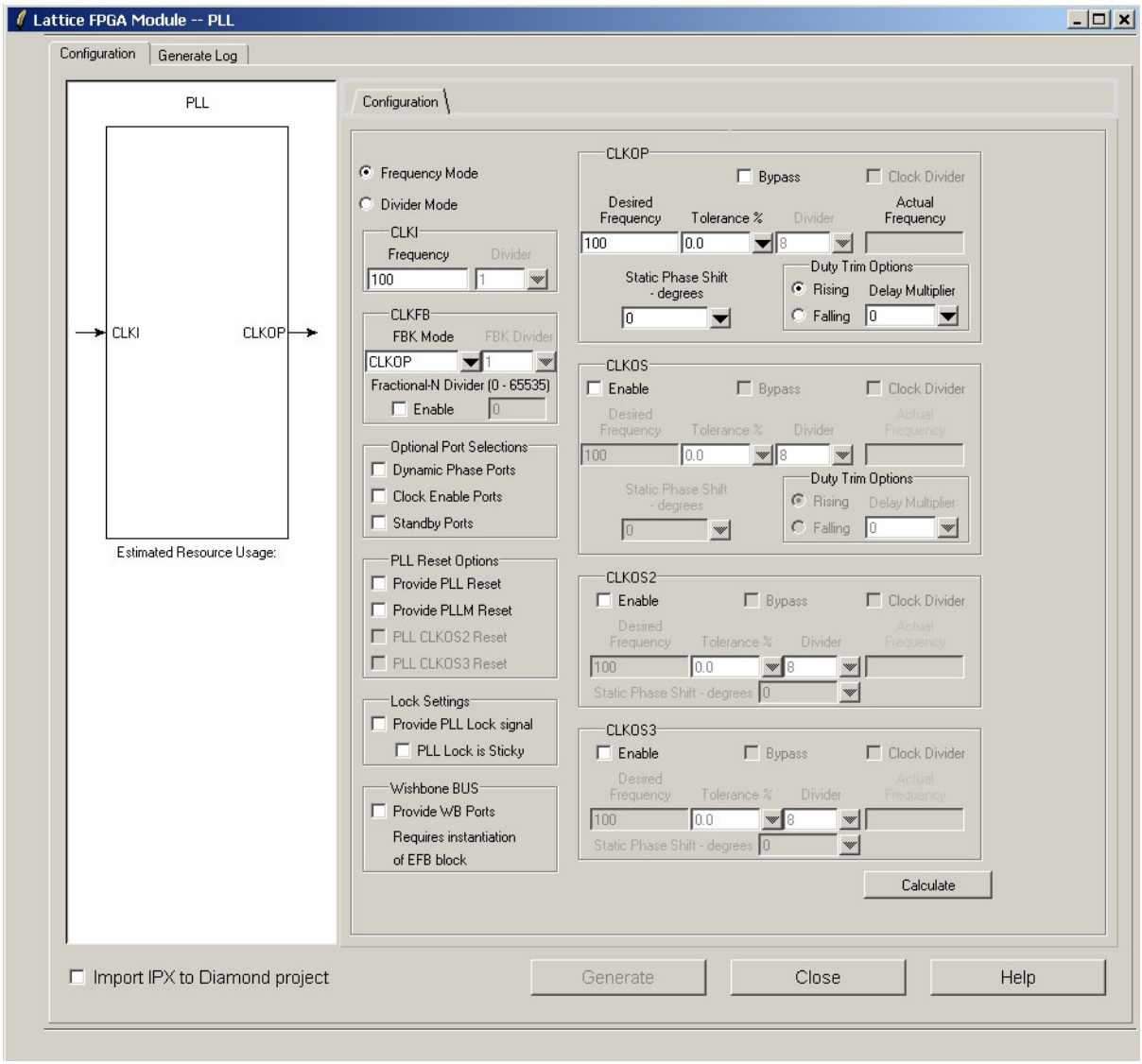


Table 13-11. User Parameters in the IPexpress GUI

User Parameter	Description	Range	Default
Frequency Mode	User enters desired CLKI and CLKOP frequency	ON/OFF	ON
Divider Mode	User enters desired CLKI frequency and divider settings	ON/OFF	OFF
CLKI	Frequency	10 to 400 MHz	100 MHz
	Divider	1 to 40	1
CLKFB	Feedback mode	CLKOP, CLKOS, CLKOS2, CLKOS3, INT_OP, INT_OS, INT_OS2, INT_OS3, UserClock	CLKOP
	Fractional-N divider enable	ON / OFF	OFF
	Fractional-N divider	0 to 65535	0
Output Port Selections	Dynamic phase ports	ON / OFF	OFF
	Clock enable ports	ON / OFF	OFF
	Standby ports	ON / OFF	OFF
PLL Reset Options	Provide PLL reset	ON / OFF	OFF
	Provide PLLM reset	ON / OFF	OFF
	Provide CLKOS2 reset	ON / OFF	OFF
	Provide CLKOS3 reset	ON / OFF	OFF
Lock Settings	Provide PLL LOCK signal	ON / OFF	OFF
	PLL LOCK is “sticky”	ON / OFF	OFF
WISHBONE Bus	Provide WISHBONE ports	ON / OFF	OFF
CLKOP	Bypass	ON / OFF	OFF
	Clock Divider (in Bypass mode only)	ON / OFF	OFF
	Desired frequency	3.125 to 400 MHz	100 MHz
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0
	Divider	1-128	8
	Actual frequency (read only)	–	–
	Static phase shift (degrees)	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°	00
	Rising edge trim	ON / OFF	OFF
	Falling edge trim	ON / OFF	OFF
	Delay multiplier	0, 1, 2, 4	0
CLKOS	Enable	ON / OFF	OFF
	Bypass	ON / OFF	OFF
	Clock divider (in Bypass mode only)	ON / OFF	OFF
	Desired frequency	0.024 – 400 MHz	100 MHz
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0
	Divider	1-128	8
	Actual frequency (read only)	–	–
	Static phase shift (degrees)	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°	00
	Rising edge trim	ON / OFF	OFF
	Falling edge trim	ON / OFF	OFF
	Delay multiplier	0, 1, 2, 4	0

Table 13-11. User Parameters in the IPexpress GUI (Continued)

User Parameter	Description	Range	Default
CLKOS2	Enable	ON / OFF	OFF
	Bypass	ON / OFF	OFF
	Clock divider (in Bypass mode only)	ON / OFF	OFF
	Desired frequency	0.024 to 400 MHz	100 MHz
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0
	Divider	1-128	8
	Actual frequency (read only)	–	–
	Static phase shift (degrees)	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°	00
CLKOS3	Enable	ON / OFF	OFF
	Bypass	ON / OFF	OFF
	Clock divider (in Bypass mode only)	ON / OFF	OFF
	Desired frequency	0.024 – 400 MHz	100 MHz
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0
	Divider	1-128	8
	Actual frequency (read only)	–	–
	Static phase shift (degrees)	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°	0°

IPexpress Output

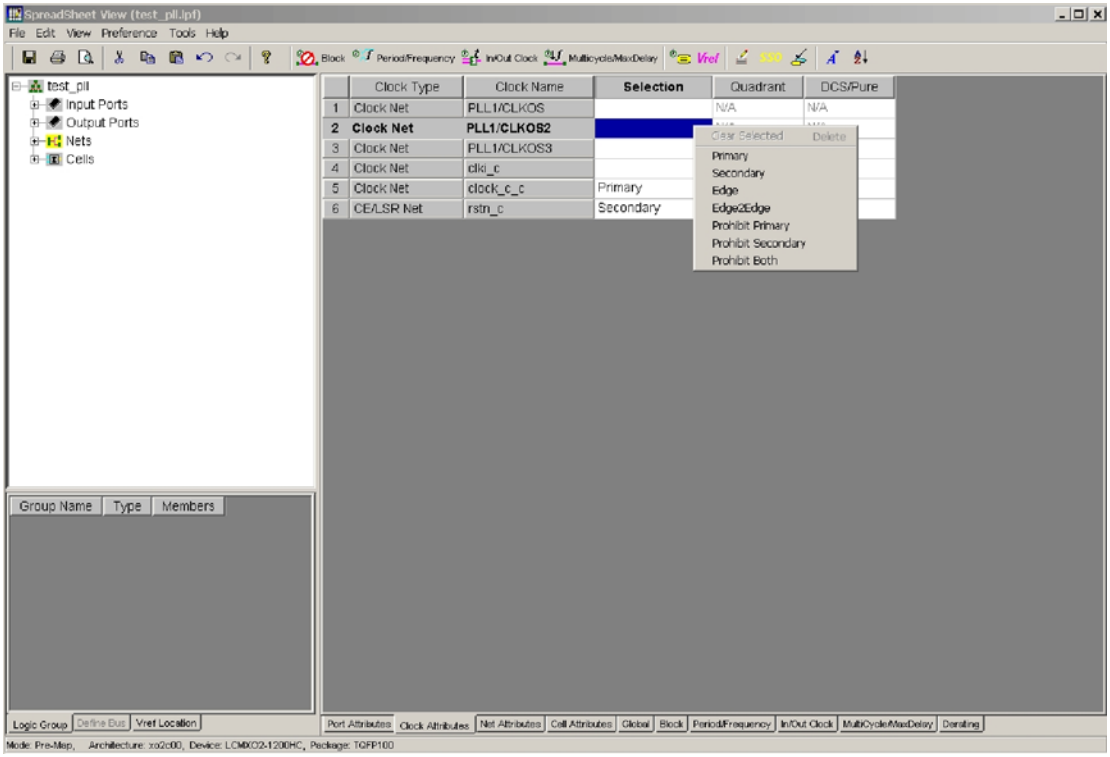
There are two IPexpress output files that are important for use in the design. The first is the <module_name.[v|vhd] file. This is the user-named module that was generated by IPexpress. This file is meant to be used in both the synthesis and simulation flows. The second is a template file, <module_name>_tmpl.[v|vhd]. This file contains a sample instantiation file of the module. This file is provided for the user to copy/paste the instance and is not intended to be used in the synthesis or simulation flows directly.

IPexpress sets attributes in the HDL module for the PLL that are specific to the data rate selected. Although these attributes can be easily changed, they should only be modified by re-running the GUI so that the performance of the PLL is maintained. After the MAP stage in the tool flow, FREQUENCY preferences will be included in the preference file to automatically constrain the clocks produced by the PLL.

Use of the Pre-MAP Preference Editor

Clock preferences can be set in the Pre-MAP Preference Editor. Figure 13-14 shows an example screen shot. The Quadrant and DCS/Pure columns are not applicable to the MachXO2 device. The Pre-MAP Preference Editor is a part of the ispLEVER® Design Planner tool. The equivalent function in Diamond is simply called the “Spreadsheet View”.

Figure 13-14. Pre-MAP Preference Editor Example



PLL Reference Clock Switch (PLLREFCS)

The MachXO2 PLL reference clock can optionally be switched between two different clock sources if desired. To use this feature the PLLREFCS primitive must be instantiated in the design. The PLLREFCS can only be used with the PLL.

When the reference clock is switched the PLL may lose lock for some period of time. In this case it can take up to the t_{LOCK} time specified in the MachXO2 Family Data Sheet to re-acquire lock. It is recommended that the PLL be reset when switching between reference clock signals which are at different frequencies.

The PLLREFCS primitive can be instantiated in the source code of a design as defined in this section. Figure 13-15 and Table 13-12 show the PLLREFCS definitions.

Figure 13-15. PLLREFCS Primitive Symbol

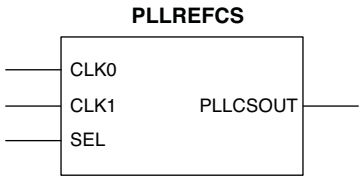


Table 13-12. PLLREFCS Primitive Port Definition

Port Name	I/O	Description
CLK0	NO	CLK0
CLK1	NO	CLK1
SEL	NO	SEL - SEL = 0 CLK0 input is selected - SEL = 1 CLK1 input is selected
PLLCSOUT	NO	PLLCSOUT

Internal Oscillator (OSCH)

The MachXO2 device has an internal oscillator that can be used as a clock source in a design. The internal oscillator accuracy is +/- 5% (nominal). This oscillator is intended as a clock source for applications that do not require a higher degree of accuracy in the clock.

The internal oscillator of the MachXO2 remains active to the user logic during transparent configuration. The clock provided by the internal oscillator to the fabric will not stop or be influenced while the oscillator is also being used internally for background configuration. Although only one internal oscillator is within the MachXO2 device, the user and configuration clocks are sourced from independent clock dividers and resources.

The oscillator output is routed through a divider to provide a flexible clock frequency source. The available output frequencies are shown in Table 13-15.

OSCH Primitive Definition

The OSCH primitive can be instantiated in the source code of a design as defined in this section. Figure 13-16 and Tables 13-13 through 13-15 show the OSCH definitions.

Figure 13-16. OSCH Primitive Symbol

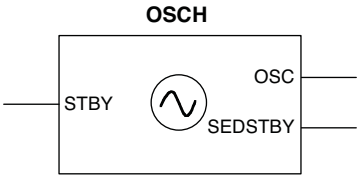


Table 13-13. OSCH Primitive Port Definition

Port Name	I/O	Description
STDBY	I	Standby – power down the oscillator in standby mode - STDBY = 0 OSC output is active - STDBY = 1 OSC output is OFF
OSC	O	Clock output port
SEDSTDBY	O	Standby – power down SED clock[1]

1. This output is used to notify the SED block that the oscillator will shut down when the device goes into standby. Only required for simulation purposes.

Table 13-14. OSCH Primitive Attribute Definition

Name	Description	Value	Default
Nominal Frequency (MHz)	NOM_FREQ	2.08, 2.15, 2.22, … 66.5, 88.67, 133.0 (See Table 13-15 for a complete listing)	2.08 MHz

Table 13-15. OSCH Supported Frequency Settings

2.08	4.16	8.31	15.65
2.15	4.29	8.58	16.63
2.22	4.43	8.87	17.73
2.29	4.59	9.17	19.00
2.38	4.75	9.50	20.46
2.46	4.93	9.85	22.17
2.56	5.12	10.23	24.18
2.66	5.32	10.64	26.60
2.77	5.54	11.08	29.56
2.89	5.78	11.57	33.25
3.02	6.05	12.09	38.00
3.17	6.33	12.67	44.33
3.33	6.65	13.30	53.20
3.50	7.00	14.00	66.50
3.69	7.39	14.78	88.67
3.91	7.82	15.65	133.00

The NOM_FREQ attribute setting must match the value in the table or the software will issue a warning message and ignore the attribute value.

The STDBY port can be used to power down the oscillator when it is not being used. This port can be connected to a user signal or an I/O pin. The user must insure that the oscillator is not turned off when it is needed for operations such as WISHBONE bus operations, SPI or I^2C configuration, SPI or I^2C user mode operations, background Flash updates or SED.

OSCH Declaration in VHDL Source Code

Library Instantiation

```
library machxo2;
use machxo2.all;
```

Component and Attribute Declaration

```
COMPONENT OSCH
-- synthesis translate_off
      GENERIC  (NOM_FREQ: string := "2.56");
-- synthesis translate_on
      PORT (   STDBY     :IN   std_logic;
               OSC       :OUT  std_logic;
               SEDSTDBY  :OUT  std_logic);
END COMPONENT;

    attribute NOM_FREQ : string;
    attribute NOM_FREQ of OSCinst0 : label is "2.56";
```

OSCH Instantiation

```
begin
OSCInst0: OSCH
-- synthesis translate_off
      GENERIC MAP  ( NOM_FREQ  => "2.56" )
-- synthesis translate_on
      PORT MAP (   STDBY=>  stdby,
                   OSC=>  osc_int,
                   SEDSTDBY=>  stdby_sed
                   );
```

OSCH Instantiation in Verilog Source Code

```
//   Internal Oscillator
//   defparam OSCH_inst.NOM_FREQ = "2.08";//  This is the default frequency
     defparam OSCH_inst.NOM_FREQ = "24.18";

OSCH OSCH_inst( .STDBY(1'b0),              //  0=Enabled, 1=Disabled
//  also Disabled with Bandgap=OFF
                .OSC(osc_clk),
                .SEDSTDBY());              //  this signal is not required if not
//  using SED
```

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

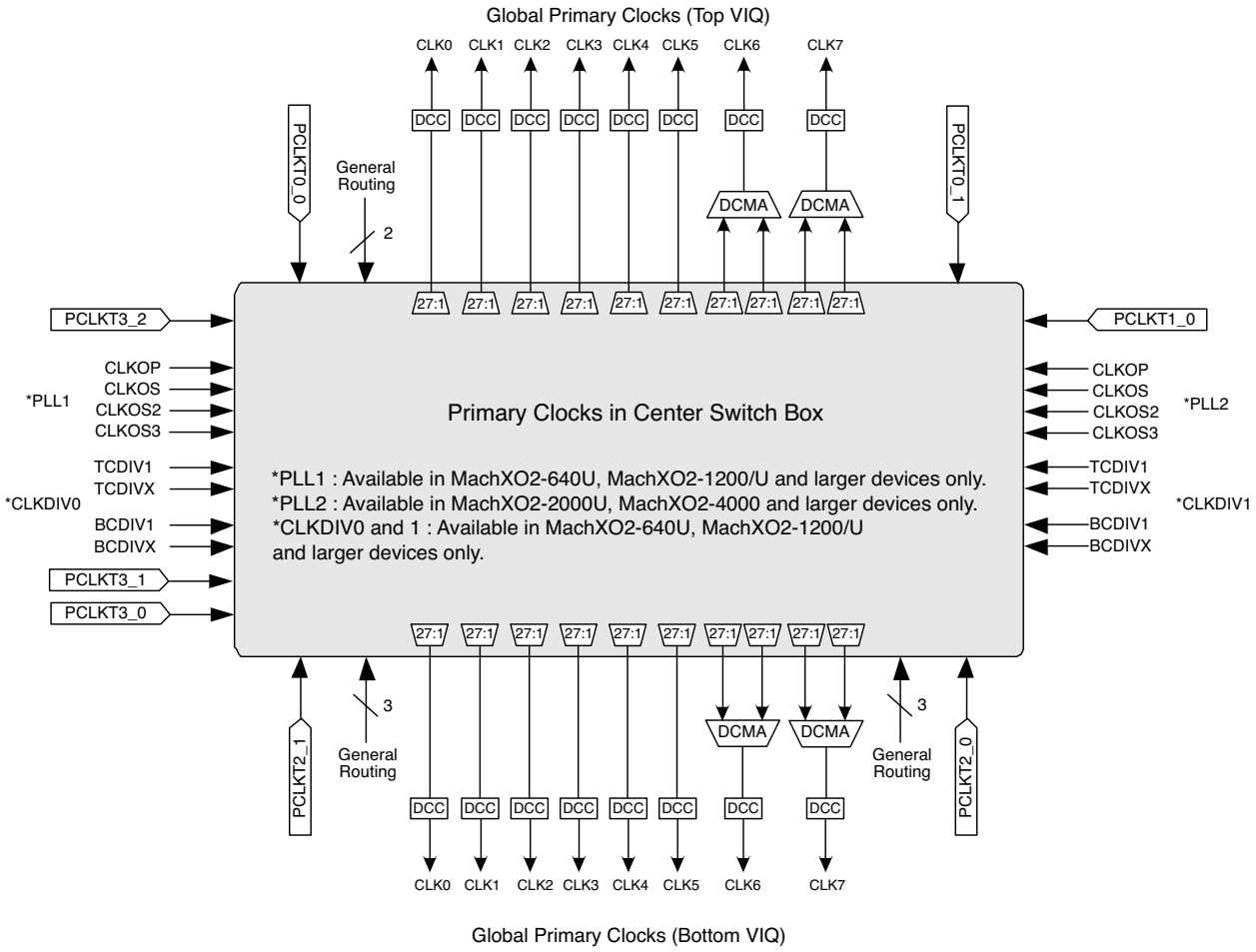
Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
January 2011	01.1	Updated for ultra-high I/O (“U”) devices.
April 2011	01.2	MachXO2 Clocking Structure (MachXO2-1200) diagram - Changed bank 4 to bank 3.
		Updated MachXO2 PLL Block Diagram. Clarified WISHBONE port connections. Removed PLLWAKESYNC and INTLOCK signal descriptions. Added Frequency Calculation and Fractional-N Synthesis Operation descriptions. Added Verilog Instantiation example for the Oscillator.
		Corrected OSCH Supported Frequency Settings table.
		Added WISHBONE Register Descriptions in PLL WISHBONE Register Descriptions.
June 2011	01.3	Clarified PCLK names in Appendix A and Appendix B.
		Added note about Lock signal glitch on “R1” devices when coming out of standby.
		Clarified VHDL code examples throughout the document.
July 2011	01.4	Updated the Edge Clock Synchronization (ECLKSYNCA) and Lock Output text sections with information on migration from MachXO2-1200-R1 to Standard (non-R1) devices.
August 2011	01.5	Clarified PLL WISHBONE operation, Appendix D.

Date	Version	Change Summary
January 2012	01.6	Document status updated from advance to final.
		Library instantiation information updated throughout the document.
February 2012	01.7	Updated document with new corporate logo.
May 2012	01.8	Updated Fractional-N section to add that additional jitter is introduced when using this function.
August 2012	01.9	Updated RESETM Input operation.
August 2012	02.0	Further clarification to RESETM Input section.
August 2012	02.1	RST Input section – Clarified function of RST signal.
September 2012	02.2	Clarified oscillator usage for user mode and configuration logic.

Appendix A. Primary Clock Sources and Distribution

Figure 13-17. MachXO2 Primary Clock Sources and Distribution



Note: The MachXO2 has eight global primary clocks. Each primary clock is driven out the top and bottom of the Primary Clock Center Switch Box. The top and bottom drivers must use the same clock source for each primary clock.

Figure 13-18. MachXO2 Primary Clock Muxes – MachXO-640U, MachXO2-1200/U and Higher Density Devices

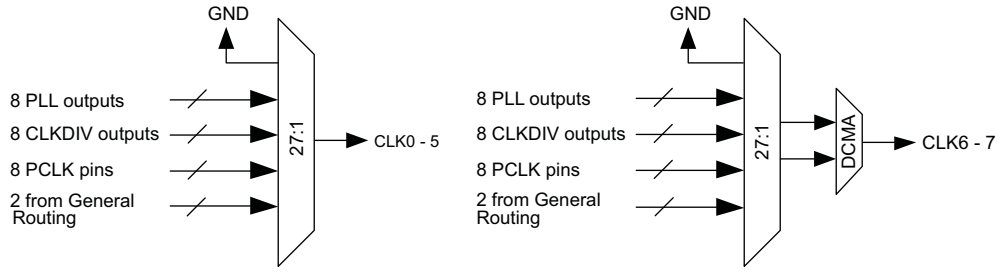
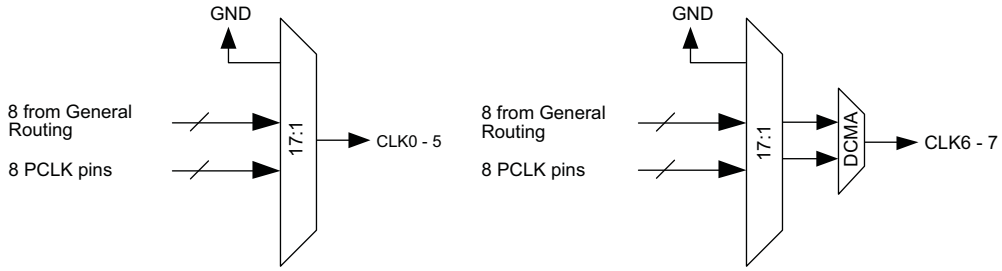
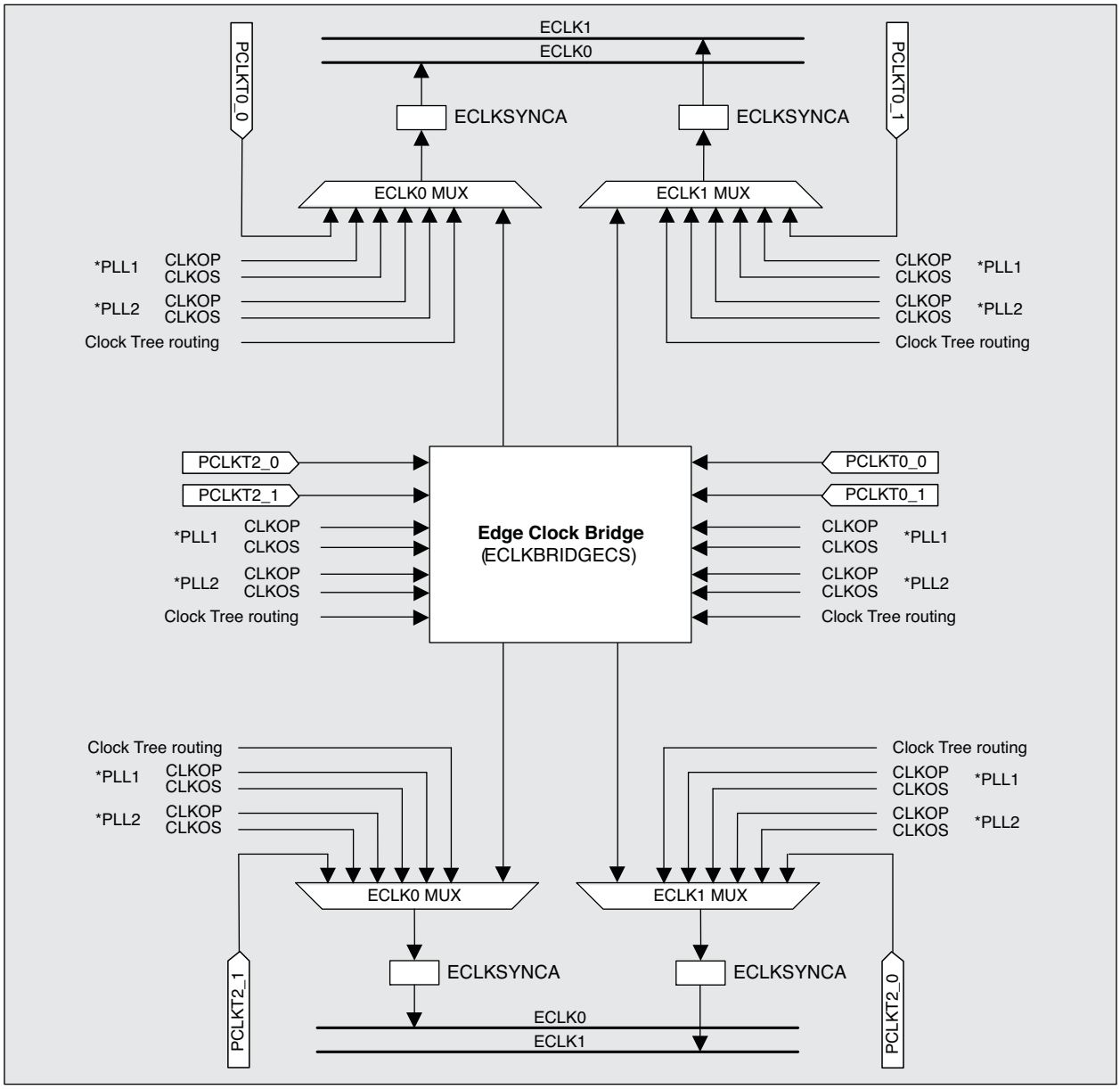


Figure 13-19. MachXO2 Primary Clock Muxes – MachXO2-256 and MachXO2-640



Appendix B. Edge Clock Sources and Connectivity

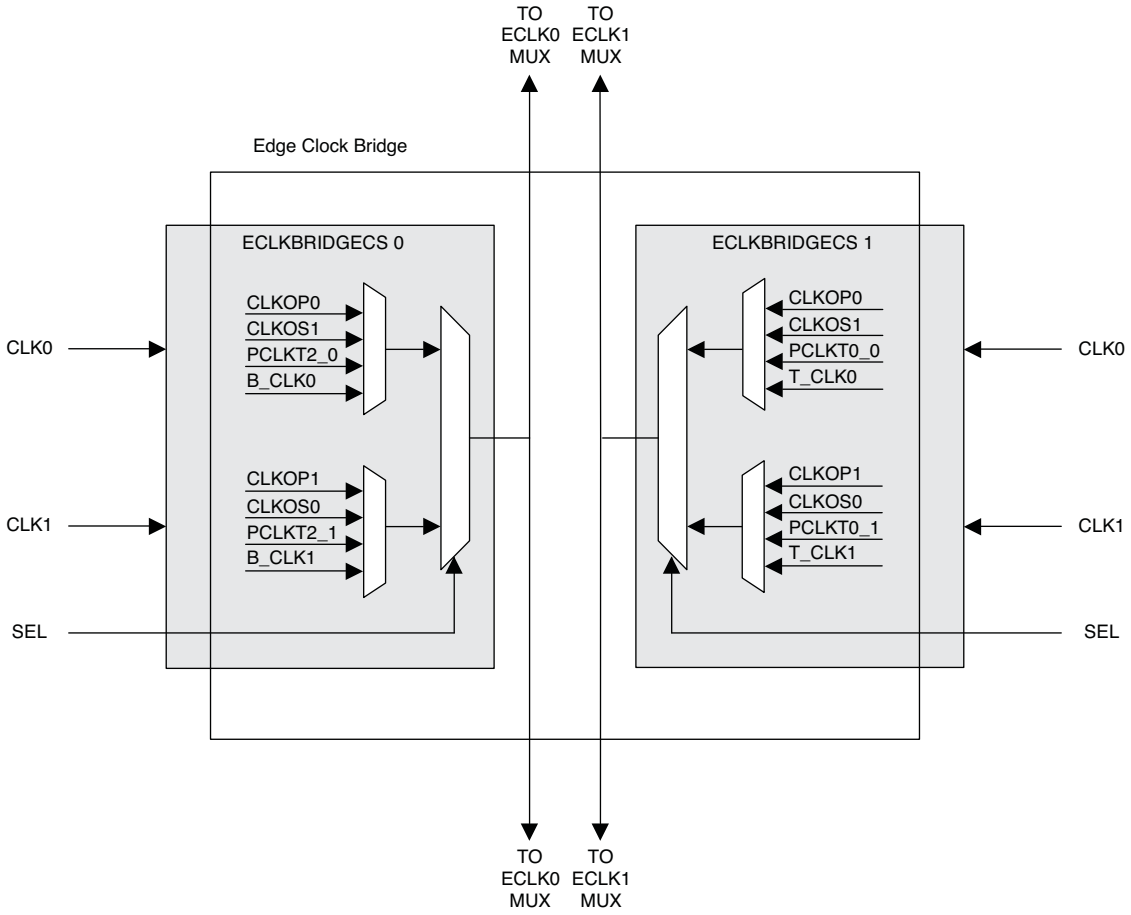
Figure 13-20. MachXO2 Edge Clock Sources and Connectivity



Notes:

1. The MachXO2 has edge clock resources on the MachXO2-640U, MachXO2-1200/U and higher density devices only.
2. The edge clock muxes ECLK0 MUX and ECLK1 MUX are routing resources available to the software. There is no dynamic switching between inputs on these muxes. To dynamically switch between edge clock drivers requires that the ECLKBRIDGECS element be instantiated in the design.

Figure 13-21. MachXO2 Edge Clock Bridge Sources and Connectivity



Notes:

1. The edge clock bridge allows a single clock signal to drive both the top and bottom edge clock with minimal skew. It can also be used where switching between the clock sources is desired.
2. The edge clock bridge resource is available in MachXO2-640U, MachXO2-1200/U and higher density devices.
3. To use the edge clock bridge the ECLKBRIDGECS primitive must be instantiated in the design. There are two ECLKBRIDGECS resources available in devices that have an edge clock bridge.

Appendix C. Clock Preferences

A few key clock preferences are introduced below. Refer to the 'Help' file for other preferences and detailed information.

FREQUENCY
The following physical preference assigns a frequency of 100 MHz to a net named clk1:

```
FREQUENCY NET "clk1" 100 MHz;
```

The following preference specifies a hold margin value for each clock domain:

```
FREQUENCY NET "RX_CLKA_CMOS_c" 100.000 MHz HOLD_MARGIN 1 ns;
```

MAXSKEW
The following preference assigns a maximum skew of 5 ns to a net named NetB:

```
MAXSKEW NET "NetB" 5 NS;
```

MULTICYCLE
The following preference will relax the period to 50 ns for the path starting at COMPA to COMPB (NET1):

```
MULTICYCLE "PATH1" START COMP "COMPA" END COMP "COMPB" NET
"NET1" 50 NS ;
```

PERIOD
The following preference assigns a clock period of 30 ns to the port named Clk1:

```
PERIOD PORT "Clk1" 30 NS;
```

PROHIBIT
The following preference prohibits the use of a primary clock to route a clock net named bf_clk:

```
PROHIBIT PRIMARY NET "bf_clk";
```

The following preference prohibits the use of a secondary high fan-out net to route a clock net named bf_clk:

```
PROHIBIT SECONDARY NET "bf_clk";
```

PROHIBIT_BOTH
When this setting is selected it causes Design Planner to generate both the PROHIBIT PRIMARY NET net_name and PROHIBIT SECONDARY NET net_name.

USE PRIMARY
Use a primary clock resource to route the specified net:

```
USE PRIMARY NET clk_fast;

USE PRIMARY DCCA NET "bf_clk";

USE PRIMARY PURE NET "bf_clk" QUADRANT_TL;
```

USE SECONDARY
Use a secondary high fan-out net resource to route the specified net:

```
USE SECONDARY NET "clk_lessfast" QUADRANT_TL;
```

USE EDGE
Use an edge clock resource to route the specified net. The net must be eligible for routing using the edge clock resources.

```
USE EDGE NET "clk_fast";
```

EDGE2EDGE
Use the ECLK bridge resource to route the specified net. The net must be eligible for routing using the edge clock resources.

```
USE EDGE2EDGE NET "clk_fast";
```

CLOCK_TO_OUT
This preference specifies a maximum allowable output delay relative to a clock.

Here are two preferences using both the CLKPORT and CLKNET keywords showing the corresponding scope of TRACE reporting.

The CLKNET will stop tracing the path before the PLL, so you will not get PLL compensation timing numbers.

```
CLOCK_TO_OUT PORT "RxAddr_0" 6.000000 ns CLKNET "pll_rxclk" ;
```

The above preference will yield the following clock path:

```
    Clock path pll_inst/pll_utp_0_0 to PFU_33:

Name         Fanout      Delay (ns)  Site              Resource
ROUTE        49          2.892       ULPPLL.MCLK to    R3C14.CLK0 pll_rxclk
                         --------
                         2.892 (0.0% logic, 100.0% route), 0 logic levels.
```

If CLKPORT is used, the trace is complete back to the clock port resource and provides PLL compensation timing numbers.

```
    CLOCK_TO_OUT PORT "RxAddr_0" 6.000000 ns CLKPORT "RxClk" ;
```

The above preference will yield the following clock path:

```
    Clock path RxClk to PFU_33:

Name         Fanout      Delay (ns)  Site              Resource
IN_DEL       ---         1.431       D5.PAD to         D5.INCK RxClk
ROUTE        1           0.843       D5.INCK to        ULPPLL.CLKIN RxClk_c
MCLK_DEL     ---         3.605       ULPPLL.CLKIN to   ULPPLL.MCLK
                                                       pll_inst/pll_utp_0_0
ROUTE        49          2.892       ULPPLL.MCLK to    R3C14.CLK0 pll_rxclk
             --------
             8.771 (57.4% logic, 42.6% route), 2 logic levels.
```

INPUT_SETUP
This preference specifies a setup time requirement for input ports relative to a clock net.

```
    INPUT_SETUP PORT "datain" 2.000000 ns HOLD 1.000000 ns CLKPORT "clk"
    PLL_PHASE_BACK ;
```

PLL_PHASE_BACK
This preference is used with INPUT_SETUP when a user needs a trace calculation based on the previous clock edge.

This preference is useful when setting the PLL output phase adjustment. Since there is no negative phase adjustment provided, the PLL_PHASE_BACK preference works as if negative phase adjustment is available.

For example:

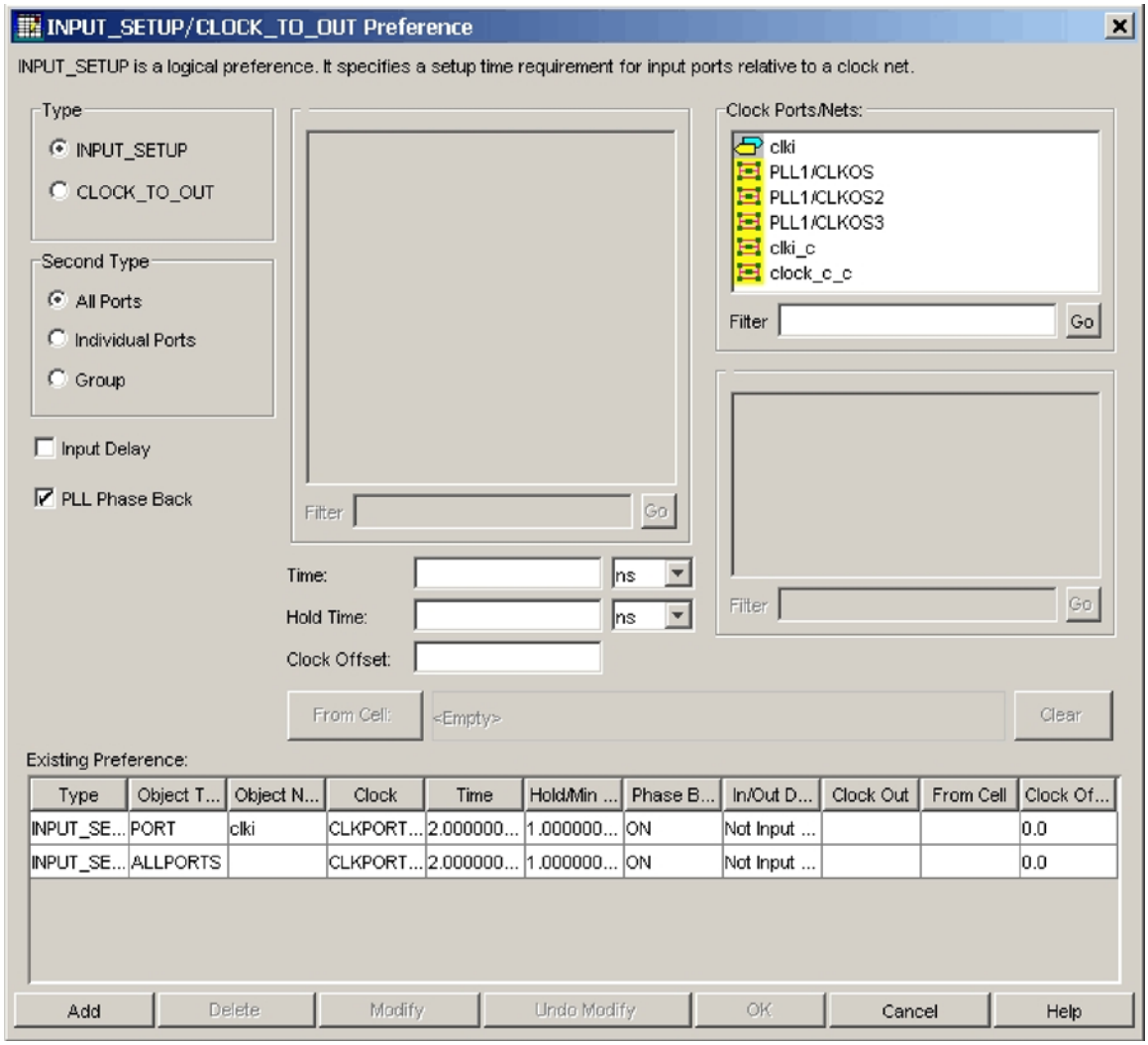
If phase adjustment of -90° of CLKOS is desired, a user can set the phase to 270° and set the INPUT_SETUP preference with PLL_PHASE_BACK.

PLL_PHASE_BACK Usage in Pre-Map Preference Editor
The Pre-Map Preference Editor can be used to set the PLL_PHASE_BACK attribute.

1. Open the Design Planner (Pre-Map).
2. In the Design Planner control window, select **View > Spreadsheet View**.
3. In the Spreadsheet View window, select **Input_setup/Clock_to_out…**

The INPUT_SETUP/CLOCK_TO_OUT preference window with the PLL phase back feature is shown in Figure 13-22.

Figure 13-22. Input_SETUP/CLOCK_to_OUT Preference Window



Appendix D. PLL WISHBONE Bus Operation

The MachXO2 PLL operating parameters can be changed dynamically via the Embedded Function Block's (EFB's) WISHBONE bus. The user must instantiate the EFB block in their design to use this feature. The user logic's WISHBONE bus is then connected to the EFB block. A hard-wired PLL Data Bus is used to communicate between the EFB and the PLL. See Using User Flash Memory and Hardened Control Functions in MachXO2 Devices (TN1205) for more information about the using the EFB block in a design.

The PLL Data Bus on the PLL module provides support for functional simulation of this operation. The user must connect the PLL Data Bus to the EFB in their HDL design in order for simulation to work properly. The PLL Data Bus ports and the corresponding EFB PLL Bus port connections are listed in Table 13-16.

Table 13-16. PLL Data Bus Port Definitions

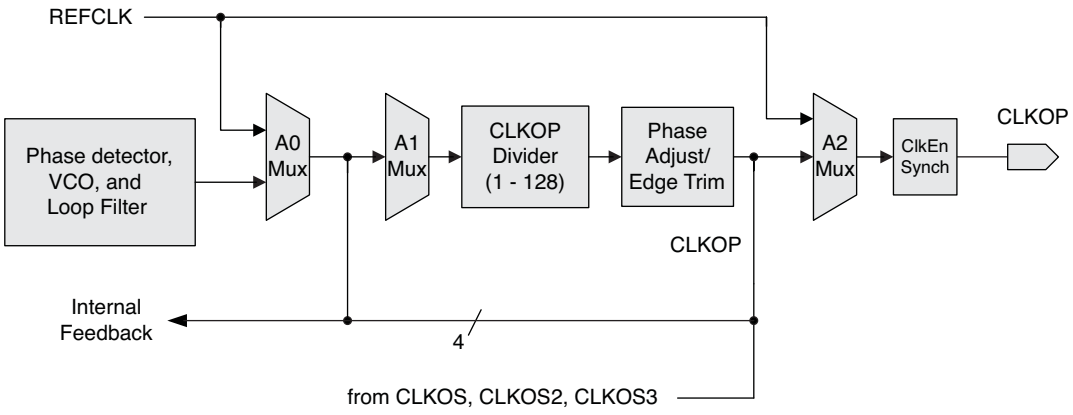
PLL Port Name	I/O	Description	EFB Port Name
PLLCLK	I	PLL data bus clock input signal	pll_bus_o[16]
PLLRST	I	PLL data bus reset. This resets only the data bus, not any register values.	pll_bus_o[15]
PLLSTB	I	PLL data bus strobe signal.	pll_bus_o[14]
PLLWE	I	PLL data bus write enable signal	pll_bus_o[13]
PLLADDR [4:0]	I	PLL data bus address	pll_bus_o[12:8]
PLLDATI [7:0]	I	PLL data bus data input	pll_bus_o[7:0]
PLLDATO [7:0]	O	PLL data bus data output	pll_bus_i[8:1]
PLLACK	O	PLL data bus acknowledge signal	pll_bus_i[0]

PLL Architecture

The MachXO2 PLL has four output sections with flexible configuration settings to support a variety of different applications. IPexpress is able to support most of the common PLL configurations, but for those users with more complex needs the WISHBONE bus can be used to change the PLL configuration which allows for more advanced support options.

Each of the four PLL output sections have similar configuration options. Each output section is assigned a letter designator; A for the CLKOP output, B for the CLKOS output, C for the CLKOS2 output, and D for the CLKOS3 output section. Within each of the four output sections there are three signal selection muxes which are used to control the PLL configuration. A diagram of the A output section is shown in Figure 13-23. The B output section is the same as the A section except the muxes are labeled B0, B1, and B2. The C and D sections are similar with muxes labeled C0, C1, C2, D0, D1, and D2. The C and D sections have the Phase Adjust block but not the Edge Trim feature.

Figure 13-23. PLL CLKOP Output Section



The EFB WISHBONE register map for the PLL registers is shown in Table 13-17 (add 0x20 for the corresponding locations to access an optional second MachXO2 PLL).

Table 13-17. EFB WISHBONE Locations for PLL Registers

Reg.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	MC1_DIVFBK_FRAC[7:0]							
1	MC1_DIVFBK_FRAC[15:8]							
2	MC1_LOADREG	MC1_DELA[6:0]						
3	MC1_PLLPDN	MC1_DELB[6:0]						
4	MC1_WBRESET	MC1_DELC[6:0]						
5	MC1_USE_DESI	MC1_DELD[6:0]						
6	MC1_REFIN_RESET	MC1_DIVA[6:0]						
7	MC1_PLLRST_ENA	MC1_DIVB[6:0]						
8	MC1_MRST_ENA	MC1_DIVC[6:0]						
9	MC1_STDBY	MC1_DIVD[6:0]						
A	MC1_ENABLE_SYNC	MC1_PHIB[2:0]			MC1_INT_LOCK_STICKY	MC1_PHIA[2:0]		
B	MC1_DCRST_ENA	MC1_PHID[2:0]			MC1_RESERVED2	MC1_PHIC[2:0]		
C	MC1_DDRST_ENA	MC1_SEL_OUTB[2:0]			MC1_INTFB	MC1_SEL_OUTA[2:0]		
D	MC1_LOCK[1:0]		MC1_SEL_OUTC[2:0]			MC1_SEL_OUTD[2:0]		
E	MC1_SEL_DIVA[1:0]		MC1_SEL_DIVB[1:0]		MC1_SEL_DIVC[1:0]		MC1_SEL_DIVD[1:0]	
F	MC1_CLKOP_TRIM[3:0]				MC1_CLKOS_TRIM[3:0]			
10	MC1_DYN_SOURCE	MC1_LOCK_SEL[2:0]			MC1_ENABLE_CLK[3:0]			
11	MC1_TRIMOS3_BYPASS_N	MC1_TRIMOS2_BYPASS_N	MC1_TRIMOS_BYPASS_N	MC1_TRIMOP_BYPASS_N	MC1_DYN_SEL[1:0]		MC1_DIRECTION	MC1_ROTATE
12	MC1_LF_RESGRND	MC1_SEL_REF1[2:0]			MC1_EN_UP	MC1_SEL_REF2[2:0]		
13	MC1_DIVFBK_ORDER[1:0]		MC1_CLKMUX_FB[1:0]		MC1_SEL_FBK[3:0]			
14	MC1_GMC_RESET	MC1_DIVREF[6:0]						
15	MC1_FORCE_VFILTER	MC1_DIVFBK[6:0]						
16	MC1_LF_PRESET	MC1_LF_RESET	MC1_TEST_ICP	MC1_EN_FILTER_OPAMP	MC1_FLOAT_ICP	MC1_GPROG[2:0]		
17	MC1_KPROG[2:0]			MC1_IPROG[4:0]				
18	MC1_GMC_PRESET	MC1_RPROG[6:0]						
19	MC1_GMCREF_SEL[1:0]		MC1_MFGOUT2_SEL[2:0]			MC1_MFGOUT1_SEL[2:0]		
1A	MC1_GMCSEL[3:0]				MC1_VCO_BYPASS_D0	MC1_VCO_BYPASS_C0	MC1_VCO_BYPASS_B0	MC1_VCO_BYPASS_A0
1B	MC1_RESERVED[4:0]					MC1_EN_PHI	MC1_DPROG[1:0]	
1C	RESERVED							LOCK_STS

Note: Registers 0 through 11 are user accessible registers. The remaining registers are reserved for Lattice use or read-only access.

Table 13-18. PLL Register Descriptions

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access	GUI Access
MC1_DIVFBK_FRAC[15:0]	0[7:0] 1[7:0]	16	Fractional-N divider value. Fractional-N divider is equal to this value / 65535.	0	Yes	Yes
MC1_LOADREG	2[7]	1	Only valid if MC1_DYN_PHASE=0. Command to start a divider output phase shift on negative edge of MC1_LOADREG bit. The divider output phase shift for CLKOP will occur if the MC1_DIVA and MC1_DELA values are not the same. A CLKOS divider output phase shift will occur if the MC1_DIVB and MC1_DELB values are not the same. A CLKOS2 divider output phase shift will occur if the MC1_DIVC and MC1_DELC values are not the same. A CLKOS3 divider output phase shift will occur if the MC1_DIVD and MC1_DELD values are not the same.	0	Yes	N/A
MC1_PLLPDN	3[7]	1	Power down the PLL when not used. Software automatically sets this to '1' when the PLL is used in a design and to '0' if the PLL is not used. 0 = Power down PLL. 1 = PLL powered up.	1	Yes	Yes automatic
MC1_WBRESET	4[7]	1	PLL reset from Wishbone – Equivalent to the RESETM port operation. 0 = PLL normal operation. 1 = PLL reset active.	0	Yes	No
MC1_USE_DESI	5[7]	1	Controls whether the Fractional-N divider is used. 0 = PLL normal operation. 1 = Use Fractional-N divider.	0	Yes	Yes
MC1_REFIN_RESET	6[7]	1	Controls whether the PLL is automatically reset when the input clock reference is switched using the PLLREFCS primitive 0 = Do not reset PLL. 1 = Automatically reset PLL if input switches.	0	Yes	No
MC1_PLLRST_ENA	7[7]	1	Enable the PLLRESET port. 0 = PLLRESET port not active. 1 = PLLRESET port is enabled.	0	Yes	Yes
MC1_MRST_ENA	8[7]	1	Enable the RESETM port. 0 = RESETM port not active. 1 = RESETM port is enabled.	0	Yes	Yes
MC1_STDBY	9[7]	1	Enable the STDBY port on PLL 0 = STDBY port not active. 1 = STDBY port is enabled.	0	Yes	Yes
MC1_ENABLE_SYNC	A[7]	1	Enable synchronous disable/enable of secondary clocks CLKOS, CLKOS2, CLKOS3 with respect to CLKOP. 0 = Synchronous disable/enable not active. 1 = Synchronous disable/enable is active.	0	Yes	No
MC1_DCRST_ENA	B[7]	1	Enable the RESETDC port – CLKOS2 reset. 0 = RESETDC port not active. 1 = RESETDC port is enabled.	0	Yes	Yes

Table 13-18. PLL Register Descriptions (Continued)

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access	GUI Access
MC1_DDRST_ENA	C[7]	1	Enable the RESETDD port – CLKOS3 reset. 0 = RESETDD port not active. 1 = RESETDD port is enabled.	0	Yes	Yes
MC1_DELA[6:0]	2[6:0]	7	CLKOP section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVA[6:0].	7	Yes	Yes
MC1_DELB[6:0]	3[6:0]	7	CLKOS section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVB[6:0].	7	Yes	Yes
MC1_DELC[6:0]	4[6:0]	7	CLKOS2 section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVC[6:0].	7	Yes	Yes
MC1_DELD[6:0]	5[6:0]	7	CLKOS3 section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVD[6:0].	7	Yes	Yes
MC1_DIVA[6:0]	6[6:0]	7	CLKOP section output divider setting equal to the Divide value - 1.	7	Yes	Yes
MC1_DIVB[6:0]	7[6:0]	7	CLKOS section output divider setting equal to the Divide value - 1.	7	Yes	Yes
MC1_DIVC[6:0]	8[6:0]	7	CLKOS2 section output divider setting equal to the Divide value - 1.	7	Yes	Yes
MC1_DIVD[6:0]	9[6:0]	7	CLKOS3 section output divider setting equal to the Divide value - 1.	7	Yes	Yes
MC1_PHIA[2:0]	A[2:0]	3	Select the VCO phase shift (0-7) for CLKOP. Each tap represents 45 degree shift of the VCO.	0	Yes	Yes
MC1_PHIB[2:0]	A[6:4]	3	Select the VCO phase shift (0-7) for CLKOS. Each tap represents 45 degree shift of the VCO.	0	Yes	Yes
MC1_PHIC[2:0]	B[2:0]	3	Select the VCO phase shift (0-7) for CLKOS2. Each tap represents 45 degree shift of the VCO.	0	Yes	Yes
MC1_PHID[2:0]	B[6:4]	3	Select the VCO phase shift (0-7) for CLKOS3. Each tap represents 45 degree shift of the VCO.	0	Yes	Yes
MC1_INT_LOCK_STICKY	A[3]	1	Sets internal lock to be sticky or not. Sticky lock will stay high once lock is achieved until the PLL is reset or powered down. Internal lock is not used in the PLL. 0 = Internal lock normal operation. 1 = Internal lock sticky operation.	1	Yes	Not used
MC1_RESERVED2	B[3]	1	Not used.	N/A	N/A	N/A

Table 13-18. PLL Register Descriptions (Continued)

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access	GUI Access
MC1_SEL_OUTA[2:0]	C[2:0]	3	Mux A2 select value for CLKOP output. Can be used to cascade dividers if desired. 000 = DIVA output to CLKOP. 001 = DIVB output to CLKOP. 010 = DIVC output to CLKOP. 011 = DIVD output to CLKOP. 100 = REFCLK output to CLKOP (same as bypass mode without using any clock divider). Other values are for Lattice internal use only.	000	Yes	No
MC1_SEL_OUTB[2:0]	C[6:4]	3	Mux B2 select value for CLKOS output. Can be used to cascade dividers if desired. 000 = DIVB output to CLKOS. 001 = DIVC output to CLKOS. 010 = DIVD output to CLKOS. 011 = DIVA output to CLKOS. 100 = REFCLK output to CLKOS (same as bypass mode without using any clock divider). Other values are for Lattice internal use only.	000	Yes	No
MC1_SEL_OUTC[2:0]	D[5:3]	3	Mux C2 select value for CLKOS2 output. Can be used to cascade dividers if desired. 000 = DIVC output to CLKOS2. 001 = DIVD output to CLKOS2. 010 = DIVA output to CLKOS2. 011 = DIVB output to CLKOS2. 100 = REFCLK output to CLKOS2 (same as bypass mode without using any clock divider). Other values are for Lattice internal use only.	000	Yes	No
MC1_SEL_OUTD[2:0]	D[2:0]	3	Mux D2 select value for CLKOS3 output. Can be used to cascade dividers if desired. 000 = DIVD output to CLKOS3. 001 = DIVA output to CLKOS3. 010 = DIVB output to CLKOS3. 011 = DIVC output to CLKOS3. 100 = REFCLK output to CLKOS3 (same as bypass mode without using any clock divider). Other values are for Lattice internal use only.	000	Yes	No
MC1_INTFB	C[3]	1	Use the PLL internal feedback for initial PLL lock. Used with INTLOCK and PLLWAKESYNC ports. NOT RECOMMENDED to change this. 0 = PLL internal feedback is not used. 1 = Use PLL internal feedback.	0	Yes	No

Table 13-18. PLL Register Descriptions (Continued)

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access	GUI Access
MC1_LOCK[1:0]	D[7:6]	2	Frequency lock-detector resolution or sensitivity. 00 = +/- 250 ppm 01 = +/- 1000 ppm 10 = +/- 4000 ppm 11 = +/- 16000 ppm	00	Yes	No
MC1_SEL_DIVA[1:0]	E[7:6]	2	Mux A1 select value for input to DIVA (CLKOP). Can be used to cascade dividers if desired. 00 = MUX A0 output. 01 = DIVD (CLKOS3) output. 10 = DIVB (CLKOS) output. 11 = DIVC (CLKOS2) output.	00	Yes	No
MC1_SEL_DIVB[1:0]	E[5:4]	2	Mux B1 select value for input to DIVB (CLKOS). Can be used to cascade dividers if desired. 00 = MUX B0 output. 01 = DIVA (CLKOP) output. 10 = DIVD (CLKOS3) output. 11 = DIVC (CLKOS2) output.	00	Yes	No
MC1_SEL_DIVC[1:0]	E[3:2]	2	Mux C1 select value for input to DIVC (CLKOS2). Can be used to cascade dividers if desired. 00 = MUX C0 output. 01 = DIVA (CLKOP) output. 10 = DIVB (CLKOS) output. 11 = DIVD (CLKOS3) output.	00	Yes	No
MC1_SEL_DIVD[1:0]	E[1:0]	2	Mux D1 select value for input to DIVD (CLKOS3). Can be used to cascade dividers if desired. 00 = MUX D0 output. 01 = DIVA (CLKOP) output. 10 = DIVB (CLKOS) output. 11 = DIVC (CLKOS2) output.	00	Yes	No
MC1_CLKOP_TRIM[3:0]	F[7:4]	4	CLKOP output trimming control. Bit 3 of TRIM[3:0] sets the edge to be affected. TRIM[3] = 0 sets Falling edge trim active. TRIM[3] = 1 sets Rising edge trim active. TRIM[2:0] is a one hot signal. TRIM[2:0] = 001 sets 70 ps trim. TRIM[2:0] = 010 sets 140 ps trim. TRIM[2:0] = 100 sets 280 ps trim.	0000	Yes	Yes
MC1_CLKOS_TRIM[3:0]	F[3:0]	4	CLKOS output trimming control. Bit 3 of TRIM[3:0] sets the edge to be affected. TRIM[3] = 0 sets Falling edge trim active. TRIM[3] = 1 sets Rising edge trim active. TRIM[2:0] is a one hot signal. TRIM[2:0] = 001 sets 70 ps trim. TRIM[2:0] = 010 sets 140 ps trim. TRIM[2:0] = 100 sets 280 ps trim.	0000	Yes	Yes

Table 13-18. PLL Register Descriptions (Continued)

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access	GUI Access
MC1_ENABLE_CLK[3:0]	10[3:0]	4	Clock output enable for each PLL output port. This fuse setting is ORed with the corresponding Enable port signal to set the clock output enable control. Software sets this value automatically based upon the settings in the GUI. NOT RECOMMENDED to change this. xxx1 = Enable CLKOP. xx1x = Enable CLKOS. x1xx = Enable CLKOS2. 1xxx = Enable CLKOS3.	0001	Yes	Yes
MC1_LOCK_SEL[2:0]	10[6:4]	3	Lock-detector operation mode – normal or sticky. Sticky lock will stay high once lock is achieved until the PLL is reset or powered down. 000 = PLL Lock normal operation. 001 = PLL Lock sticky operation. 100 = alternate PLL Lock normal operation. Other values are not supported modes.	000	Yes	Yes
MC1_DYN_SOURCE	10[7]	1	Specify whether the Wishbone or external ports control the dynamic phase settings. 0 = Wishbone registers are in control. 1 = External Ports are in control.	1	Yes	Indirect
MC1_DIRECTION	11[1]	1	Only valid if MC1_DYN_PHASE=0. Specify direction of the dynamic phase change for MC1_ROTATE command. 0 = Phase rotates to a later phase. 1 = Phase rotates to an earlier phase.	0	Yes	N/A
MC1_ROTATE	11[0]	1	Only valid if MC1_DYN_PHASE=0. Command to start a change from current VCO phase to later/earlier phase. Phase changes on negative edge of MC1_ROTATE bit. Each step change represents a 45 degree change of VCO. (MC1_ROTATE is equivalent to the PHASESTEP signal.)	0	Yes	N/A
MC1_ DYN_SEL[1:0]	11[3:2]	2	Only valid if MC1_DYN_PHASE=0. Specifies which port is being controlled by dynamic phase controls. 00 = Enable CLKOS 01 = Enable CLKOS2 10 = Enable CLKOS3 11 = Enable CLKOP	00	Yes	N/A
MC1_TRIMOP_BYPASS_N	11[4]	1	Bypass the CLKOP output trim circuit. This setting selects whether to bypass the trim circuit. 0 = Bypass the trim circuit. 1 = Do not bypass the trim circuit.	0	Yes	Indirect
MC1_TRIMOS_BYPASS_N	11[5]	1	Bypass the CLKOS output trim circuit. This setting selects whether to bypass the trim circuit. 0 = Bypass the trim circuit. 1 = Do not bypass the trim circuit.	0	Yes	Indirect

Table 13-18. PLL Register Descriptions (Continued)

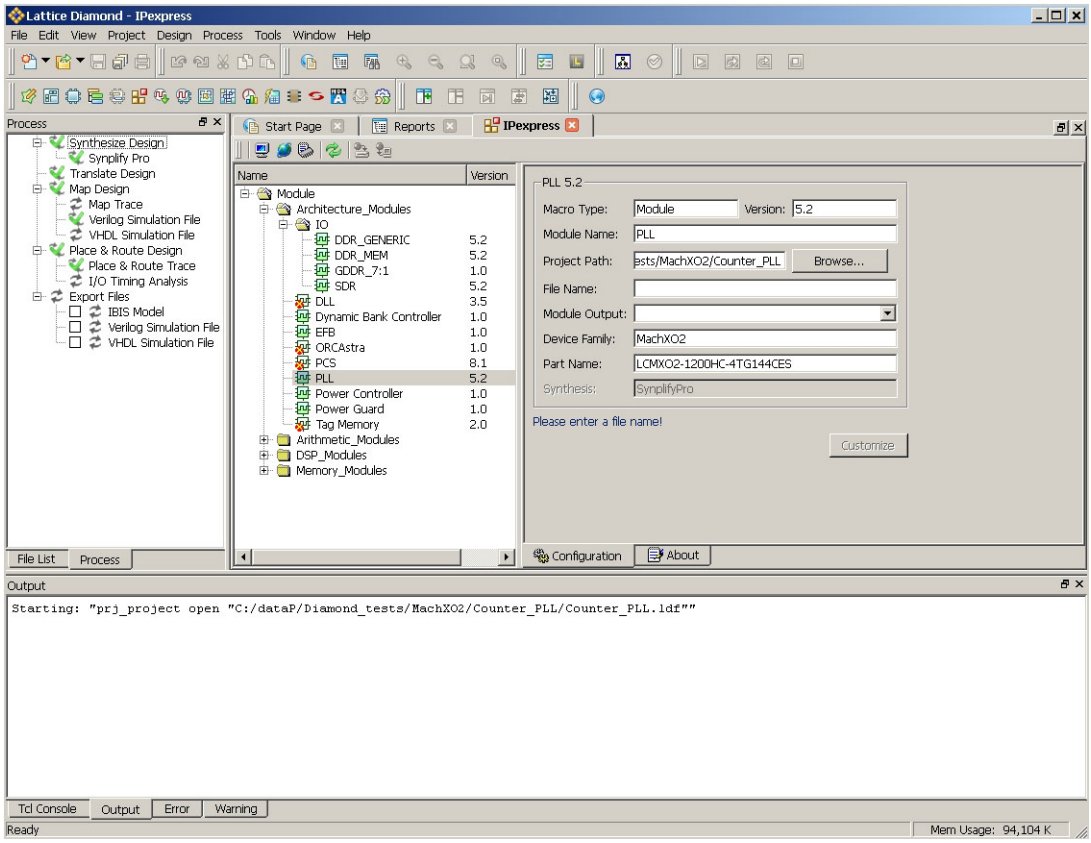
Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access	GUI Access
MC1_TRIMOS2_BYPASS_N	11[6]	1	Bypass the CLKOS2 output trim bits. There is not a trim control on CLKOS2. There is a dummy trim circuit used to equalize the delays between CLKOP, CLKOS, CLKOS2, & CLKOS3 outputs when trim is active on the CLKOP or CLKOS outputs. This setting selects whether to bypass the trim circuit. 0 = Bypass the trim circuit. 1 = Do not bypass the trim circuit.	0	Yes	Indirect
MC1_TRIMOS3_BYPASS_N	11[7]	1	Bypass the CLKOS3 output trim bits. There is not a trim control on CLKOS3. There is a dummy trim circuit used to equalize the delays between CLKOP, CLKOS, CLKOS2, & CLKOS3 outputs when trim is active on the CLKOP or CLKOS outputs. This setting selects whether to bypass the trim circuit. 0 = Bypass the trim circuit. 1 = Do not bypass the trim circuit.	0	Yes	Indirect

Appendix E. MachXO2 Device Usage with Lattice Diamond Design Software

When using the Lattice Diamond software with the MachXO2 device, there are a few minor differences from the screen shots shown in Figures 13-12 and 13-14 in this technical note. Figure 13-13 is the same in Diamond as it is in ispLEVER.

When configuring the PLL from Diamond using IPexpress, the user must supply a file name and also select the module output type as VHDL or Verilog. The module output type selection is made using the pull-down selection box. Figure 13-24 shows a Diamond example screen for this usage.

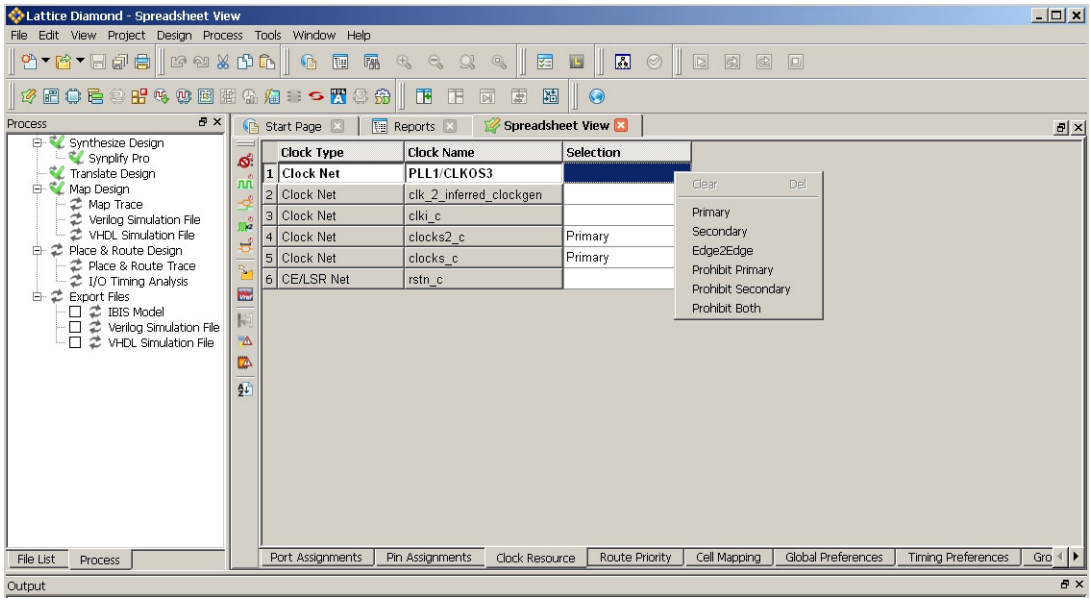
Figure 13-24. IPexpress Main Window for PLL Module Using Diamond



Once the file name and output type are filled in, clicking on the **Customize** button will open the Configuration tab window as shown in Figure 13-13.

When using Diamond to set the Clock preferences as Primary, Secondary, or Edge clocks, simply open the **Spreadsheet View** and select the **Clock Resource** tab. Then select the appropriate Clock preference from the pull-down menu by right-clicking in the selection window for the desired clock signal. Figure 13-25 shows a Diamond example screen for this usage.

Figure 13-25. Spreadsheet View for Clock Selection Using Diamond



MachXO2 Programming and Configuration Usage Guide

April 2013 Technical Note TN1204

Introduction

The MachXO2™ is an SRAM-based Programmable Logic Device that includes an internal Flash memory which makes the MachXO2 appear to be a non-volatile device. The MachXO2 provides a rich set of features for programming and configuration of the FPGA. You have many options available to you for building the programming solution that fits your needs. Each of the options available will be described in detail so that you can put together the programming and configuration solution that meets your needs.

MachXO2 Features

Key programming and configuration features of MachXO2 devices are:

- Instant-on configuration from internal Flash PROM – powers up in milliseconds
- Single-chip, secure solution
- Multiple programming and configuration interfaces:
  - 1149.1 JTAG
  - Self download
  - Slave SPI
  - Master SPI
  - Dual Boot
  - I^2C
  - WISHBONE bus
- User Flash Memory (UFM) for non-volatile data storage:
  - Configuration Flash memory overflow
  - EBR Initialization data
  - Application specific data
- Transparent programming of non-volatile memory
- Optional dual boot with external SPI memory
- Optional security bits for design protection

Definition of Terms

This document uses the following terms to describe common functions:

- **Programming**: Programming refers to the process used to alter the contents of the internal or external non-volatile configuration memory.
- **Configuration**: Configuration refers to a change in the state of the MachXO2 SRAM memory cells.
- **Configuration Mode**: The configuration mode defines the method the MachXO2 uses to acquire the configuration data from the non-volatile memory.
- **Configuration Data** – This is the data read from the non-volatile memory and loaded into the FPGA's SRAM configuration memory. This is also referred to as a bitstream, or device bitstream.
- **JEDEC** – The JEDEC file contains the configuration data programmed into the MachXO2 Configuration Flash, User Flash Memory, Feature Row, and Feature Bits. Format information is provided later in this technical note.
- **BIT** – The BIT file is the configuration data for the MachXO2 that is stored in an external SPI Flash. It is a binary file and is programmed unmodified into the SPI Flash.

- **Port** – A port refers to the physical connection used to perform programming and some configuration operations. Ports on the MachXO2 include JTAG, SPI, I^2C, and WISHBONE physical connections.
- **User Mode** – The MachXO2 is in user mode when configuration is complete, and the FPGA is performing the logic functions you have programmed it to perform.
- **Offline mode** – Offline mode is a term that is applied to both non-volatile memory programming and SRAM configuration. When using offline mode programming/configuration the FPGA no longer operates in user mode. The contents of the non-volatile or SRAM configuration memory are updated, but the MachXO2 does not perform your logic operations until offline mode programming/configuration is complete.
- **Transparent Mode** – Transparent mode is used to update the Configuration Flash, and User Flash Memory while leaving the MachXO2 in User Mode.
- **Number Formats** – The following nomenclature is used to denote the radix of numbers
  - 0x: Numbers preceded by '0x' are hexadecimal
  - b (suffix): Numbers suffixed with 'b' are binary
  - All other numbers are decimal
- **Internal Flash Memory** – JED file or bit file can be programmed directly into the internal flash sector. User does not need to know where an actual page of the configuration data starts. The MachXO2 configuration engine handles the parsing in the flash to SRAM transfer.

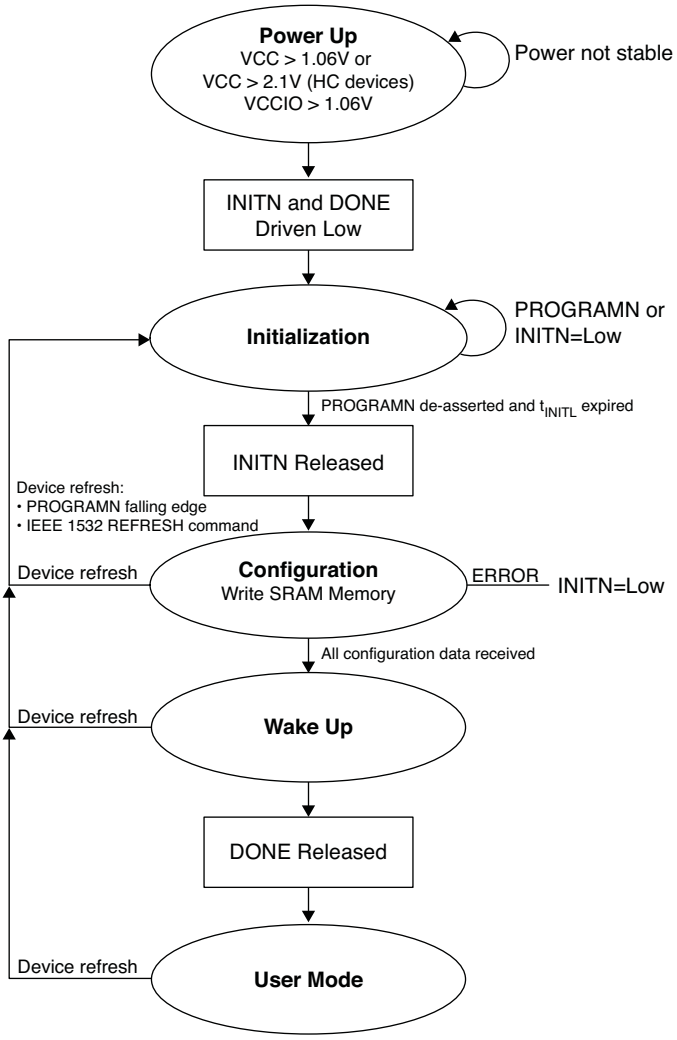
Configuration Details

MachXO2 devices contain two types of memory, SRAM and Flash. SRAM memory contains the active configuration, essentially the "fuses" that define the behavior of the FPGA. The active configuration is, in most cases, retrieved from a non-volatile memory. The non-volatile memory holds the configuration data that is loaded into the FPGAs SRAM. The MachXO2 provides an internal Flash memory that stores the configuration data loaded into the MachXO2 SRAM.

Configuration Process and Flow

Prior to becoming operational, the FPGA goes through a sequence of states, including initialization, configuration and wake-up.

Figure 14-1. Configuration Flow



The MachXO2 sysCONFIG ports provide industry standard communication protocols for programming and configuring the FPGA. Each of the protocols shown in Table 14-1 provides a way to access the MachXO2 device's internal Flash memory, or to load its configuration SRAM. The Memory Space Accessibility section provides information about the capabilities of each sysCONFIG port.

The sysCONFIG ports capable of accessing the Flash memory have a priority order. Table 14-1 lists each of the sysCONFIG ports in their priority order. The MSPI configuration port does not have the ability to alter the Flash memory space, and as a result is not a factor in the sysCONFIG port priority scheme. The priority scheme is important to be aware of, as a Configuration Logic operation using a low priority sysCONFIG port can be interrupted by a higher priority sysCONFIG port. The operation of the Configuration Logic is not defined when a low priority sysCONFIG port is interrupted by a higher priority sysCONFIG port. Do not permit simultaneous access to the Configuraiton Logic using a sysCONFIG port.

Power-up Sequence

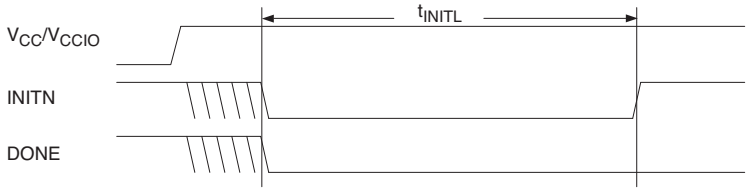
In order for the MachXO2 to operate, power must be applied to the device. During a short period of time, as the voltages applied to the system rise, the FPGA will have an indeterminate state.

As power continues to ramp, a Power On Reset (POR) circuit inside the FPGA becomes active. The POR circuit, once active, makes sure the external I/O pins are in a high-impedance state. It also monitors the V_{CC} and V_{CCIO0} input rails. The POR circuit waits for the following conditions:

- V_{CC} > 1.06V (or 2.1V for HC devices)
- V_{CCIO0} > 1.06V

When these conditions are met the POR circuit releases an internal reset strobe, allowing the device to begin its initialization process. The MachXO2 asserts INITN active low, and drives DONE low. When INITN and DONE are asserted low the device moves to the initialization state, as shown in Figure 14-1.

Figure 14-2. Configuration from Power-On-Reset Timing



Initialization

The MachXO2 enters the memory initialization phase immediately after the Power On Reset circuit drives the INITN and DONE status pins low. The purpose of the initialization state is to clear all of the SRAM memory inside the FPGA.

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

- The t_{INITL} time period has elapsed
- The PROGRAMN pin is deasserted
- The INITN pin is no longer asserted low by an external master

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

During the t_{INITL} time period the FPGA is clearing the configuration SRAM. When the MachXO2 is part of a chain of devices each device will have different t_{INTIL} initialization times. The FPGA with the slowest t_{INTIL} parameter can prevent other devices in the chain from starting to configure. Premature release of the INITN in a multi-device chain may cause configuration of one or more chained devices to fail to configure intermittently.

The active-low, open-drain initialization signal INITN must be pulled high by an external resistor when initialization is complete. To synchronize the configuration of multiple FPGAs, one or more INITN pins should be wire-ANDed. If one or more FPGAs or an external device holds INITN low, the FPGA remains in the initialization state.

Configuration

The rising edge of the INITN pin causes the FPGA to enter the configuration state. The FPGA is able to accept the configuration bitstream created by the Diamond development tools.

The MachXO2 begins fetching configuration data from non-volatile memory. The memory used to configure the MachXO2 is either the internal Flash, or an external SPI Flash. The MachXO2 does not leave the Configuration

state if there are no memories with valid configuration data. It is necessary to program the non-volatile memory internal or attached to the FPGA, or to program it using the JTAG port.

During the time the FPGA receives its configuration data the INITN control pin takes on its final function. INITN is used to indicate an error exists in the configuration data. When INITN is high, configuration proceeds without issue. If INITN is asserted low, an error has occurred and the FPGA will not operate.

Wake-up

Wake-up is the transition from configuration mode to user mode. The MachXO2's fixed four-phase wake-up sequence starts when the device has correctly received all of its configuration data. When all configuration data is received, the FPGA asserts an internal DONE status bit. The assertion of the internal DONE causes a Wake Up state machine to run that sequences four controls. The four control strobes are:

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

The first phase of the Wake-Up process is for the MachXO2 to release the Global Output Enable. When it is asserted, permits the FPGA's I/O to exit a high-impedance state and take on their programmed output function. The FPGA inputs are always active. The input signals are prevented from performing any action on the FPGA flip-flops by the assertion of the Global Set/Reset (GSR).

The second phase of the Wake-Up process releases the Global Set/Reset and the Global Write Disable controls.

The Global Set/Reset is an internal strobe that, when asserted, causes all I/O flip-flops, Look Up Table (LUT) flip-flops, distributed RAM output flip-flops, and Embedded Block RAM output flip-flops that have the **GSR enabled** attribute to be set/cleared per their hardware description language definition.

The Global Write Disable is a control that overrides the write enable strobe for all RAM logic inside the FPGA. The inputs on the FPGA are always active, as mentioned in the Global Output Enable section. Keeping GWDIS asserted prevents accidental corruption of the instantiated RAM resources inside the FPGA.

The last phase of the Wake-Up process is to assert the external DONE pin. The external DONE is a bi-directional, open-drain I/O only when it is enabled. An external agent that holds the external DONE pin low prevents the wake-up process of the MachXO2 from proceeding. Only after the external DONE, if enabled, is active high does the final wake-up phase complete. Wake-Up completes uninterrupted when the external DONE pin is not enabled.

Once the final wake-up phase is complete, the FPGA enters user mode.

User Mode

The MachXO2 enters User Mode immediately following the Wake-Up sequence has completed. User Mode is the point in time when the MachXO2 begins performing the logic operations you designed. The MachXO2 remains in this state until one of three events occurs:

- The PROGRAMN input pin is asserted
- A REFRESH command is received via one of the configuration ports
- Power is cycled

Clearing the Configuration Memory and Re-initialization

The current user mode configuration of the MachXO2 remains in operation until it is actively cleared, or power is lost. Several methods are available to clear the internal configuration memory of the MachXO2. The first is to

remove power and reapply power. Another method is to toggle the PROGRAMN pin. Lastly you can reinitialize the memory through a Refresh command. Any active configuration port can be used to send a Refresh command.

- Assertion of the PROGRAMn input
- Cycling power to the MachXO2
- Sending the Refresh command using a configuration port

Invoking one of these methods causes the MachXO2 to drive INITN and DONE low. The MachXO2 enters the initialization state as described earlier.

Memory Space Accessibility

The two internal memories, Flash and SRAM, of the MachXO2 have the ability to be read and written. Each port on the MachXO2 has a different level of access to each memory space. Table 14-2 provides a cross-reference of the MachXO2 ports and the memory space they can access.

As can be seen from Table 14-1, the JTAG port has the ability to read and write both of the internal memory spaces. No other port has ability to read the SRAM configuration memory. The JTAG port has the ability to access the two memory spaces in either Offline or Transparent mode. Every other port has some limitation on the functions that can be performed.

Table 14-1. Memory Space Accessibility of Different Ports

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

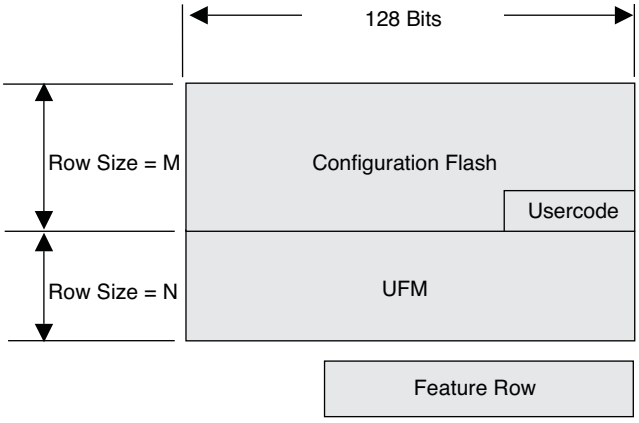
1. In Transparent mode only.
2. See "Clearing the Configuration Memory and Re-initialization" on page 14-5.

Bitstream/PROM Sizes

The MachXO2 is a SRAM based FPGA. The SRAM configuration memory must be loaded from a non-volatile memory that can store all of the configuration data. The size of the configuration data is variable. It is based on the amount of logic available in the FPGA, and the number of pre-initialized Embedded Block RAM (EBR) components. A MachXO2 design using the largest device, with every EBR pre-initialized with unique data values, and generated without compression turned on requires the largest amount of storage.

Storing configuration data in the MachXO2's internal Flash memory has special considerations. The Flash memory in the MachXO2 provides three independent sectors. The first sector is dedicated for use in holding compressed configuration data, and is called Configuration Flash. The second sector, called the User Flash Memory, provides three different functions. It provides additional Configuration Flash storage for large configuration data images, it can store EBR contents, or it is available for use as general purpose Flash memory. The third sector is the Feature Row.

Figure 14-3. Flash Memory Space of a MachXO2 Device



The Configuration Flash is, for most designs, large enough to store the compressed configuration data that is loaded into the SRAM configuration memory. However, as the amount of logic in the design increases, and the amount of pre-initialized EBR increases, the size of the configuration data also increases. The increase in size can cause the configuration data to overflow into the UFM sector. It is also possible, but unlikely, that the configuration data can get too large for the internal Flash memory altogether. In the event configuration data grows too large to fit in the combined Configuration Flash/UFM memory space the design needs to be modified so that it is smaller, or an external configuration memory must be used. You can provide input to the software generating the configuration data to prevent the overflow into the UFM.

In the event the configuration data is too large for the combined Configuration Flash and UFM memory you can store the device bitstream in an external SPI Flash. Table 14-2 shows the maximum uncompressed bitstream sizes allowing you to select a SPI Flash.

Table 14-2. Maximum Configuration Bits

Device	Uncompressed Bitstream Size Without EBR	Uncompressed Bitstream Size With EBR	Maximum Internal Flash	Units
LCMXO2-256	0.09	N/A	0.071	Mb
LCMXO2-640	0.19	0.20	0.17	Mb
LCMXO2-640U			0.33	Mb
LCMXO2-1200	0.35	0.41	0.33	Mb
LCMXO2-1200U			0.47	Mb
LCMXO2-2000	0.51	0.58	0.47	Mb
LCMXO2-2000U			0.80	Mb
LCMXO2-4000	0.93	1.02	0.80	Mb
LCMXO2-7000	1.47	1.70	1.38	Mb

Feature Row

The MachXO2 includes a Feature Row that is used to control FPGA resources. For example, the Feature Row is used to determine how the MachXO2 SRAM configuration memory is loaded. In other FPGAs this operation is controlled using external I/O pins. The Feature Row permits more flexibility in selecting the functions available for configuration, increases the number of available I/O on the device, and eliminates the need to make changes to your hardware.

Feature Row can be erased or programmed independently. When Feature Row is erased, Feature Row will set its value back to HW default Mode state.

A full list of the functions controlled by the Feature Row and their default values are shown in Table 14-3.

Table 14-3. MachXO2 Feature Row Elements

Feature	SW Default Mode State (Programmed)	HW Default Mode State (Erased)
PROGRAMN Persistence	Disabled	Enabled
INITn Persistence	Disabled	Disabled
DONE Persistence	Disabled	Disabled
Custom IDCODE	0x00000000	0x00000000
TraceID™	00000000	00000000
Security[1]	OFF	OFF
JTAG Port Persistence	Enabled	Enabled
SSPI Port Persistence	Disabled	Enabled
I2C Port Persistence	Disabled	Enabled
MSPI Port Persistence	Disabled	Disabled
I2C Slave Address[2]	0x00	0x00
UFM OTP	OFF	OFF
SRAM OTP	OFF	OFF
Config Flash OTP	OFF	OFF
my_ASSP Enable	OFF	OFF

1. Enabled/disabled using the CONFIG_SECURE preference.
2. Address is assigned in IPexpress™.

It is strongly recommended that the Feature Row only be modified during development, and rarely, if ever, upgraded in the field. The reason for this recommendation is the Feature Row is responsible for controlling the availability of the Configuration Ports. It is possible to cause active Configuration Ports to become unavailable, preventing future updates.

Changing the Feature Row can also prevent the MachXO2 from configuring. The PROGRAMN, INITN, and DONE control and status pins are enabled and disabled using the Feature Row. The PROGRAMN input pin may be recovered for use as a general purpose I/O. Erasing Feature Row state causes the PROGRAMN input to act as PROGRAMN, not as a general purpose I/O. If the general purpose I/O is driven active low the MachXO2 will never be allowed to complete its configuration process.

Feature Row can be erased or altered by Diamond Programmer. It will be erased and reprogrammed during Flash erase, program and verify sequence, both offline and online. During offline flash programming, if you do not want Feature Row to be erased and reprogrammed, Lattice recommends that you use "XFLASH Erase, Program, Verify, Refresh" operation.

Feature Row settings can be altered using the Diamond Spreadsheet View. Spreadsheet View allows you to edit the configuration settings for the MachXO2, and then saves your settings in the Lattice Preference File (LPF). These settings are applied to the MachXO2 configuration data during the Map, Place, and Route build phases.

Key Features

- Not intended to be modified in the field; only for development.
- Change in Feature Row settings may cause active configuration ports to become unavailable.
- Can be altered using Diamond Programmer or Diamond Spreadsheet View.
- Will be erased and re-programmed during Flash updates. So keep Feature Row contents consistent.

Configuration Modes

The MachXO2 configuration SRAM memory must be loaded with valid configuration data before the FPGA will operate. The MachXO2 provides only four methods of getting the configuration data into the SRAM memory. Each of these methods has its own set of advantages. The four methods available are shown in Table 14-4.

Table 14-4. Configuration Modes

Mode	Number of Pins	Max. Frequency
1149.1 JTAG	4 (5)	25 MHz
Self-Download Mode	0	N/A
External Download	4	50 MHz
Dual Boot Download	0/4	N/A / 50 MHz

The primary configuration mode, for a majority of MachXO2 designs, is Self-Download Mode. It has an advantage in configuration speed because the internal configuration clock runs at frequencies higher than can be applied to an external memory. It does not require an extra PROM, which increases the cost of your product. It does not rely on an external programmer to load the SRAM using the JTAG port.

The External Download mode's advantage is that it makes all of the User Flash Memory available for your use. You do not have to be concerned about the Configuration Flash image overflowing into the UFM, or overflowing the available internal Flash memory.

The Dual Boot mode's advantage is the MachXO2 configures more reliably. The MachXO2 loads the internal Flash memory image first, and should that fail, a fail-safe configuration data image can be downloaded into the MachXO2's SRAM, allowing the FPGA to continue to operate. A failed reprogramming of the internal memory, usually as a result of a loss of power, is the primary reason MachXO2 would fail to configure.

The JTAG port's advantage is that it provides the widest set of functions and features for programming, configuring, and testing the MachXO2 system.

sysCONFIG™ Ports

Table 14-5. MachXO2 Programming and Configuration Ports

Interface	Port	Description
JTAG	JTAG (IEEE 1149.1 and IEEE 1532 compliant)	4-wire or 5-wire JTAG Interface
sysCONFIG	SSPI	Slave Serial Peripheral Interface (SPI)
	MSPI	Master Serial Peripheral Interface (SPI)
	I^2C	Inter-integrated Circuit (I^2C) Interface
Internal	WISHBONE	Internal WISHBONE bus interface

sysCONFIG Pins

The MachXO2 provides a set of sysCONFIG I/O pins that you use to program and configure the FPGA. The sysCONFIG pins are grouped together to create ports (i.e. JTAG, SSPI, I^2C, MSPI) that are used to interact with the FPGA for programming, configuration, and access of resources inside the FPGA. The sysCONFIG pins in a configuration port group may be active, and used for programming the FPGA, or they can be reconfigured to act as general purpose I/O.

Recovering the configuration port pins for use as general purpose I/O requires you to adhere to the following guidelines:

- You must DISABLE the unused port. You can accomplish this by using the Diamond Spreadsheet View's Global Preferences tab. Each configuration port is listed in the sysCONFIG options tree.

- You must prevent external logic from interfering with device programming. Make sure that recovered sysCONFIG pins are not asserted when the MachXO2 is in Feature Row HW Default Mode state. One example is driving PROGRAMN with an active low signal after the MachXO2 is in Feature Row HW Default Mode state. Failure to reprogram the Feature Row with PROGRAMN disabled prevents the FPGA from configuring and entering user mode.
- Use care when using JTAGENB to selectively enable and disable the JTAG port. Any external logic connected to the JTAG I/O must not contend with the JTAG programming port.

Table 14-6 lists the default state of the shared sysCONFIG pins. As you can see, a Default Mode Feature Row device has the JTAG, SPI Slave and I^2C ports enabled. Upon entry to User Mode the MachXO2, the default state of the SSPI, and I^2C sysCONFIG pins become general purpose I/O. This means you lose the ability to program the MachXO2 using SSPI, or I^2C when using the default sysCONFIG port settings. To retain the SSPI, or I^2C sysCONFIG pins in user mode, be sure to ENABLE them using the Diamond Spreadsheet View editor.

Unless specified otherwise, the sysCONFIG pins are powered by the VCCIO0 voltage. It is crucial you take this into consideration when provisioning other logic attached to Bank 0.

The function of each sysCONFIG pin is described in detail.

Table 14-6. Default State of the sysCONFIG Pins

Pin Name	Associated sysCONFIG Port	Default Pin Function in Default Mode Feature Row (Configuration Mode)	Pin Direction (Configuration Mode)	Default Function in User Mode Feature Row
PROGRAMN[1]	SDM	PROGRAMN	Input with weak pull up	User-defined I/O
INITN	SDM	I/O	I/O with weak pull up	User-defined I/O
DONE	SDM	I/O	I/O with weak pull up	User-defined I/O
MCLK/CCLK	SSPI/MSPI	SSPI	Input with weak pull up	User-defined I/O
SN	SSPI/MSPI	SSPI	Input with weak pull up	User-defined I/O
SI/SISPI	SSPI/MSPI	SSPI	Input	User-defined I/O
SO/SPISO	SSPI/MSPI	SSPI	Output	User-defined I/O
CSSPIN	MSPI	I/O	I/O with weak pull up	User-defined I/O
SCL	I^2C	I^2C	Bi-Directional	User-defined I/O
SDA	I^2C	I^2C	Bi-Directional	User-defined I/O

1. The default for PROGRAMN in user mode was PROGRAMN in ispLEVER® 8.1 SP1 and Lattice Diamond® 1.1.

Self Download Port Pins

PROGRAMN: The PROGRAMN is an input used to configure the FPGA. The PROGRAMN pin, when enabled, is sensitive to a high-to-low transition, and has an internal weak pull-up. When PROGRAMN is asserted low, the FPGA exits user mode and starts a device configuration sequence at the Initialization phase, as described earlier. Holding the PROGRAMN pin low prevents the MachXO2 from leaving the Initialization phase. The PROGRAMN has a minimum pulse width assertion period in order for it to be recognized by the FPGA. You can find this minimum time in the MachXO2 Family Data Sheet in the AC timing section.

Be aware of the following special cases when the PROGRAMN pin is active:

- If the device is currently being programmed via JTAG then PROGRAMN will be ignored until the JTAG mode programming sequence is complete.
- Toggling the PROGRAMN pin during device configuration will interrupt the process and restart the configuration cycle.
- Asserting PROGRAMN on a device in Feature Row HW Default Mode state disables the SSPI and I^2C ports. Start SSPI or I^2C programming operations after PROGRAMN is deasserted.

- PROGRAMN is active during power-up, even when PROGRAMN has been reserved as a general purpose I/O. Do not allow any input signal attached to PROGRAMN to transition from high to low at a frequency greater than the VCC (min) to INITN rising edge time period. High to low PROGRAMN assertions more frequently prevent the MachXO2 from configuring, causing the FPGA to remain in a continuous RESET condition. See Figure 14-4.

Figure 14-4. Period PROGRAMN is Always Observed

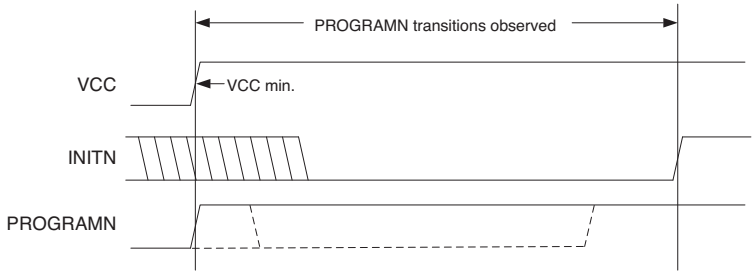
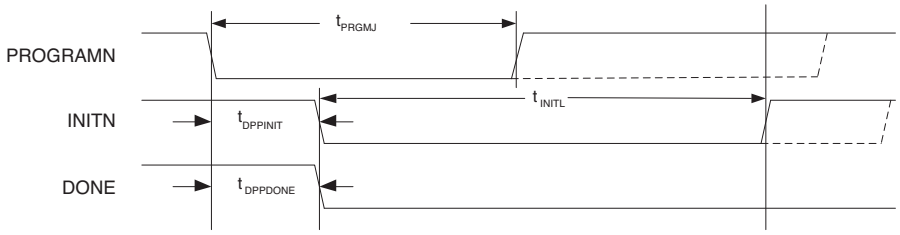


Figure 14-5. Configuration from PROGRAMN Timing



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

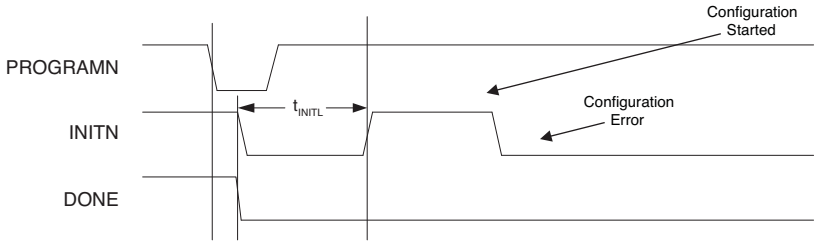
- After power is applied, after a PROGRAMN assertion, or a REFRESH command it goes low to indicate the SRAM configuration memory is being erased. The low time assertion is specified with the t_{INTIL} parameter.
- After the t_{INTIL} time period has elapsed the INITn pin is deasserted (i.e. is active high) to indicate the MachXO2 is ready for its configuration bits. The MachXO2 begins loading configuration data from either the internal Flash memory or an external SPI Flash.
- INITn can be asserted low by an external agent before the t_{INTIL} time period has elapsed in order to prevent the FPGA from reading configuration bits. This is useful when there are multiple programmable devices chained together. The programmable device with the longest t_{INTIL} time can hold all other devices in the chain from starting to get data until it is ready itself.
- The last function provided by INITn is to signal an error during the time configuration data is being read. Once t_{INTIL} has elapsed and the INITn pin has gone high, any subsequent INITn assertion signals the MachXO2 has detected an error during configuration.

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

- Power has just been applied
- PROGRAMN falling edge occurred
- The IEEE 1532 REFRESH command has been sent using a slave configuration port (JTAG, SSPI, I^2C or WISHBONE).

If the INITN pin is asserted due to an error condition, the error can be cleared by correcting the configuration bitstream and forcing the FPGA into the Initialization state.

Figure 14-6. Configuration Error Notification



The INITN pin of a MachXO2 device is not visible external to the device when in the Feature Row HW Default Mode state. The INITN pin, when in this mode, is pulled high by default. The INITN behavior described in Figure 14-6 is only visible outside the MachXO2 when the INITN pin is enabled.

The INITN can be recovered as a general purpose I/O. By default, the INITN pin is disabled. You can use the Diamond Spreadsheet View to enable it.

If an error is detected when reading the bitstream, INITN will go low, the internal DONE bit will not be set, the DONE pin will stay low, and the device will not wake up. The device will fail configuration when the following happens:

- The bitstream CRC error is detected
- The invalid command error detected
- A time out error is encountered when loading from the on-chip Flash
- The program done command is not received when the end of on-chip SRAM configuration or on-chip Flash memory is reached

DONE: The DONE pin is a bi-directional open drain with a weak pull-up that signals the FPGA is in User mode. DONE is first able to indicate entry into User mode only after an internal DONE bit is asserted. The internal DONE bit defines the beginning of the FPGA Wake-Up state.

The DONE output pin is controlled by the SDM_PORT configuration parameter that is modified in the Diamond Spreadsheet View. By default the DONE pin is a general purpose I/O when the MachXO2 is in the Feature Row HW Default Mode state. The default mode causes the MachXO2 to automatically sequence through the Wake-Up sequence after the internal DONE bit is asserted. The FPGA does not stall waking up waiting for the DONE pin to be asserted high.

The FPGA can be held from entering User mode indefinitely by having an external agent keep the DONE pin asserted low. In order to use DONE to stall entering User mode the SDM_PORT must enable the DONE I/O, and the FPGA Feature Row must be programmed. A common reason for keeping DONE driven low is to allow multiple FPGAs to be completely configured. As each FPGA reaches the DONE state, it is ready to begin operation. The last FPGA to configure can cause all FPGAs to start in unison.

The DONE pin drives low in tandem with the INITN pin when the FPGA enters Initialization mode. As described earlier, this condition happens when power is applied, PROGRAMN is asserted, or an IEEE 1532 Refresh command is received via an active configuration port.

Sampling the DONE pin is a way for an external device to tell if the FPGA has finished configuration. However, when using IEEE 1532 JTAG to configure SRAM the DONE pin is driven by a boundary scan cell, so the state of the DONE pin has no meaning during IEEE 1532 JTAG configuration (once configuration is complete, DONE takes on the behavior defined by the SDM_PORT setting in the Feature Row). The DONE pin is also pulled high when the FPGA is in the Feature Row HW Default Mode state. This behavior can make a part appear to be successfully configured to other logic monitoring the DONE pin.

Master and Slave SPI Configuration Port Pins

Table 14-7. Master SPI Configuration Port Pins

Pin Name	Function	Direction	Description
MCLK/CCLK	MCLK	Output with weak pullup	Master clock used to time data transmission/reception from the MachXO2 Configuration Logic to a slave SPI PROM. A 1K pull-up resistor is recommended on MCLK for External and Dual Boot configuration modes.
CSSPIN	CSSPIN	Output	Chip select used to enable an external SPI PROM containing configuration data
SI/SISPI	SISPI	Output	SISPI carries output data from the MachXO2 Configuration Logic to the slave SPI PROM
SO/SPISO	SPISO	Input	SPISO carries output data from the slave SPI PROM to the MachXO2 Configuration Logic
SN	SN/IO	Input	MachXO2 Configuration Logic slave SPI chip select input. Pull high externally whenever the MSPI port is active.

Table 14-8. Slave SPI Configuration Port Pins

Pin name	Function	Direction	Description
MCLK/CCLK	CCLK	Input with weak pullup	Clock used to time data transmission/reception from an external SPI master device to the MachXO2 Configuration Logic.
SI/SISPI	SI	Input	SI carries output data from the external SPI master to the MachXO2 Configuration Logic
SO/SPISO	SO	Input	SO carries output data from the MachXO2 Configuration Logic to the external SPI master
SN	SN	Input with weak pullup	MachXO2 Configuration Logic slave SPI chip select input. SN is an active low input.

MCLK/CCLK: The MCLK/CCLK, when active, are clocks used to sequentially load the configuration data for the FPGA. The pin functions as:

The MCLK/CCLK pin's default state for a MachXO2 in the Feature Row HW Default Mode state is to act as the configuration clock (i.e., CCLK). This allows an external Slave SPI master controller to program the MachXO2. The maximum CCLK frequency and the data setup/hold parameters can be found in the AC timing section of the MachXO2 Family Data Sheet. The Feature Row must be configured to ENABLE the Slave SPI Port if you want to use the port to reprogram the MachXO2 after it enters user mode.

The MCLK/CCLK pin functions as a Master Clock (MCLK) when the MachXO2 is configured in Dual Boot or External Boot modes. A 1K pull-up resistor is recommended when using these modes. The MCLK becomes an output and provides a reference clock for a SPI Flash attached to the MachXO2's Master SPI Configuration port. MCLK actively drives until all of the configuration data has been received. When the MachXO2 enters user mode the MCLK output tri-states. This allows the MCLK to become a general purpose I/O. The MCLK is reserved for use, in most post-configuration applications, as the reference clock for performing memory transactions with the external SPI PROM.

The MachXO2 generates MCLK from an internal oscillator. The initial frequency of the MCLK is nominally 2.08 MHz. The MCLK frequency can be altered using the MCCLK_FREQ parameter. You can select the MCCLK_FREQ using the Diamond Spreadsheet View. For a complete list of the supported MCLK frequencies, see Table 14-9.

Table 14-9. MachXO2 MCLK Valid Frequencies (MHz)

2.08	9.17	33.25
2.46	10.23	38.00
3.17	13.30	44.33
4.29	14.78	53.20
5.54	20.46	66.50
7.00	26.60	88.67
8.31	29.56	133.00

During the initial stages of device configuration the frequency value specified using MCCLK_FREQ is loaded into the FPGA. Once the MachXO2 accepts the new MCLK_FREQ value the MCLK output begins driving the selected frequency. Make certain when selecting the MCLK_FREQ that you do not exceed the frequency specification of your configuration memory, or of your PCB. Review the MachXO2 AC specifications in the MachXO2 Family Data Sheet when making MCLK_FREQ decisions.

SN: The SN pin is the Slave SPI ports chip select. An external SPI bus master asserts the SN pin active low in order to perform actions using the MachXO2's programming and configuration logic. The SN pin is available when the MachXO2 is in the Feature Row HW Default Mode state, and in user mode when the Slave SPI port is set to the ENABLE setting. The SN pin is a general purpose I/O in user mode when the Slave SPI port is set to the DISABLE setting.

Proper operation of the MachXO2 depends upon maintaining the SN pin in the correct state:

- SN must be deasserted (i.e. held high) when configuring using Master SPI mode
- SN must be deasserted when the MachXO2 is in user mode, and SPI memory transactions are initiated using the internal WISHBONE bus
- SN must be deasserted when accessing the Configuration Logic in the MachXO2 using I^2C
- When SN is asserted, CSSPIN must be deasserted. Deasserting CSSPIN places the shared SPI pins into a high impedance state.
  - The Master SPI port and the Slave SPI port share three common pins, SI/SISPI, SO/SPISO, and MCLK/CCLK. The MachXO2 permits both ports to be available at the same time. They are not permitted to be accessed at the same time. The Slave SPI and the Master SPI port must be time multiplexed when both ports are enabled.

Lattice recommends the SN pin be pulled high externally to augment the weak internal pull-up.

CSSPIN: The CSSPIN pin is an active low chip select used by the Master SPI configuration mode to enable an external SPI Flash. When the MachXO2 is programmed to configure in either External or Dual Boot mode the CSSPIN pin is asserted to the attached SPI Flash. The MachXO2 asserts CSSPIN until all configuration data bytes have been loaded, at which time the CSSPIN enters a high impedance state.

When the MachXO2 is in the Feature Row HW Default Mode state the CSSPIN is a general purpose I/O with a weak pulldown. It must have an external pullup resistor when the External and Dual Boot configuration modes are used. CSSPIN must ramp in tandem with the SPI PROM VCC input. It remains a general purpose I/O when the FPGA enters user mode. You must ENABLE the Master SPI port to reserve CSSPIN for use by the internal SPI Master logic.

When configuring from an external SPI Flash, ensure that the SPI Flash V_{CC} and the MachXO2 V_{CCIO2} are at the same level. Ensure that the SPI Flash V_{CC} meets is at the recommended operating level.

Some SPI PROM manufacturers require the chip select input of the PROM ramp in unison to the PROMs VCC rail. The CSSPIN pin, by default, has a weak pull-down resistor internally. Adding a 4.7K to 10K ohm pull-up resistor to the CSSPIN pin on the MachXO2 is recommended.

SI/SISPI: The SI/SISPI is a dual function bi-directional pin. The direction depends upon whether a Master or Slave mode is active. The SI/SISPI is an input data pin when using the Slave SPI mode and is an output data pin when using the Master SPI mode. In Master SPI mode, the MachXO2 drives SI/SISPI until all configuration data bytes have been loaded, at which time the SI/SISPI enters a high impedance state.

At least one of the sysCONFIG preferences, SLAVE_SPI_PORT or MASTER_SPI_PORT, must be set to ENABLE in order to preserve this pin as SI/SISPI and allow access to the SPI interface.

SO/SPISO: The SO/SPISO pin is a dual function bi-directional pin. The direction depends upon whether a Master or Slave mode is active. The SO/SPISO is an input data pin when using the Master SPI mode and is an output data pin when using the Slave SPI mode.

At least one of the sysCONFIG preferences, SLAVE_SPI_PORT or MASTER_SPI_PORT, must be set to ENABLE in order to preserve this pin as SO/SPISO and allow access to the SPI interface.

I^2C Configuration Port Pins

SCL: The MachXO2 provides an I^2C configuration port. The SCL is the I^2C Serial Clock pin, and is used to initiate and time transactions on the I^2C bus. It is a bi-directional, open-drain signal that is an output when the MachXO2 I^2C controller is mastering transactions on the bus, and is an input when an external I^2C master is accessing resources inside the MachXO2. SCL requires an external pull-up resistor in order to operate.

The SCL pin is available when the MachXO2 is in the Feature Row HW Default Mode state. You must ENABLE the I2C_PORT for the I^2C port to continue to be available in user mode. The SCL pin becomes a general purpose I/O if you do not ENABLE the I2C_PORT.

SDA: The SDA pin is the I^2C serial data input/output pin. It is bi-directional, open-drain, and requires an external pull-up resistor in order to operate. The pin changes direction dynamically during data transactions on the I^2C bus. The current state depends on the current bus master and the operation being performed by that master.

The SDA pin is available when the MachXO2 is in the Feature Row HW Default Mode state. You must ENABLE the I2C_PORT if you want the I^2C port to continue to be available in user mode. The SDA pin becomes a general purpose I/O if you do not ENABLE the I2C_PORT.

JTAG Configuration Port Pins

The JTAG pins provide a standard IEEE 1149.1 Test Access Port (TAP). The JTAG port is the only configuration port on the MachXO2 that is capable of performing configuration, programming, and multi-device configuration functions. Programming and configuration over the JTAG port uses IEEE 1532 compliant commands. In addition to the IEEE 1532 capabilities, the MachXO2 provides all of the mandatory IEEE 1149.1 Test Access Port commands allowing printed circuit board assembly verification.

The JTAG port is enabled by default when the MachXO2 is in the Feature Row HW Default Mode state. Like all of the other configuration port pins the JTAG pins can become general purpose I/O. Unlike the other ports, the default state for the JTAG port is to remain active in user mode (i.e. ENABLE state). The JTAG pins can be recovered to be general purpose I/O by setting the JTAG_PORT preference to the DISABLE state. It is recommended the JTAG port remain dedicated programming pins.

The JTAG port, when set in the DISABLE state, enables the JTAGENB input. JTAGENB permits the JTAG pins to be multiplexed. Asserting JTAGENB high causes the JTAG pins to take on the IEEE 1149.1 personality. De-asserting JTAGENB (i.e. driven low) causes the JTAG port pins to become general purpose I/O. Design the JTAG port circuitry carefully when taking advantage of JTAG port pin multiplexing. Avoid bus contention between logic attached to the JTAG port.

When the device is programmed through IEEE 1149.1 control, the sysCONFIG programming pins, such as DONE, cannot be used to determine programming progress. This is because the state of the boundary scan cell will drive the pin, per the IEEE JTAG standard, rather than normal internal logic.

Table 14-10. JTAG Port Pins

Pin Name	Pin Function (Configuration Mode)	Pin Direction (Configuration Mode)	Default Function (User Mode)
TDI	TDI	Input with weak pull-up	TDI
TDO	TDO	Output with weak pull-up	TDO
TCK	TCK	Input	TCK
TMS	TMS	Input with weak pull-up	TMS
JTAGENB	I/O	Input/output with weak pull-down	I/O

TDO: The Test Data Output (TDO) pin is used to shift out serial test instructions and data. When TDO is not being driven by the internal circuitry, the pin will be in a high impedance state. The only time TDO is not in a high impedance state is when the JTAG state machine is in the Shift IR or Shift DR state. This pin should be wired to TDO of the JTAG connector, or to TDI of a downstream device in a JTAG chain. An internal pull-up resistor on the TDO pin is provided. The internal resistor is pulled up to VCCIO Bank 0.

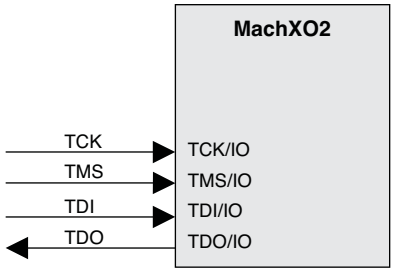
TDI: The Test Data Input (TDI) pin is used to shift in serial test instructions and data. This pin should be wired to TDI of the JTAG connector, or to TDO of an upstream device in a JTAG chain. An internal pull-up resistor on the TDI pin is provided. The internal resistor is pulled up to VCCIO of Bank 0.

TMS: The Test Mode Select (TMS) pin is an input pin that controls the progression through the 1149.1 compliant state machine states. The TMS pin is sampled on the rising edge of TCK. The JTAG state machine remains in or transitions to a new TAP state depending on the current state of the TAP, and the present state of the TMS input. An internal pull-up resistor is present on TMS per the JTAG specification. The internal resistor is pulled to the VCCIO of Bank 0.

TCK: The test clock pin (TCK) provides the clock used to time the other JTAG port pins. Data is shifted into the instruction or data registers on the rising edge of TCK and shifted out on the falling edge of TCK. The TAP is a static design permitting TCK to be stopped in either the high or low state. The maximum input frequency for TCK is specified in the DC and Switching Characteristics section of the MachXO2 Family Data Sheet. The TCK pin does not have a pull-up. An external pull-down resistor of 4.7 K ohms is recommended to avoid inadvertently clocking the TAP controller as power is applied to the MachXO2.

JTAGENB: The JTAG ENABLE pin, also known as the IEEE 1149.1 conformance pin, is an input pin that can be used to multiplex the JTAG port. The JTAGENB pin is only active in user mode. The JTAGENB pin is a user I/O while the JTAG port is in the ENABLE state. Figure 14-7 shows the default behavior of the JTAG port of a MachXO2 device.

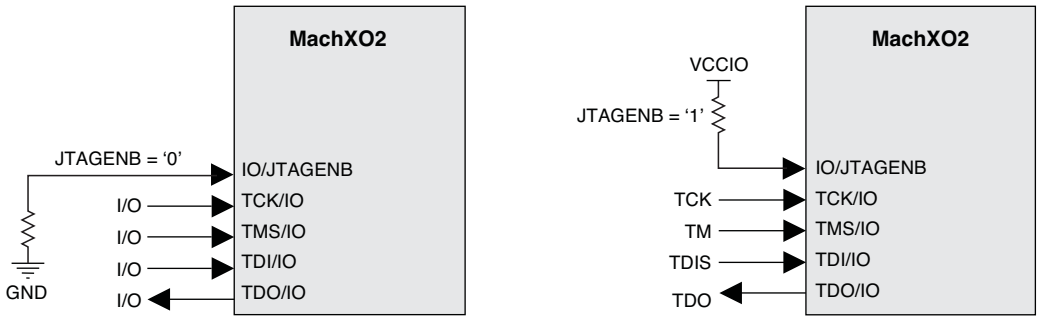
Figure 14-7. Default JTAG Port with JTAG_PORT = ENABLE



The JTAG port can become general purpose I/O. By setting the JTAG_PORT preference in the Diamond Spreadsheet View to the DISABLE state. When the JTAG port is in the DISABLE state the JTAGENB pin becomes a dedi-

cated input. Driving the JTAGENB low disables the JTAG port and the four JTAG pins become general purpose I/Os. Driving the JTAGENB input high enables the JTAG port. Figure shows JTAG port behavior under the control of the JTAGENB.

Figure 14-8. JTAG Port Behavior with JTAG_PORT = DISABLE



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

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

Configuration Modes

The MachXO2 provides multiple options for loading the configuration SRAM from a non-volatile memory. The previous section described the physical interface necessary to interact with the MachXO2 configuration logic. This section focuses on describing the functionality of each of the different configuration modes. Descriptions of important settings required in the Diamond Spreadsheet View are also discussed.

SDM Mode

Self Download Mode is the primary configuration method for the MachXO2. The advantages of Self Download Configuration Mode include:

- **Speed**: The MachXO2 is ready to run in a few milliseconds depending on the density of the device.
- **Security**: The configuration data is never seen outside the device during the load to SRAM. You can prevent the internal memory from being read.
- **Reduced cost**: There is no need to purchase a PROM specifically reserved for programming the MachXO2.
- **Reduced board space**: Elimination of an external PROM allows your board to be smaller.
- **Improved reliability**: The MachXO2 can boot from an external PROM if the internal Flash memory gets corrupted during a system update.

The MachXO2 retrieves the configuration data from the internal Flash memory when it is using Self Download Mode. SDM is triggered when power is applied, a REFRESH command is received, or by asserting the PROGRAMN pin. As shown in Figure 14-5, the internal Flash memory has three sectors. The first sector, in most cases, is large enough to store the MachXO2 device's configuration data. The size of the configuration data changes based on how well it can be compressed and how many pre-initialized EBR components are in the design. As the size of the configuration data increases, the Configuration Flash sector can overflow. The overflow can be handled by allowing the configuration data to overflow into the User Flash Memory sector. It is, in rare cases, possible for the configuration data to overflow the Configuration Flash (CFM), and the User Flash Memory (UFM). Self Download Mode cannot be used when the Configuration Flash and User Flash Memory overflow occurs. Master SPI Configuration Mode must be used in the event of the CF/UFM overflow.

The normal situation for the configuration data is to fit completely within the Configuration Flash sector. Designs that do not use very much pre-initialized EBR will almost always meet this condition. The UFM is available for use as an internal Flash memory array. It is recommended that the CONFIGURATION option be set to CFG. This setting prevents the configuration data from overflowing into the UFM, assuring that data provisioned for the UFM is not overwritten during a device update.

The User Flash Memory, which is the second Flash sector, provides three different use models:

- Configuration data overflow from Configuration Memory
- Initialization of EBR and user-defined storage
- User-defined storage

Diamond, by default, builds the pre-initialized EBR data into the configuration data image. This may cause the configuration data to overflow into the UFM sector. In order to change the default state you need to use the Diamond Spreadsheet tool to modify the sysCONFIG's CONFIGURATION entry. The default state for the CONFIGURATION entry is to be set to CFG.

The configuration data can be logically split to place the pre-initialization data for the EBR into the UFM. Setting the CONFIGURATION option to CFG_EBRUFM causes the Diamond software to place the configuration data into the Configuration Flash, and the EBR initialization data into the UFM. This locates the EBR initialization data into the first pages of the UFM. The current Diamond development tools do not provide a way to map the EBR initialization data stored in the UFM to the associated EBR in the FPGA fabric.

In addition to the automatic assignment of the initialized EBR data, you have the ability to add a data block for your own purposes. Using IPexpress, you can associate a memory initialization file to the UFM. This data is stored in the last memory locations of the UFM in order to prevent collisions with the EBR initialization data.

The user-defined storage mode of operation permits the sector to behave like a general purpose Flash memory. You can choose to use IPexpress to pre-initialize data, or you can use it as if it were a discrete Flash memory device with a single erasable sector.

In all three cases, the UFM can only be erased by erasing the whole sector. It is your responsibility to restore configuration data, EBR initialization data, and your implementation specific data. In other words, you need to read all data in the UFM, merge your changes, erase the UFM, and write the new data back into the UFM.

Master SPI Configuration Mode (MSPI)

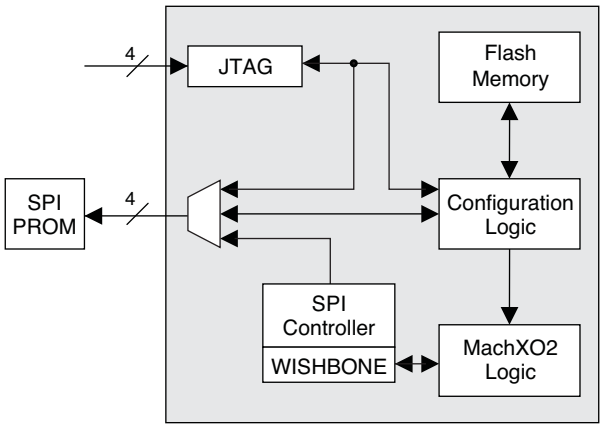
Master SPI Configuration Mode is the only other self-controlled configuration mode available to the MachXO2. When the MachXO2 has the Master SPI Configuration mode (MSPI) enabled it is able to automatically retrieve the configuration data from an externally attached SPI Flash. The MSPI configuration port is not available when the MachXO2 is in the Feature Row HW Default Mode state. When configuring using the MSPI mode be sure to enable the MSPI port in the Feature Row. Lattice recommends having a secondary configuration port available, one that is active when the MachXO2 is in Feature Row HW Default Mode state, that allows you to recover the MachXO2 in the event of a programming error.

Table 14-11. Master SPI Port Pins

Pin Name	I^2C Function
MCLK	Clock output from the MachXO2 Configuration Logic and Master SPI controller. Connect MCLK to the SCLK input of the Slave SPI device.
SISPI	Serial Data output from the MachXO2 to the slave SPI SI input.
SPISO	Serial Data input to the MachXO2 configuration logic from the slave SPI SO output.
CSSPIN	Chip select output from the MachXO2 configuration logic to the slave SPI Flash holding configuration data for the MachXO2.

Table 14-2 provides information about the amount of memory needed for MachXO2 configuration data by device density. Select a SPI Flash that accepts 03 hex Read Opcodes. The MachXO2 is only able to use the 03 hex Read Opcode.

Figure 14-9. Master SPI Configuration Mode



The MachXO2 begins retrieving configuration data from the SPI Flash when power is applied, a REFRESH command is received, or the PROGRAMN pin is asserted and released. The MCLK/CCLK I/O takes on the Master Clock (MCLK) function, and begins driving a nominal 2.08 MHz clock to the SPI Flash's SCLK input. CSSPIN is asserted low, commands are transmitted to the PROM over the SI/SISPI output, and data is read from the PROM on the SO/SPISO input pin. When all of the configuration data is retrieved from the PROM the CSSPIN pin is deasserted, and the MSPI output pins are tri-stated.

The MCLK frequency always starts downloading the configuration data at the nominal 2.08 MHz frequency. The MCCLK_FREQ parameter, accessed using Spreadsheet View, can be used to increase the configuration frequency. The configuration data in the PROM has some padding bits, and then the data altering the MCLK base frequency is read. The MachXO2 reads the remaining configuration data bytes using the new MCLK frequency.

After the MachXO2 enters user mode the Master SPI configuration port pins tri-state. This allows data transfers across the SPI. There are two primary methods available for transferring data across the SPI bus. The first method available to you is to enable the Embedded Function Block (EFB) in the MachXO2. Using IPexpress™ you instantiate the EFB, and you choose the features you want active. One of the features available in the EFB is a SPI Master Controller. The SPI Master Controller in the EFB attaches directly Master SPI configuration port pins. The controller provides a set of status, control, and data registers for initiating SPI bus transactions. The registers are accessed using the internal WISHBONE data bus. Logic residing in the programmable section of the the MachXO2 can be created to perform transactions across the WISHBONE bridge to the EFB, which in turn generate SPI bus transactions.

The second way to perform Master SPI configuration port transactions is to master them from the JTAG port. The MachXO2 includes a JTAG to MSPI passthru circuit that allows the slave SPI Flash to be erased, programmed, and read. The primary method for programming the attached SPI Flash is to use Diamond Programmer to transfer a configuration data file from your personal computer. This is useful during board development and debug. *Note: To support JTAG to MSPI passthru programming mode a 1Kohm pull-up resister is required on MCLK.*

Another way to program a SPI Flash using the JTAG port is to use the Lattice ispVME solution. ispVME is C code written for an embedded microprocessor. The microprocessor reads a data file crafted by the Diamond Deployment Tool, and runs the ispVME code. The firmware uses port I/O to drive the JTAG port of the MachXO2, which in turn passes the data to the Master SPI port. Refer to the ispVME tool suite for information about updating an attached SPI Flash using a microprocessor.

The advantage of using the JTAG port for programming the SPI Flash attached to the MachXO2 is that the MachXO2 is permitted to be in the Feature Row HW Default Mode state. JTAG is able to program a device in Flash

Mode Feature Row or User Mode Feature Row state. In order to do so, the Master SPI port pins must be enabled. The passthru is an integral part of the JTAG TAP system. Obviously, the JTAG port must be available in order for this method to succeed.

To set the MachXO2 for operation using the MSPI configuration mode you must:

- Store the entire configuration data in an external SPI Flash
- The data must start at offset 0x000000 within the PROM
- Set the preferences as shown in Table 14-12
- Enable JEDEC File creation in the Diamond Process Pane
- Run the Export Files process to build your design

Table 14-12. Master SPI Configuration Software Settings

Preference	Setting
MASTER_SPI_PORT	ENABLE
CONFIGURATION	EXTERNAL
GENERATE_BITSTREAM	ENABLE

The Export Files process generates both a JEDEC file and BIT file. It is important that both files be used. The JEDEC file must be programmed into the MachXO2 in order to write the Feature Row. The JEDEC file enables the MSPI configuration port.

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

- Diamond Programmer can transmit the SPI Flash data using a JTAG download cable
- A microprocessor running ispVME
- Automatic Test Equipment can program the SPI Flash using JTAG
- Pre-programmed SPI Flash memories can be pre-assembled onto your printed-circuit board

Once the MachXO2 Feature Row is programmed, and the SPI Flash contains your configuration data, you can test the configuration. Assert the PROGRAMN, transmit a REFRESH command, or cycle power to the board, and the MachXO2 will configure from the external SPI Flash.

Dual Boot Configuration Mode

Dual Boot Configuration Mode is a combination of Self Download Mode and Master SPI Configuration Mode. The MachXO2, when set up in Dual Boot Mode, tries to configure first from the internal Flash memory using SDM. If the SDM configuration fails, the MachXO2 attempts to configure itself using MSPI mode. The load order can't be reversed.

The internal data is the primary configuration data, and the external data is the golden configuration data. The primary image can fail in one of two ways:

- A bitstream CRC error is detected from on-chip Flash memory
- A time-out error is encountered when loading from on-chip Flash

A CRC error is caused by incorrect data being written into the internal Flash memory. Data is read from the Flash memory in rows. As each row enters the Configuration Engine the data is checked for CRC consistency. Before the data enters the Configuration SRAM the CRC must be correct. Any incorrect CRC causes the device to erase the Configuration SRAM and retrieve configuration data from the external PROM.

It is possible for the data to be correct from a CRC calculation perspective, but not be functionally correct. In this instance the internal DONE bit will never become active. The MachXO2 counts the number of master clock pulses it has provided after the Power On Reset signal was released. When the count expires without DONE becoming active the FPGA attempts to get it's configuration data from the external PROM.

Dual boot configuration mode typically requires two configuration data files. One of the two configuration data files is a fail-safe image that is rarely, if ever, updated. The second configuration data file is a working image that is routinely updated. The working image is stored in the Configuration Flash, the fail-safe image is stored in the external SPI memory. One Diamond project can be used to create both the working and the fail-safe configuration data files. Configure the Diamond project with an implementation named **working**, and an implementation named **failsafe**. Read the Diamond Online Help for more information about using Diamond implementations.

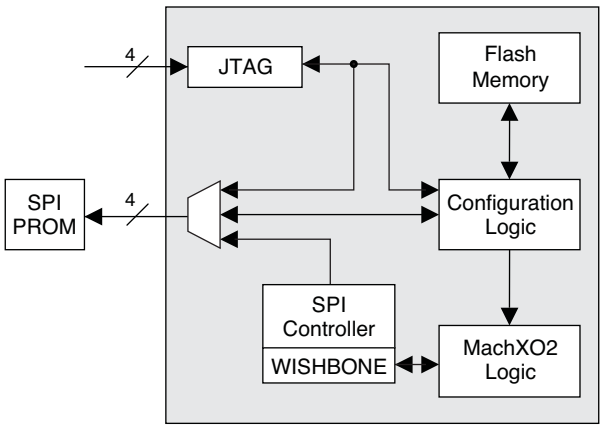
Use the following preferences to build a dual-boot design:

Preference	Dual-Boot Setting
CONFIGURATION	CFG | CFG_EBRUFM | CFGUFM
MASTER_SPI_PORT	ENABLE
GENERATE_BITSTREAM	ENABLE
COMPRESS_CONFIG	ON | OFF

Diamond creates a JEDEC file for the primary configuration data that is stored in the internal Flash memory. A BIT file is created for the golden configuration data that is stored in the external SPI Flash. The golden configuration data must be located in the external SPI Flash starting at address 0x010000. This differs from a single image Master SPI Configuration Mode, which requires the configuration data be stored at offset 0x000000.

To prevent the MachXO2 from using dual boot mode when using the User Master SPI controller set the MASTER_SPI_PORT preference to EFB_USER. This reserves the Master SPI configuration port pins and prevents dual-boot.

Figure 14-10. MSPI Mode



Slave SPI Mode (SSPI)

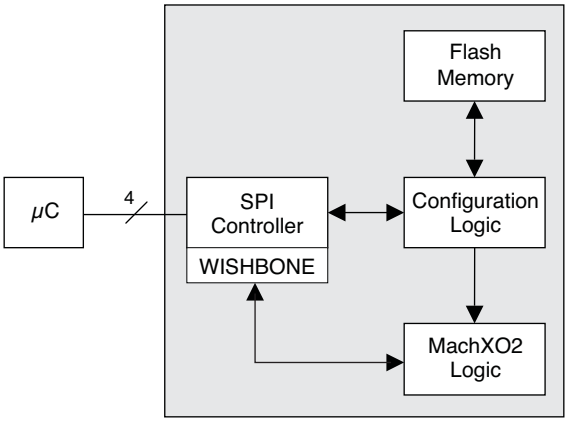
The MachXO2 provides a Slave SPI configuration port that allows you to access features provided by the Configuration Logic. You can reprogram the Configuration Flash, UFM and Feature Row, and access status/control registers within the Configuration Logic block. The Flash memories is done using either offline or transparent operations. The SRAM is not directly accessible using the Slave SPI port. It is necessary to send a REFRESH command to load a new Flash image into the SRAM.

Table 14-13. Slave SPI Port Pins

Pin Name	Description
CCLK	Configuration clock input that is driven by a SPI master controller.
SI	Serial Data Input to the MachXO2 Configuration Logic for command and data.
SO	Serial Data Output from the MachXO2 configuration logic.
SN	Chip select to enable the MachXO2 configuration logic.

In the Slave SPI mode, the MCLK/CCLK pin becomes CCLK (i.e. Configuration clock). Input data is read into the MachXO2 device on the SI pin at the rising edge of CCLK. Output data is valid on the SO pin at the falling edge of CCLK. The SN acts as the chip select signal. When SN is high, the SSPI interface is deselected and the SO/SPISO pin is tri-stated. Commands can be written into and data read from the MachXO2 when SN is asserted. The MachXO2 SSPI port only accepts Mode 0 bus transactions to the Configuration Logic.

Figure 14-11. Slave SPI Configuration Mode



The SSPI port is active when the MachXO2 is in Feature Row HW Default Mode state. Diamond's default preference for the SLAVE_SPI_PORT is to DISABLE the port. Use the Spreadsheet View to ENABLE the SLAVE_SPI_PORT preference in your design to keep the SSPI port active in user mode. Lattice recommends you keep a secondary programming port active in the event the SSPI port is accidentally disabled.

The SSPI port is used to erase, program, and verify the Configuration Flash, User Flash Memory, and the Feature Row. It is not capable of directly accessing the configuration SRAM. To prevent unintentional erasure of the Feature Row, it is recommended the SSPI port be used to perform transparent updates of the Flash memory. The SSPI port can issue a REFRESH command to make a newly programmed image active. The REFRESH command can be safely used when the MachXO2 is using External or Dual Boot configuration mode because the REFRESH operation will not begin until SN is deasserted.

Programming the MachXO2 using the SSPI port is complex. Lattice provides 'C' source code called SSPIEmbedded to insulate you from the complexity of programming the MachXO2. It is recommended that SSPIEmbedded be used when you want to reprogram the MachXO2 Flash memory.

In addition to reprogramming the Flash memory the SSPI port can be used to access several status and control registers in the MachXO2. A list of the available commands and information about the registers is described in TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices. Accessing the status registers is less complex and does not require the use of the SSPIEmbedded code.

I²C Configuration Mode

The MachXO2 has an I²C Configuration port for use in accessing the configuration logic. An I²C master can communicate to the configuration logic using 10-bit or 7-bit addressing modes. The I²C SCL input can accept a clock frequency up to 400KHz. You can reprogram the Configuration Flash, UFM and Feature Row, and access sta-

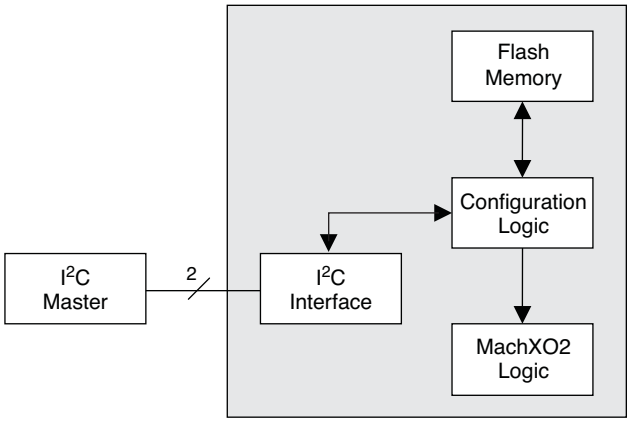
tus/control registers within the configuration logic block. Reprogramming the Flash memories can be done either offline or in transparent operations. You cannot directly update the configuration SRAM. It is necessary to send a REFRESH command after reprogramming the Flash memory in order for the configuration SRAM to be updated.

Table 14-14. I²C Port Pins

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

The I²C Configuration port is available when the MachXO2 is in Feature Row HW Default Mode state. The default state set for the I2C_PORT in the Diamond design software is to place the I2C_PORT in the DISABLE state. You must make sure the I2C_PORT is set to the ENABLE state to leave the I²C interface active in user mode. Lattice recommends making a second configuration port available (e.g. JTAG) in order to recover from erroneously disabling the I²C port.

Figure 14-12. I²C Configuration Logic



There are two hardened I²C controllers in a MachXO2 device, a primary and a secondary. The primary controller provides an interface to the MachXO2 Configuration Logic, and access to Wishbone registers. Access to the Wishbone registers is referred to as User Mode I²C. The primary I²C controller is the only one that permits access to the Configuration Logic. The Secondary I²C controller is always a User Mode I²C controller.

When the MachXO2 is in Feature Row HW Default Mode state the I²C port is enabled, and you may interact with the primary I²C controller. Whenever the I²C port is enabled access to the Configuration Logic is possible. It is not necessary to instantiate the EFB to gain access to the Configuration Logic.

The Primary I²C controller provides access to the Configuration Logic when:

- The MachXO2 is in Feature Row HW Default Mode state
- The EFB is not instantiated, and the I²C port pins are in the ENABLE state
- The EFB is instantiated, and the I²C port pins are in the ENABLE state

An external I²C master accesses the Configuration Logic using address 1000000 (7-bit mode) or 1111000000 (10-bit mode) unless the EFB I²C base address has been modified. Use IPexpress, not Spreadsheet View, to modify the address to which the Primary and Secondary I²C controllers respond. It is necessary to instantiate the EFB in order to change the address. The address is shared by the Primary and Secondary I²C controllers.

Table Table 14-15 shows the address decoding used to access the I²C resources in the MachXO2.

Table 14-15. Slave Addresses for I²C Ports

Slave Address	I²C Function
yyyxxxxx**00**	Primary I²C Controller Configuration Logic address. Always responds to 7-bit or 10-bit addresses.
yyyxxxxx**01**	User Mode Primary I²C Controller address.
yyyxxxxx**10**	User Mode Secondary I²C Controller address.
yyyxxxxx**11**	Primary I²C Configuration Logic Reset. Always responds to 7-bit or 10-bit addresses.

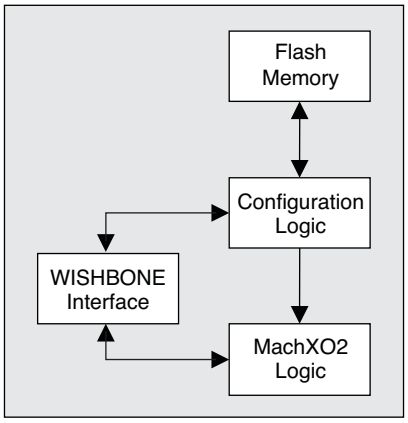
The Primary I²C core can be used for accessing the User Flash Memory (UFM) and for programming the Configuration Flash. However, the Primary I²C port cannot be used for both UFM/Configuration access and User functions in the same design. The operation of the User Mode Primary and User Mode Secondary I²C controllers is described in TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices. Interacting with these I²C slave devices is not covered in this document.

The fourth I²C resource in the MachXO2 is located at offset 3. In some instances an I²C memory transaction to the configuration logic may be interrupted or abandoned. It is possible for a command to be accepted by the configuration logic that causes the configuration logic to respond with data. In the event that the I2C memory transaction is interrupted or abandoned, the configuration logic continues to return the queued data. New incoming I²C commands may be considered padding bytes or may be misinterpreted. Clear this condition by writing any value to offset 3. The configuration logic command interpreter will reset, any queued data will be flushed, and subsequent I²C memory transactions to the Configuration Logic will operate correctly.

WISHBONE Configuration Mode

The MachXO2 can access the Configuration Flash, User Flash Memory, and the Feature Row from an internal WISHBONE bus. To use the WISHBONE bus the Embedded Function Block must be inserted into your design. You design logic to interface to the EFB and then perform WISHBONE bus transactions to access resources attached to the configuration logic.

Figure 14-13. WISHBONE Configuration Mode



The MachXO2 must be in user mode in order to access the WISHBONE interface. Accessing and updating the resources made available by the configuration logic must be done in Transparent mode. Attempting accesses to the configuration logic in offline mode causes a deadlock because the MachXO2 leaves user mode.

Appendix C describes how you can perform erase, program, and verify functions using the WISHBONE interface. You can get more detailed information about the MachXO2 WISHBONE interface by reading TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

JTAG Mode

The JTAG port is the most flexible configuration and programming port available on the MachXO2. The JTAG provides:

- Offline Flash memory programming
- Transparent Flash memory programming
- Offline SRAM configuration
- Full access to the MachXO2 Configuration Logic
- Device chaining
- IEEE 1149.1 testability
- IEEE 1532 Compliant programming

The JTAG port is available when the MachXO2 is in Feature Row HW Default Mode state. The port is enabled by default by Diamond 1.4. The MachXO2 JTAG port pins are not dedicated to performing the IEEE 1149.1 TAP function. The JTAG port may be recovered for use as general purpose I/O. See "sysCONFIG Pins" on page 14-9 for details on recovering the JTAG port pins for use as general purpose I/O.

The MachXO2 JTAG port is a valuable asset due to its flexibility. It provides the best capabilities for system and device debug. Lattice recommends the JTAG port remain accessible in every MachXO2 design. Advantages for keeping the JTAG port active include:

- **Multi-chain Architectures**: The JTAG port is the only configuration and programming port that permits the MachXO2 to be combined in a chain of other programmable logic.
- **Reveal Debug**: The Lattice Reveal debug tool is an embed-able logic analyzer tool. It allows you to analyze the logic inside the MachXO2 in the same fashion as an external logic analyzer permits analysis of board level logic. Reveal access is only available via the MachXO2 JTAG port.
- **SRAM Readback**: The JTAG port is the only sysCONFIG port able to directly access the MachXO2's configuration SRAM. It is occasionally necessary to perform failure analysis for SRAM based FPGAs. A key component to failure analysis can involve reading the configuration SRAM. This kind of failure analysis is lost when the JTAG port is not enabled.
- **Boundary Scan Testability**: Board level connectivity testing performed using IEEE 1149.1 JTAG is a key capability for assuring the quality of assembled printed-circuit-boards. Preserving the MachXO2 JTAG port is vital for boundary scan testability. Lattice provides Boundary Scan Description Language files for the MachXO2 on the Lattice website.

TransFR Operation

The MachXO2, like other Lattice FPGAs, provides for the TransFR™ capability. TransFR is described in TN1087 Minimizing System Interruption During Configuration Using TransFR Technology. The MachXO2 operates differently than earlier generations of Lattice FPGAs. Earlier generations depend solely on the JTAG port for TransFR operation. The JTAG port uses the BSCAN cells to either capture the current state on the I/O, or to force the I/O to a known state.

The MachXO2 allows TransFR programming to occur on two other ports. The I^2C, and SSPI configuration ports can perform programming operations, but do not have access to the JTAG BSCAN cells. In order to make TransFR available for the I^2C and SSPI programming ports a new TransFR process was created.

The MachXO2 enables the TransFR feature using configuration SRAM bits. In order for TransFR to operate correctly the configuration active in the SRAM must have the ENABLE_TRANSFR set to the ENABLE state. The new configuration data being programmed into the MachXO2 must also have the ENABLE_TRANSFR set to the ENABLE state. When both of these conditions are met the MachXO2 preserves the I/O personality and voltage lev-

els when a Refresh command is issued. The MachXO2 latches and holds the state of the output buffer at the INITN falling edge and maintains the state on the outputs until the GOE strobe is asserted during the Wake Up process. Input data is used as soon as the internal DONE strobe is asserted during the Wake Up process.

In the event the MachXO2 fails to configure, the outputs remain held pending the GOE assertion in the Wake Up process. The outputs, as a result, remain in a known state should the MachXO2 perform a Dual Boot fail safe load. This behavior is superior to the Boundary Scan method used in prior device families.

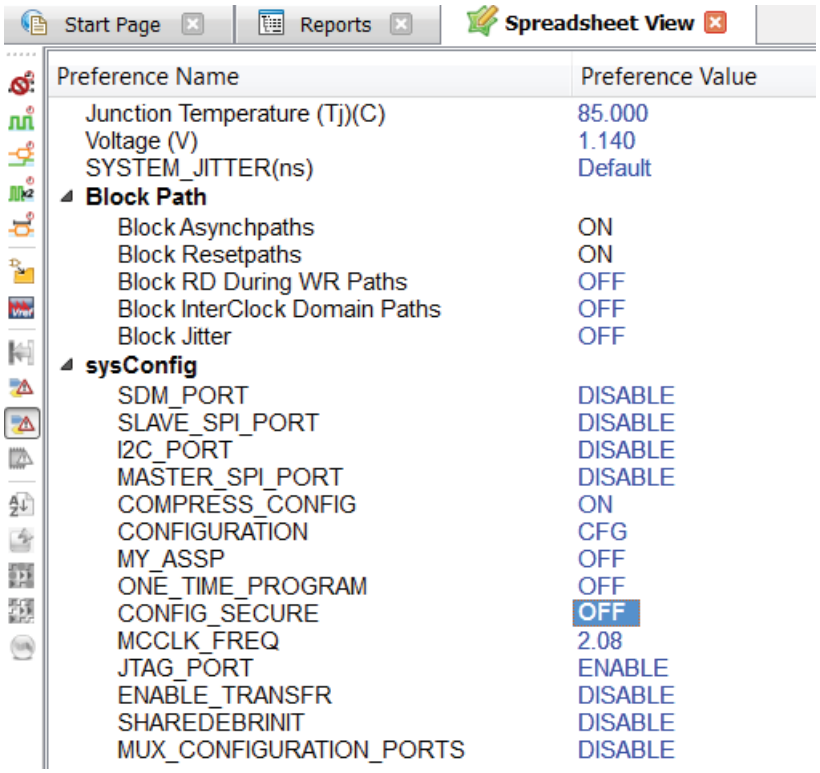
Software Selectable Options

The operation of the MachXO2 configuration logic is managed by options selected in the Diamond design software. Other FPGAs provide dedicated I/O pins to select the configuration mode. The MachXO2 uses the non-volatile Feature Row to select how it will configure. The Feature Row's default state needs to be modified in almost every design. You use the Diamond Spreadsheet View to make the changes to the operation of the MachXO2 Feature Row which alters the operation of the configuration logic.

The configuration logic preferences are accessed using Spreadsheet View. Click on the Global Preferences tab, and look for the sysCONFIG tree. The sysCONFIG section is shown in Figure 14-14. The sysCONFIG preferences are divided into three categories:

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

Figure 14-14. sysCONFIG Preferences in Global Preferences Tab, Diamond Spreadsheet View



Configuration Mode and Port Options

The configuration and port options allow you to decide which configuration ports continue to operate after the MachXO2 device is in user mode. You can also control the availability of status pins, as well as the speed at which

configuration data is read from an external PROM. The selections made here are saved in the Feature Row and remain in effect until the Feature Row is erased. The only exception is the MCCLK_FREQ parameter, which is stored in the configuration data.

The configuration and port options can be used in any combination.

Table 14-16. Configuration Mode/Port Options

Option Name	Default Setting	All Settings
JTAG_PORT	ENABLE	DISABLE, ENABLE
SLAVE_SPI_PORT	DISABLE	DISABLE, ENABLE
MASTER_SPI_PORT	DISABLE	DISABLE, ENABLE, EFB_USR
I2C_PORT	DISABLE	DISABLE, ENABLE
SDM_PORT[1, 2]	DISABLE	DISABLE, PROGRAMN, PROGRAMN_DONE, PROGRAMN_DONE_INITN
MCCLK_FREQ	2.08	See description below
ENABLE_TRANSFR	DISABLE	DISABLE, ENABLE

1. The default for SDM_PORT was PROGRAMN in ispLEVER 8.1 SP1 and Diamond 1.1.
2. The 32 QFN package does not have an INITN pin. Because of this, the option SDM_PORT = PROGRAMN_DONE_INITN is not available.

JTAG Port

The JTAG_PORT preference allows you to decide how the JTAG configuration port pins operate when the MachXO2 device is in user mode. There are two states the JTAG_PORT can be set to:

- **ENABLE** – In this mode the JTAG I/O are dedicated and provide an IEEE 1149.1 JTAG interface
- **DISABLE** – In this mode the JTAG I/O pins are controlled dynamically using the JTAGENB pin

The JTAGENB pin is only available when the JTAG_PORT is in the DISABLE state. JTAGENB, when asserted high, makes the four JTAG I/O act as an IEEE 1149.1 JTAG port. JTAGENB driven low causes the four I/O to be available for use as general purpose I/O.

Lattice recommends designing so that the JTAG port can be accessed in the event reprogramming the MachXO2 disables your primary configuration port.

Slave SPI Port

The SLAVE_SPI_PORT allows you to preserve the Slave SPI configuration port after the MachXO2 device enters user mode. There are two states to which the SLAVE_SPI_PORT preference can be set:

- **ENABLE** – This setting preserves the SPI port I/O when the MachXO2 device is in user mode. When the pins are preserved, an external SPI master controller can interact with the configuration logic. The preference also prevents you from over-assigning I/O to the port pins.
- **DISABLE** – This setting disconnects the SPI port pins from the configuration logic. By itself it does not make the port pins general purpose I/O. Both the SLAVE_SPI_PORT and MASTER_SPI_PORT must be in the DISABLE state for the SPI port pins to be general purpose I/O.

The SLAVE_SPI_PORT can be enabled at the same time as the MASTER_SPI_PORT. It is necessary to guarantee that the internal SPI master controller not perform SPI transactions at the same time as an external SPI master. It is your responsibility to prevent two SPI masters from operating simultaneously.

Master SPI Port

The MASTER_SPI_PORT allows you to preserve the SPI configuration port after the MachXO2 device enters user mode. There are three states to which the MASTER_SPI_PORT preference can be set:

- **ENABLE** – This setting preserves the SPI port I/O when the MachXO2 is in user mode. This preference makes External or Dual Boot configuration modes active. Using this preference in combination with CONFIGURATION =

EXTERNAL enables external boot mode. This preference in combination with CFG, CFG_EBRUFM, CFGUFM enables Dual Boot mode. After entering user mode, the SPI controller in the EFB has access to the SPI port for performing SPI bus transactions. The preference also prevents you from over-assigning I/O to the port pins.

- **EFB_USER** – This setting preserves the SPI port I/O when the MachXO2 is in user mode. After entering user mode, the SPI controller in the EFB has access to the SPI port for performing SPI bus transactions. The preference also prevents you from over-assigning I/O to the port pins.
- **DISABLE** – This setting disconnects the SPI port pins from the configuration logic. By itself it does not make the port pins general purpose I/O. Both the SLAVE_SPI_PORT and MASTER_SPI_PORT must be in the DISABLE state for the SPI port pins to be general purpose I/O.

The MASTER_SPI_PORT can be enabled at the same time as the SLAVE_SPI_PORT. It is necessary to guarantee that the internal SPI Master controller not perform SPI transactions at the same time as an external SPI Master. It is your responsibility to prevent two SPI masters from operating simultaneously.

I^2C Port

The I2C_PORT allows you to preserve the I^2C configuration port after the MachXO2 device enters user mode. There are two states to which the I2C_PORT preference can be set:

- **ENABLE** – This setting preserves the I^2C port I/O when the MachXO2 is in user mode. When the pins are preserved, an external I^2C Master controller can interact with the configuration logic. The preference also prevents you from over-assigning I/O to the port pins.
- **DISABLE** – This setting disconnects the I^2C port pins from the configuration logic. The port pins become general purpose I/O.

In order to use the primary and secondary I^2C controllers in the EFB, the I2C_PORT must be in the ENABLE state.

SDM Port

The SDM_PORT allows you to select the programming status pins after the MachXO2 device enters user mode. There are four states to which the SDM_PORT preference can be set:

- **DISABLE** – This setting causes the PROGRAMN, DONE, and INITN status pins to become general purpose I/O.
- **PROGRAM** – This setting preserves the PROGRAMN pin when the MachXO2 device is in user mode. Asserting this pin active low causes the MachXO2 device to reconfigure. The DONE and INITN pins are general purpose I/O.
- **PROGRAM_DONE** – This setting preserves the PROGRAMN and DONE pins when the MachXO2 device enters user mode. INITN is a general purpose I/O.
- **PROGRAM_DONE_INITN** – This setting preservers PROGRAM, DONE, and INITN in user mode.

Lattice recommends setting the SDM_PORT to PROGRAMN when using Master SPI or Dual Boot configuration modes. The PROGRAMN pin is the only way to perform a “warm” reconfiguration of the MachXO2 device, unless another configuration port is available to transmit a REFRESH command.

MCCLK Frequency

The MCLK_FREQ preference allows you to alter the MCLK frequency used to retrieve data from an external SPI Flash when using EXTERNAL or Dual Boot configuration modes. The MachXO2 uses a nominal 2.08MHz (+/- 5.5%) clock frequency to begin retrieving data from the external SPI Flash. The MCLK_FREQ value is stored in the incoming configuration data. It is not stored in the Feature Row. The MachXO2 device reads a series of padding bits, a “start of data” word (0xBDB3) and a control register value. The control register contains the new MCLK_FREQ value. The MachXO2 switches to the new clock frequency shortly after receiving the MCLK_FREQ value. The MCLK_FREQ has a range of possible frequencies available from 2.08 MHz up to 133 MHz (see Table 14-9). Take care not to exceed the maximum clock rate of your SPI Flash, or of your printed circuit board.

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

ENABLE_TRANSFR

The TransFR function used by the MachXO2 requires the configuration data loaded into the configuration SRAM, and any future configuration data file loaded into the internal Flash memory have the ENABLE_TRANSFR set to the ENABLE state. See "TransFR Operation" on page 14-25, and TN1087 Minimizing System Interruption During Configuration Using TransFR Technology for more information about using TransFR with the MachXO2.

Bitstream Generation Options

The Bitstream Generation options allow you to decide how the Diamond development tools create the configuration data for the MachXO2 device. The CONFIGURATION, USERCODE, CUSTOM_IDCODE, and SHAREDEBRINIT settings are saved in the Feature Row and remain in effect until the Feature Row is erased. The other options allow you to control the JEDEC and BIT files that are generated by Diamond. Table 14-17 gives a summary of these options.

Table 14-17. Bitstream Generation Options

Option Name	Default Setting	All Settings
GENERATE_BITSTREAM	DISABLE	DISABLE, ENABLE
COMPRESS_CONFIG	ON	OFF, ON
CONFIGURATION[1]	CFG	CFG, CFG_EBRUFM, CFGUFM, EXTERNAL
USERCODE_FORMAT	BINARY	HEX, BINARY, ASCII
USERCODE	<all zero>	32-bit arbitrary
CUSTOM_IDCODE_FORMAT	BINARY	HEX, BINARY
CUSTOM_IDCODE	<all zero>	32-bit arbitrary
SHAREDEBRINIT	DISABLE	DISABLE, ENABLE
MUX_CONFIGURATION_PORTS	DISABLE	DISABLE, ENABLE

1. The default was CFG_EBRUFM in ispLEVER 8.1 SP1 and Diamond 1.1.

GENERATE BITSTREAM

The GENERATE_BITSTREAM preference informs Diamond to create a BIT file in addition to a JEDEC file. The default setting for GENERATE_BITSTREAM, DISABLE, prevents Diamond from creating a BIT file. The GENERATE_BITSTREAM, in the ENABLE state, causes Diamond to build both a JEDEC file and a BIT file. The BIT file is a binary data file that can be programmed directly into an external SPI Flash.

Bitstream generation is affected by the COMPRESS_CONFIG preference. Read the COMPRESS_CONFIG section for additional information.

COMPRESS_CONFIG

The COMPRESS_CONFIG preference alters the way JEDEC and BIT files are generated. The COMPRESS_CONFIG default setting is to be ON. There are three legal combinations for the COMPRESS_CONFIG and GENERATE_BITSTREAM preferences, shown in Table 14-18.

Table 14-18. Valid GENERATE_BITSTREAM/COMPRESS_CONFIG Combinations

COMPRESS_CONFIG	GENERATE_BITSTREAM	JEDEC Generated	
ON	OFF	Valid[1]	Yes
ON	ON	Valid	Yes
OFF	OFF	Invalid	N/A
OFF	ON	Valid	No

1. Default setting for COMPRESS_CONFIG and GENERATE_BITSTREAM.

JEDEC files, when they are built, are always compressed. Lattice recommends using COMPRESS_CONFIG=ON and GENERATE_BITSTREAM=ON when using External or Dual Boot configuration modes. The configuration time will be slightly reduced when reading configuration data from the external PROM and the Diamond tool will create a JEDEC file you can program into the internal Flash memory.

CONFIGURATION

The CONFIGURATION preference allows you to control the Configuration Flash and UFM sectors. The CONFIGURATION preference has four possible settings:

- **CFG** – The CFG preference is the default mode for building configuration data. The configuration bitstream is stored in the Configuration Flash and is not permitted to overflow into the UFM sector. The configuration data includes EBR initialization data. The UFM sector is available for your use as general purpose Flash memory in user mode.
- **CFG_EBRUFM** – This preference creates configuration data that is stored in the Configuration Flash. EBR initialization data is stored in the lowest page addresses of the UFM sector. The UFM sector is available in user mode. You must restore the EBR initialization data when making changes to the UFM to guarantee correct operation.
- **CFGUFM** – This preference creates configuration data that is stored in the Configuration Flash. This mode differs from CFG by allowing the configuration data to overflow into the UFM. The configuration data increases in size as EBR initialization data is added to the design.
- **EXTERNAL** – This preference generates configuration data that is stored in an external memory. The UFM sector is available as general purpose Flash memory in user mode.

The CONFIGURATION preference defaults to the CFG state in the current release of the Diamond software. The Diamond design software only generates JEDEC files when your entire design fits within the Configuration Flash memory. The UFM is guaranteed to be available when the MachXO2 device enters user mode.

In the event the configuration data does not fit in the Configuration Flash memory try using the CFG_EBRUFM preference. This preference works well if your design has a significant amount of initialized EBR, and you still want access to UFM pages to store data. Depending on the amount of initialized EBR the UFM may still have sufficient space available for storing your data.

Use the CFGUFM option when the Configuration Flash is not large enough to store the configuration data, and you do not need to use UFM for storing your own data. It is possible, in rare instances, for the size of the configuration data to exceed the combined space of the Configuration Flash and UFM.

Use the EXTERNAL preference to build configuration data for use with Master SPI Configuration Mode. When the configuration data exceeds the combined space available in the Configuration Flash and UFM it is necessary to switch to EXTERNAL mode. EXTERNAL mode does not use any Configuration Flash or UFM resources. The UFM is available for your use in user mode. The GENERATE_BITSTREAM option must be in the ENABLE state when EXTERNAL is selected.

The MachXO2-256 device does not contain any UFM. The only configuration options available to this device are the CFG and EXTERNAL modes.

USERCODE

The MachXO2 Configuration Flash sector contains a 32-bit register for storing a user-defined value. The default value stored in the register is 0x00000000. Using the USERCODE preference you can assign any value to the register you desire. Suggested uses include the configuration data version number, a manufacturing ID code, date of assembly, or the JEDEC file checksum.

The format of the USERCODE field is controlled using the USERCODE_FORMAT preference. Data entry can be performed in either Binary, Hex, or ASCII formats.

USERCODE_FORMAT

The USERCODE_FORMAT preference selects the format for the data field used to assign a value in the USERCODE preference. The USERCODE_FORMAT has three options:

- **Binary** – USERCODE is set using 32 ‘1’ or ‘0’ characters.
- **Hex** – USERCODE is set using eight hexadecimal digits (i.e., 0-9A-F)
- **ASCII** – USERCODE is set using up to four ASCII characters

CUSTOM_IDCODE
The CUSTOM_IDCODE preference is used to assign a 32-bit register that resides in the Feature Row. The CUSTOM_IDCODE field is only active when the MY_ASSP preference is in the ON state. The value assigned can be entered in binary or hexadecimal, according to the CUSTOM_IDCODE_FORMAT preference. See "MY_ASSP" on page 14-32 for more information about how to assign a value to the CUSTOM_IDCODE preference.

CUSTOM_IDCODE_FORMAT
The CUSTOM_IDCODE_FORMAT preference selects the format for the data field used to assign a value in the CUSTOM_IDCODE preference. The CUSTOM_IDCODE_FORMAT has two options:

- **Binary** – CUSTOM_IDCODE is set using 32 ‘1’ or ‘0’ characters.
- **Hex** – CUSTOM_IDCODE is set using eight hexadecimal digits (i.e., 0-9A-F)

SHAREDEBRINIT
When set to ENABLE, this preference allows one copy of a unique memory initialization file to be stored in the Flash memory. This copy of the initialization values can be shared among multiple EBRs. Doing so reduces the bitstream size of the design and saves UFM space for other applications.

MUX_CONFIGURATION_PORTS
The MUX_CONFIGURATION_PORTS is used in the event that all configuration ports are disabled. Disabling all of the available configuration ports turns the MachXO2 into a “write one time” device. MUX_CONFIGURATION_PORTS confirms the removal of all configuration ports. The control is only active when all of the other configuration ports are set to the DISABLE state. MUX_CONFIGURATION_PORTS set to the ENABLE state enables the JTAGENB input pin, permitting the JTAG port pins to be multiplexed. Setting MUX_CONFIGURATION_PORTS to the DISABLE state causes the Diamond build tools to honor the removal of all other configuration ports, allowing the MachXO2 to become a “write one time” device.

Security Options

The Security Options allow you to select from a range of options for tracking or securing the MachXO2 device. Table 14-19 provides a summary of these options.

Table 14-19. Security Options

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

TRACEID
The MachXO2 introduces a new feature called TraceID. TraceID stamps each MachXO2 with a unique 64-bit ID. No two MachXO2 devices will have the same TraceID value even when they are loaded with the same configuration data. This differs from a USERCODE which is present in the configuration data. Every device that receives the configuration data using a USERCODE receives the same USERCODE value.

The TraceID is 64 bits long with the least significant 56 bits being immutable data. The 56 bits are a combination of the wafer lot, the wafer number and the X/Y coordinates locating the die on the wafer. The most significant eight

bits are provided by you and are stored in the Feature Row. The TraceID is changed using the Diamond Spreadsheet View. You enter a unique 8-bit binary value in the TraceID field and generate configuration data.

You can read more about the TraceID feature in TN1207, Using TraceID in MachXO2 Devices.

MY_ASSP
Every Lattice device has its own identification code identifying the device family, device density, and other parameters (e.g. voltage, device stepping, etc.). The code is accessible from any MachXO2 configuration port. The value stored in the IDCODE register allows you to uniquely identify a Lattice device.

The MY_ASSP preference permits you to change the value returned when the IDCODE is read from the FPGA. Set the MY_ASSP preference to the ON state. Turning the MY_ASSP ON enables the CUSTOM_IDCODE preference.

CUSTOM_IDCODE
The CUSTOM_IDCODE is the value you assign to override the default IDCODE in the MachXO2 device. You are only allowed to enter a 32-bit hexadecimal or binary value when the MY_ASSP preference is ON.

Overriding the IDCODE prevents the Lattice programming software from being able to identify the MachXO2 device, and as a result, prevents Programmer from being able to directly program the MachXO2 device. It is necessary to migrate to generating Serial Vector Format (SVF) files in order to program MY_ASSP enabled MachXO2 devices.

CONFIG_SECURE
When this preference set to ON, the read-back of the SRAM memory and the Configuration Flash memory are blocked. The read-back of the UFM will also be blocked if the bitstream overflows into the UFM block. The MachXO2 device cannot be read back, nor can it be programmed without erasing. The device must be erased in order to reset the security setting. The CONFIG_SECURE fuse and the Configuration Flash are erased in tandem. Once the security fuses are reset, the device can be programmed again.

ONE_TIME_PROGRAM
The MachXO2 has One Time Programmable (OTP) fuses that can be used to prevent the on-chip memory from being erased or programmed. The MachXO2 device has three OTP security fuses, one for each of the following memory sectors: SRAM, Configuration Flash, and UFM. This preference provides options to set the OTP security for each memory sector.

- **FLASH** – The Configuration Flash cannot be erased or programmed
- **FLASH_UFM** – The Configuration Flash and UFM cannot be erased or programmed
- **FLASH_UFM_SRAM** – The Configuration Flash, UFM, and SRAM cannot be erased or programmed

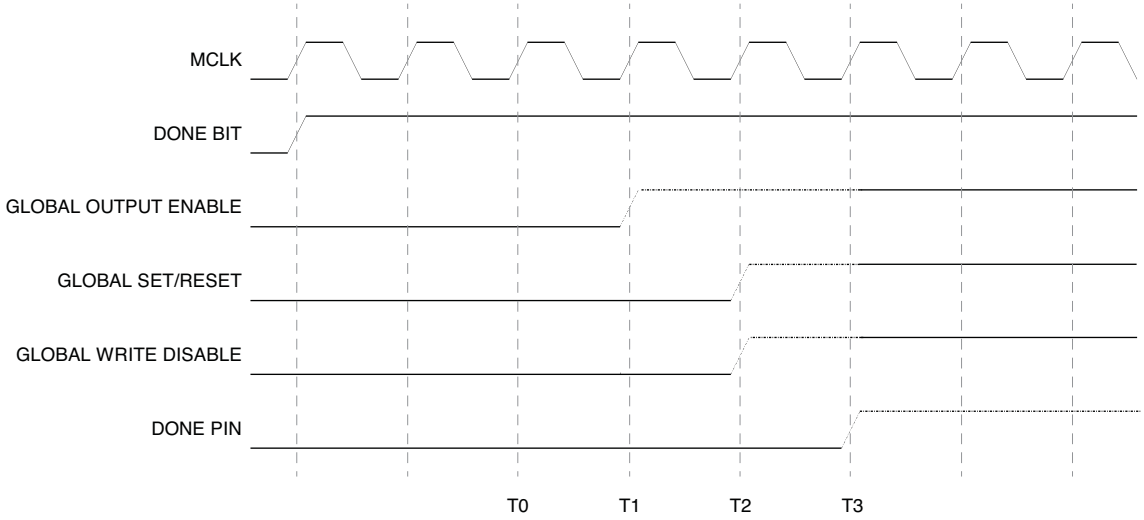
Once the ONE_TIME_PROGRAM preference is set for the Flash memory, the on-chip Flash memory cannot be erased or programmed. The configuration data is prevented from further modification, but the SDM mode can still be used to configure the device.

When the ONE_TIME_PROGRAM preference is set for the FLASH_UFM_SRAM memory, the device acts like an ASIC. You are no longer able to reprogram the internal Flash or UFM, and the SRAM cannot be changed from the JTAG port. Configuration of SRAM from on-chip Flash memory or external SPI Flash is still enabled.

Device Wake-up Sequence

When configuration is complete (the SRAM has been loaded), the device will wake up in a predictable fashion. If the MachXO2 device is the only device in the chain, or the last device in a chain, the wake-up process should be initiated by the completion of configuration. Once configuration is complete, the internal DONE bit will be set and then the wake-up process will begin. Figure 14-15 shows the wake-up sequence using the internal clock.

Figure 14-15. Wake-up Sequence Using Internal Clock



Wake-up Signals

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

- GSR is used to set and reset the core of the device. GSR is asserted (low) during configuration and de-asserted (high) in the wake-up sequence.
- When the GWDIS signal is low it safeguards the integrity of the RAM Blocks and LUTs in the device. This signal is low before the device wakes up. This control signal does not control the primary input pin to the device but controls specific control ports of EBR and LUTs.
- When low, GOE prevents the device's I/O buffers from driving the pins. The GOE only controls **output** pins. Once the internal DONE is asserted the MachXO2 will respond to input data.
- When high, the DONE pin indicates that configuration is complete and that no errors were detected.

Wake-up Clock Selection

The clock source used to complete the four state transitions in the wake-up sequence is user-selectable. Once the MachXO2 is configured, it enters the wake-up state, which is the transition between the configuration mode and user mode. This sequence is synchronized to a clock source, which defaults to MCLK/CCLK when sysCONFIG is used, or TCK when JTAG is used.

You can change the clock used by instantiating the START macro in your Verilog or VHDL. The clock must be supplied on an external input pin, because the MachXO2 does not begin internal operations until the Wake-up sequence is complete. There is no external indication the device is ready to perform the last four state transitions. You must either provide a free running clock frequency, or you must wait until the device is guaranteed to be ready to wake up. Using the START macro provides another mechanism for holding off configuring one or more programmable devices and then starting them synchronously.

Verilog

```
module START (STARTCLK);
input  STARTCLK;
endmodule

START u1 (.STARTCLK(<clock_name>)) /* synthesis syn_noprune=1 */;
```

VHDL

```
COMPONENT START
    PORT(
          STARTCLK    :  IN STD_ULOGIC
        );
END COMPONENT;
attribute syn_noprune: boolean ;
attribute syn_noprune of START: component is true;

begin
u1: START port map (STARTCLK =><clock name>);
```

Advanced Configuration Information

Flash Programming

The MachXO2's internal Flash memory is the heart of the FPGA's configuration system. It is flexible, allowing you to store the FPGA's configuration data, as well as storing design specific data in the User Flash Memory. It is also a resource that uses a precise erase and programming sequence. Lattice provides several methods for programming the MachXO2 Flash memory:

- : JTAG or Slave SPI programming
- **VMEmbedded**: 'C' source for use with an embedded microprocessor controlling the JTAG port
- **SSPIEmbedded**: 'C' source for use with an embedded microprocessor controlling the SSPI port
- **Custom**: The information in this section, and information from TN1246, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide, permits creation of a custom solution.

The Flash memory space can be accessed by the JTAG port, I^2C port, SPI port, or through the WISHBONE bus. These configuration ports may use offline or transparent programming modes to erase, program, and verify the MachXO2 Flash memory resources. The WISHBONE interface is only permitted to use transparent programming operations. The sequence and timing of the commands presented to the Configuration Logic are identical across all of the configuration ports. There are slight differences due to communication protocol standards when transmitting commands and data. The command and timing flow common to all configuration ports is described first. Protocol variances are described afterward.

Each MachXO2 contains a certain quantity of Configuration Flash memory and User Flash Memory. The amount of memory depends on the device density of the MachXO2. Figure 14-20 shows the number of Flash memory pages available for each MachXO2 device density. Each page represents 128 bits of data.

Table 14-20. Number of Pages of Flash Memory for the MachXO2 Family

MachXO2 Device Density	Configuration Flash (Pages)	UFM (Pages)	CFG + UFM Bridged[1] (Usable Pages)
7000	9,212	2,048	11,257
4000, 2000U	5,758	768	6,524
2000, 1200U	3,198	640	3,836
1200, 640U	2,175	512	2,686
640	1,151	192	1,342
256	575	0	575

1. CFG+UFM (CONFIGURATION = CFGUFM) page count may be less than the sum of its parts due to device limitations.

MachXO2 JEDEC File Format

All Lattice non-volatile devices support JEDEC files. Utilities are available in the Deployment Tool software for converting the JEDEC file into other programming file formats, such as STAPL, SVF, or bitstream (hex or binary). The relevant detail about the JEDEC file is provided in the table below for completeness.

Table 14-21. MachXO2 JEDEC File Format

JEDEC Field	Syntax	Description
Don't Care	My design	Characters appearing before the ^B character are don't care. All character sets or internal language can be used here except ^B.
Start-of-text	^B	^B (Control-B 0x02) marks the beginning of the JEDEC file. Only ASCII characters are legal after ^B. The character * is the delimiter to mark the ending of a JEDEC field. The CR and LF are treated as regular white spaces and have no delimiter function in a JEDEC file.
Header	My design	The first field is the header, which does not have an identifier to indicate its start. Only ASCII characters are legal after ^B. The header is terminated by an asterisk character *.
Field Terminator	*	Each field in the JEDEC file will be terminated with an asterisk.
Note (Comment)	NOTE my design	The key word N marks the beginning of the comment. It can appear anywhere in the JEDEC file. Lattice's JEDEC files add "OTE" to the N key word to make it a more meaningful word NOTE.
Fuse Count	QF3627736	The key word QF identifies the total real fuse count of the device[1].
Default Fuse State	F0 or F1	The key word F identifies the fuse state of those fuses not included in the link field. F0 = fill them with zeros (0), F1 = fill them with ones (1). It is defined for the purpose of reducing JEDEC file size. It has no meaning in Lattice's JEDEC file. Lattice recommends using compression to reduce file size instead.
Security Setting	G0 or G1	JEDEC standard defines G<0,1> to program security <0=no, 1=yes>
OTP and Security Setting	G0, G1, G2, or G3	Lattice enhances the G field to cover OTP fuse programming as well. G<0=both no, 1=only security yes, 2=only OTP yes, 3=both yes>.
Link Field	L0000000 101011…100011 …….. 111111…101100 110 101011…100011 …….. 111111…101100 110 …….. …….. 101011…100011 …….. 111111…101100 110* NOTE SED_CRC* L3627704 111111….111111* CC1B9	The keyword L identifies the first fuse address of the fuse pattern that follows after the white space. The number of digit shown following the L keyword must be the same as that on the QF field. In this example, QF3627736 has seven digits, thus L0000000 should have seven zeros. The fuse address traditionally starts counting from 0. The link field is the most critical portion of the JEDEC file where the programming pattern is stored. The programming data is written into this field in the manner mirroring exactly the fuse array layout of the silicon physically. Row address is written from top to bottom in ascending order: Top = Row 0, Bottom = Last Row. The column address is written from left to right in ascending order: Left most = bit 0, Right most = last bit. Row 0 is selected first by the INIT_ADDRESS command. The first bit to shift into the device is bit 0 for programming. The first to shift out from the device is also bit 0 when verify. The end of the Configuration Flash data is marked by "NOTE END CONFIG DATA*". It is not necessary to program any page data containing all '0' values. UFM pages, if present in the JEDEC, are preceded by a "NOTE TAG DATA*" line. If the JEDEC file is encrypted, all the data in the link field are encrypted. The column size will increase accordingly to include filler bits to make the column size packet (128-bit, or 16 bytes, per packet) bounded.

Table 14-21. MachXO2 JEDEC File Format (Continued)

JEDEC Field	Syntax	Description
Fuse Checksum	CC1B9	The checksum of all the fuses = Fuse count. The fuse state of all the fuses can be found from the Link field. If it is not specified in the link field, then use the Default Fuse State in their places. If the JEDEC file is encrypted, the fuse checksum is calculated after encryption. The fuse checksum prior to encryption can be found on one of the comments.
U Field	UA Home	This is the place to store the 32-bit USERCODE. The 32-bit USERCODE can be expressed in UA = ASCII, UH = ASCII Hex, U = Binary. Lattice enhanced this field for storing the CRC value of encrypted JEDEC[2].
E Field	EH 012..ABCDEF	JEDEC standard defines this field to hold the architecture fuses. Lattice uses this field to store the Feature Row and FEABITS. The Feature Row data is on the first line. The FEABITS values are on line 2.
End-of-text	^C	^C (CTLC) marks the ending of the JEDEC file.
Transmission Checksum	ABCD	This is the checksum of the whole file starting from ^B to ^C. All characters and white space, including the ^B and ^C, are included in the checksum calculation.

1. For encrypted JEDEC file, the first sixteen (16) bits of USERCODE is the CRC value calculated from of the row 0 only; the second sixteen (16) bits is the CRC value calculated from row 0 to the last row.

An example of a MachXO2 unencrypted JEDEC file is shown in Figure 14-16.

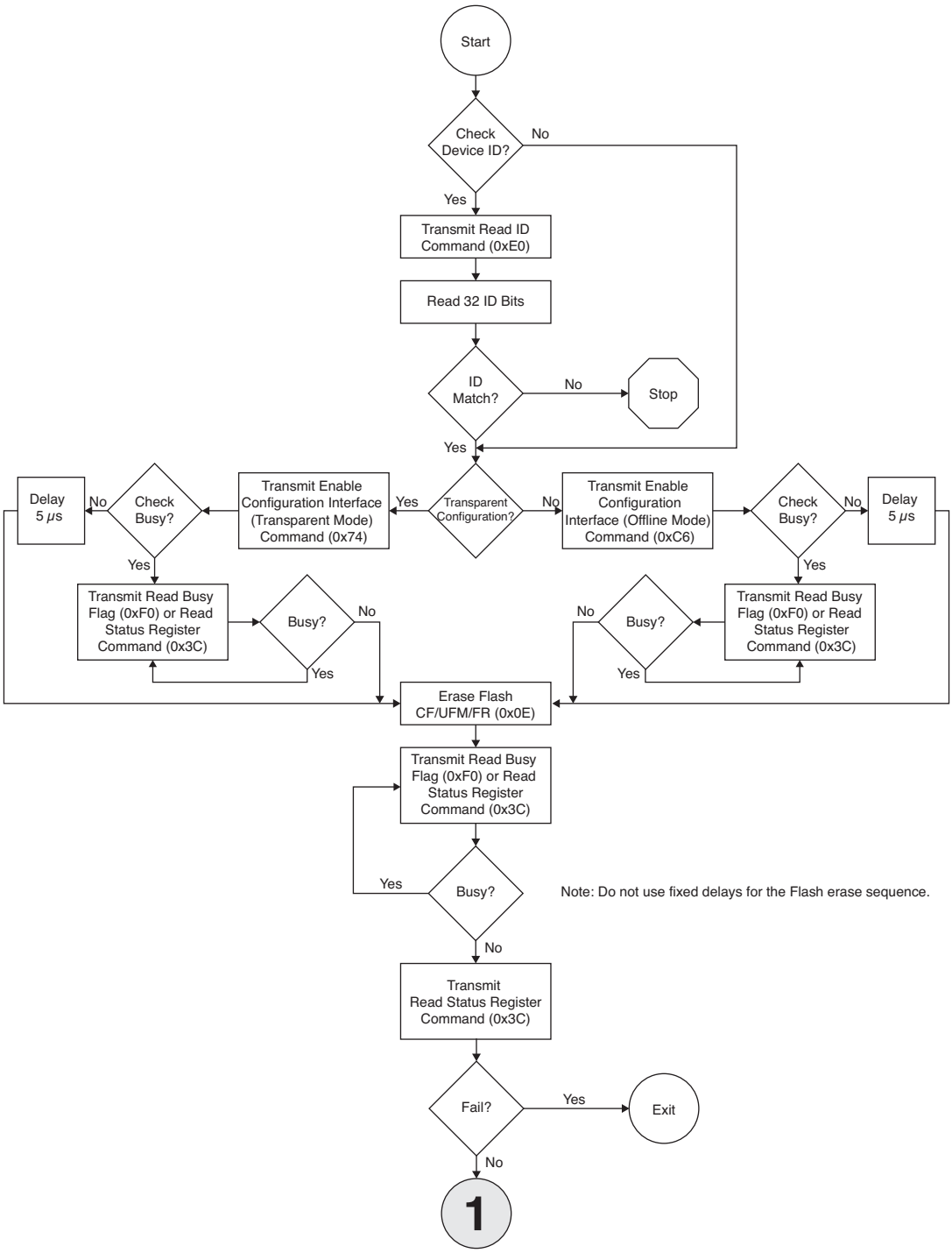
Figure 14-16. Unencrypted JEDEC File Example

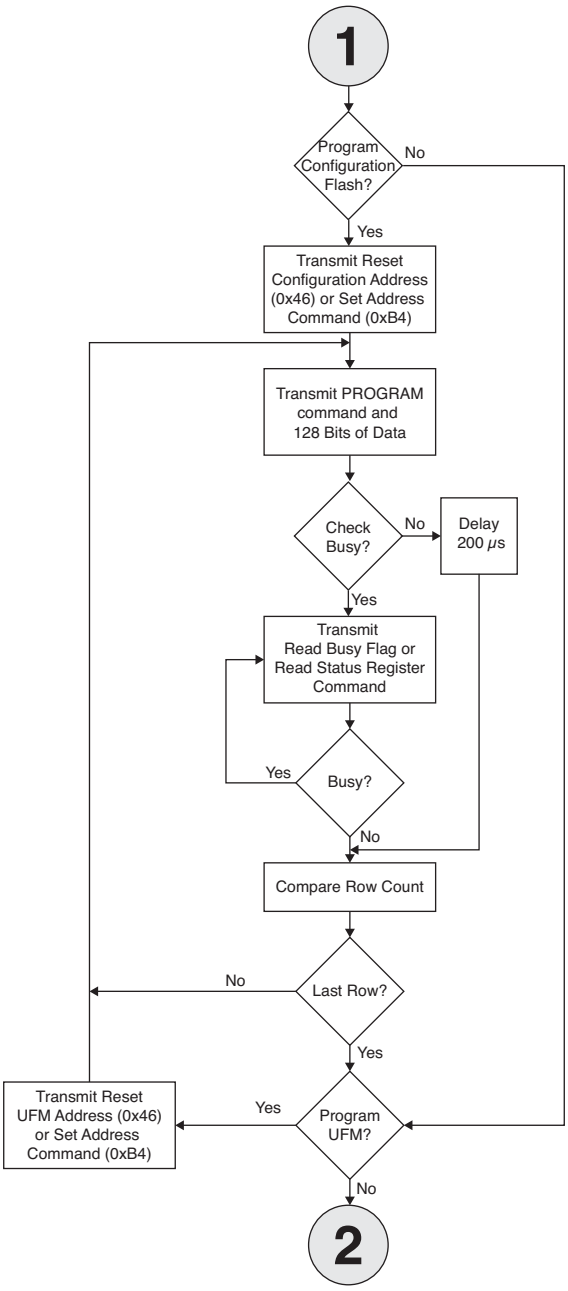
```
*
NOTE Diamond_1.2_Production (92) JEDEC Compatible Fuse File.*
NOTE Copyright (C), 1992-2010, Lattice Semiconductor Corporation.*
NOTE All Rights Reserved.*
NOTE DATE CREATED:Fri Dec 02 14:50:08 2011*
NOTE DESIGN NAME:control_SoC_demo_impl1.ncd*
NOTE DEVICE NAME:LCMXO2-1200ZE-1CSBGA132ES*
NOTE TRACEID 0x84*
NOTE LVDS_72P M9*
NOTE LVDS_72N N10*
NOTE LVDS_68P P8*
NOTE LVDS_68N M8*
NOTE PIN ASSIGNMENTS*
NOTE PINS cap_btn1 : M10 : inout*
NOTE PINS LCD_COM0 : B14 : out*
NOTE PINS xout : N7 : out*
...
NOTE PINS xin : M5 : in*
NOTE PINS lcc_analog_cmp_p : P8 : in*
QP132*
QF343936*
G0*
F0*
L000000
111111111111111110111101101100111111111111111111001110110000000000000000000000000000010000000000000000000000000011000000000110
010000100101000010010000010010000000100111111111010001100000000000000000000000000101110001110000000000001010011010000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000001000100010001000000000000100010000000000010001000000000010001000000000100010001000100000010001000000100010000000
010001000100010000001000100000000000000001000100000000000000000000000000000000000100010001001000000000000100100000000
000000100100000000000010010000000001000100100100000001001000000010010000000100100000000100010001001000000010010000000000000010010
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
...
000000000000000000000000000000000000000010111101011010101111111111111111111111111111111111111111111111111111111111111111101110010
000100000100000000001011100000000000000001100000000000000000011100100001000001010100000101000000001000000000000100001001
111111100000000000000000000000000000000000000011000010100000000000000000000011001010111111101011101010111110111000101000100
001000100000000000000000000000001100000000000000000000000000001111111111111111111111111111111111111111111111111111111111111111
*
NOTE EBR_INIT DATA*
L137984
111111111111111111111111111111111101100000000000000000000000000000000000000011000000000010110010000100000000000010000000
000111101000111101000111101000111101000111101000111101000111101000111101000111101000111101000111101000111101000111101000111100
011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110001
...
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
001000100000000000000000000000000000000000000000000000000000000010111100000000000000000000000011111111111111111111111111111111
*
NOTE END CONFIG DATA*
L184832
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
...
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
*
NOTE TAG DATA*
L343808
010001000100010000001000100000000000000001000100000000000000000000000000000000000010001000100100000000000010010000000
000000100100000000000010010000000001000100100100000001001000000010010000000100100000000100010001001000000010010000000000000010010
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
*
C5CC8*
NOTE FEATURE_ROW*
E00000000000000000000000000000010000100001000000000000000000
0000010000000000*
NOTE User Electronic Signature Data*
UHCAFEBABE*
 6243
```

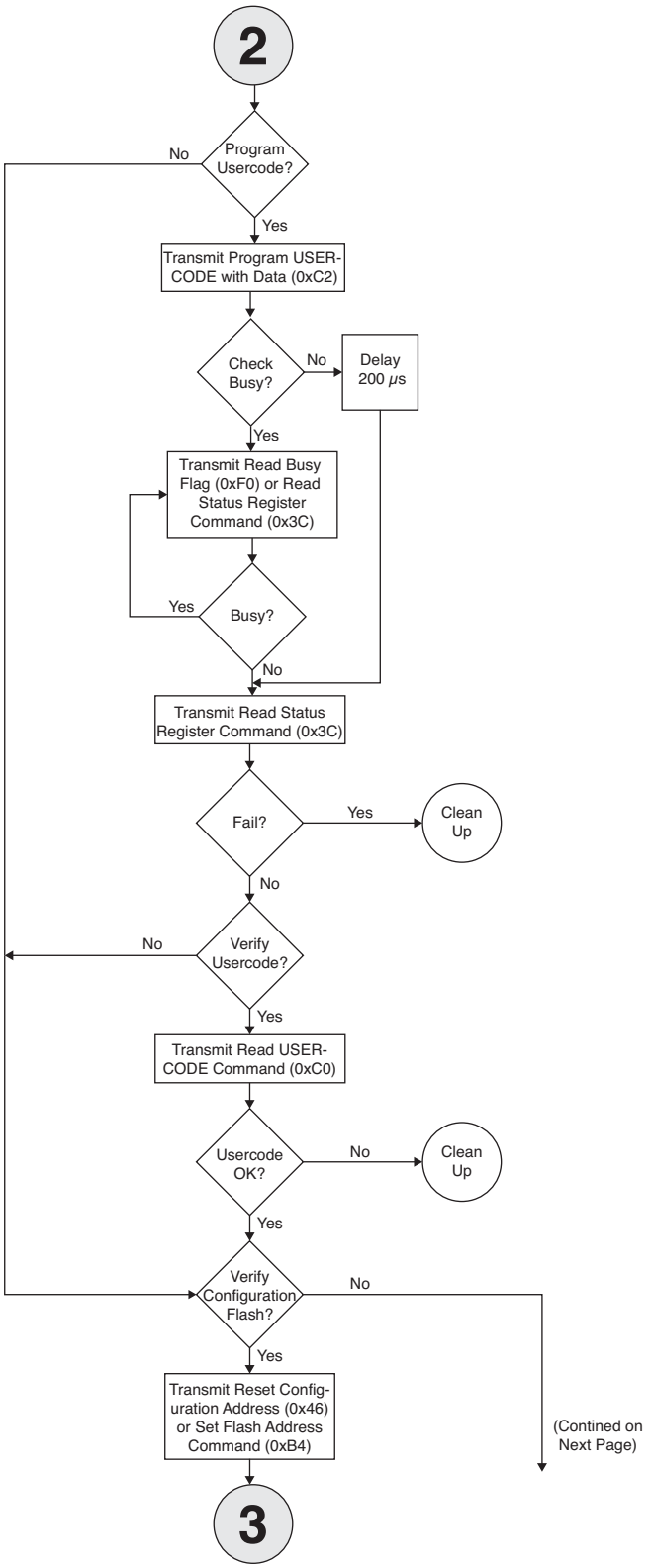
MachXO2 Flash Memory Programming Flow

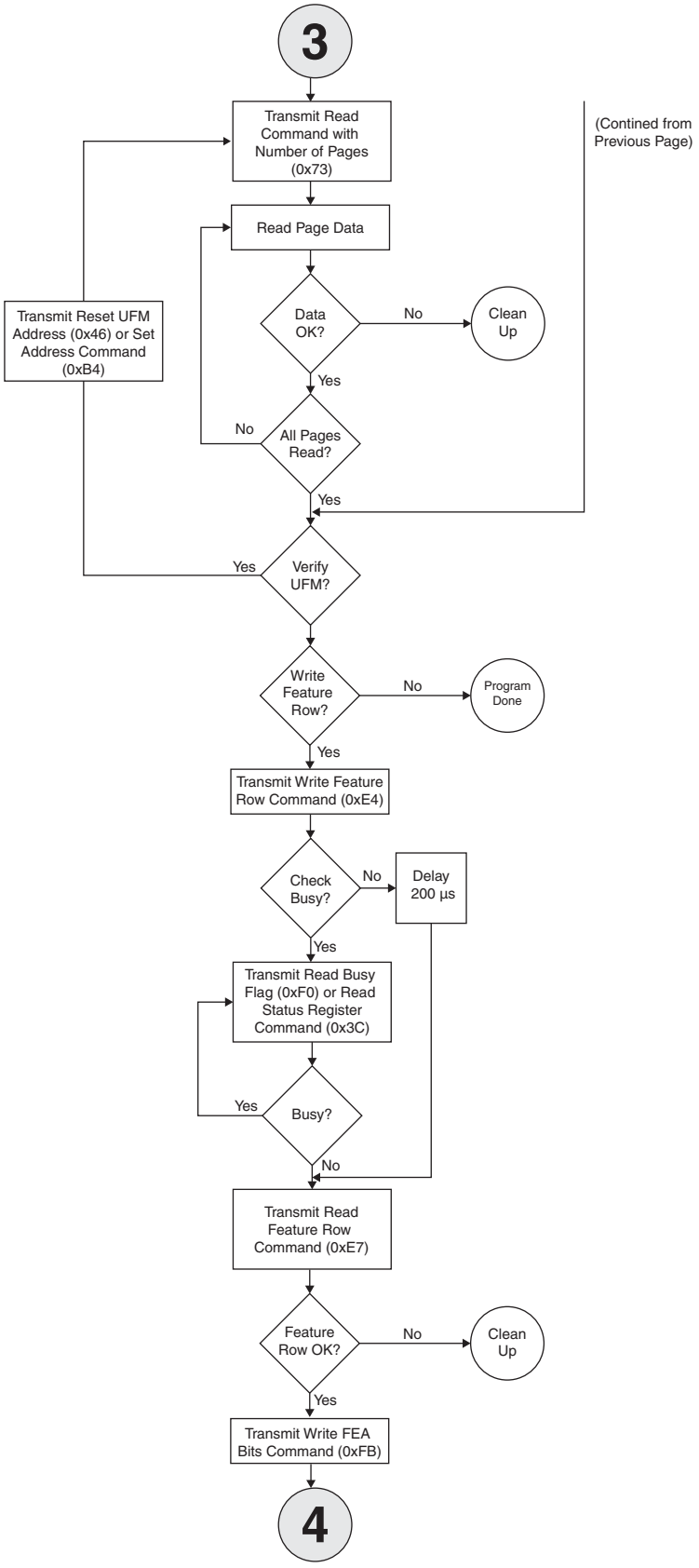
The MachXO2 Flash memory erasure, and programming requires a specific set of steps and timing. The flow chart in this section describes the command sequences and the timing required for successful Flash programming. The commands and timing are common between all of the configuration ports. There are some minor variations in the protocol, but not the timing, based on the configuration port used. Exceptions are described in the configuration port specific sections.

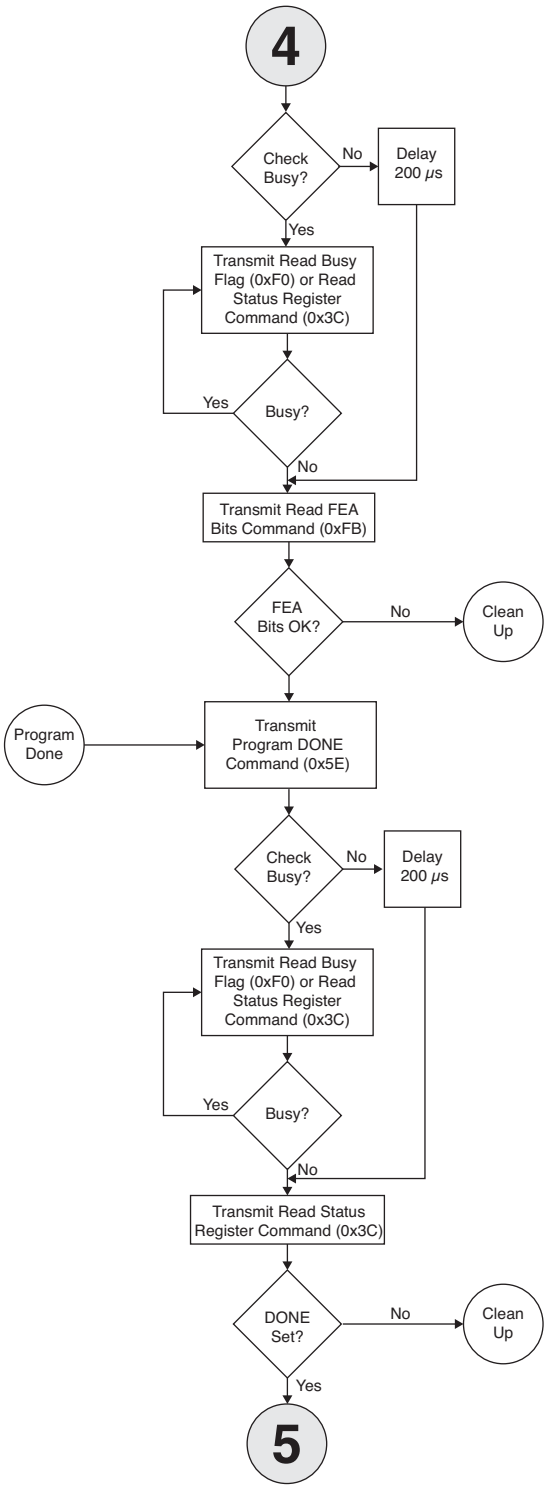
Figure 14-17. MachXO2 Flash Memory Programming Flow

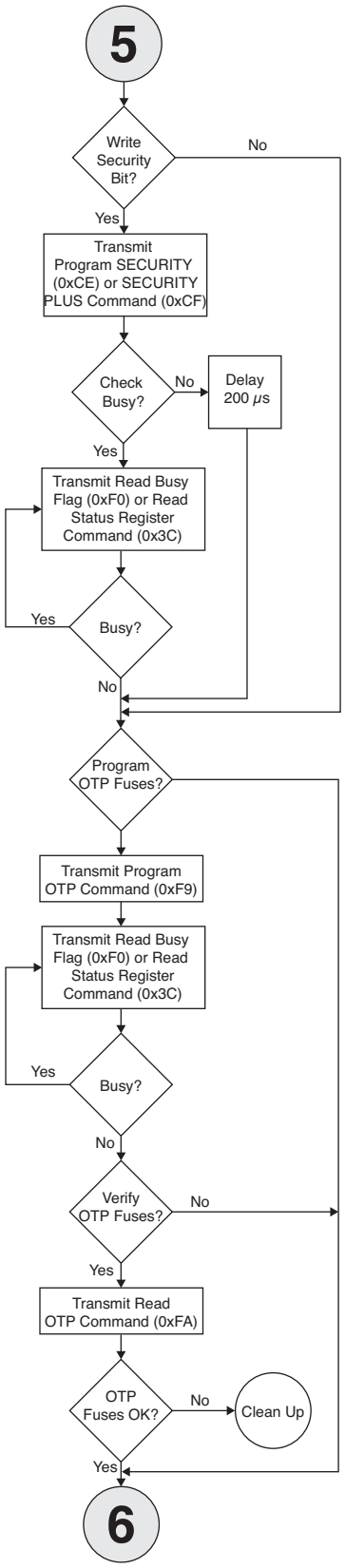


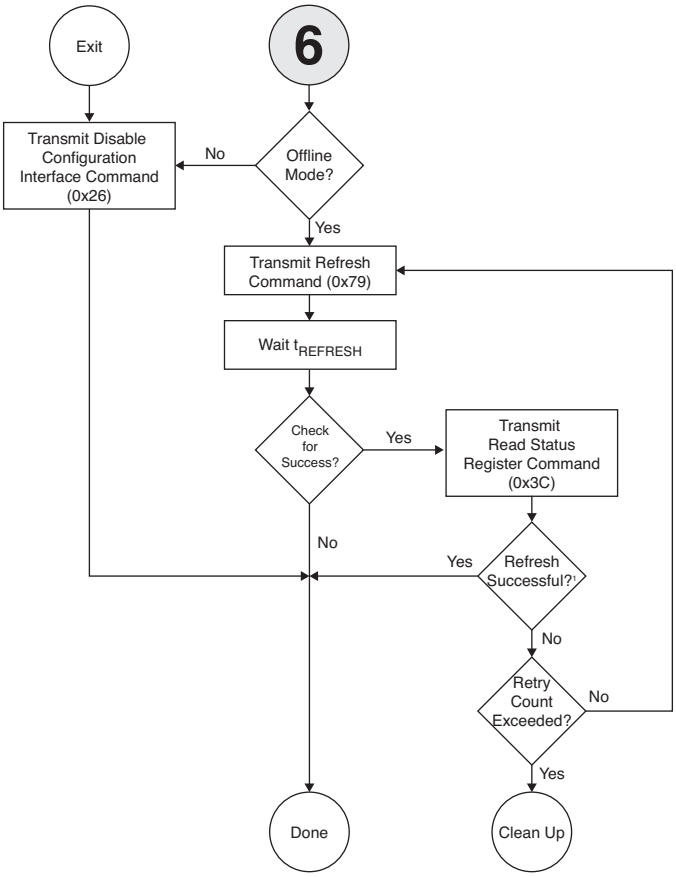

1
Program Configuration Flash?
No
Yes
Transmit Reset Configuration Address (0x46) or Set Address Command (0xB4)
Transmit PROGRAM command and 128 Bits of Data
Check Busy?
No
Delay 200 µs
Yes
Transmit Read Busy Flag or Read Status Register Command
Yes
Busy?
No
Compare Row Count
No
Last Row?
Yes
Transmit Reset UFM Address (0x46) or Set Address Command (0xB4)
Yes
Program UFM?
No
2


2
Program Usercode?
No
Yes
Transmit Program USER-CODE with Data (0xC2)
Check Busy?
No
Delay 200 µs
Yes
Transmit Read Busy Flag (0xF0) or Read Status Register Command (0x3C)
Busy?
Yes
No
Transmit Read Status Register Command (0x3C)
Fail?
Yes
Clean Up
No
Verify Usercode?
No
Yes
Transmit Read USER-CODE Command (0xC0)
Usercode OK?
No
Clean Up
Yes
Verify Configuration Flash?
No
Yes
Transmit Reset Config-uration Address (0x46) or Set Flash Address Command (0xB4)
(Contined on Next Page)
3

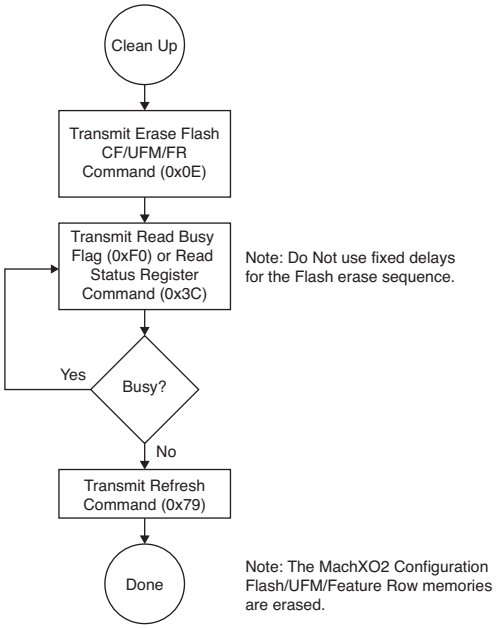

3
Transmit Read Command with Number of Pages (0x73)
(Contined from Previous Page)
Read Page Data
Data OK?
No
Clean Up
Yes
Transmit Reset UFM Address (0x46) or Set Address Command (0xB4)
No
All Pages Read?
Yes
Yes
Verify UFM?
Write Feature Row?
No
Program Done
Yes
Transmit Write Feature Row Command (0xE4)
Check Busy?
No
Delay 200 µs
Yes
Transmit Read Busy Flag (0xF0) or Read Status Register Command (0x3C)
Yes
Busy?
No
Transmit Read Feature Row Command (0xE7)
Feature Row OK?
No
Clean Up
Yes
Transmit Write FEA Bits Command (0xFB)
4


4
Check Busy?
No
Delay 200 µs
Yes
Transmit Read Busy Flag (0xF0) or Read Status Register Command (0x3C)
Yes
Busy?
No
Transmit Read FEA Bits Command (0xFB)
FEA Bits OK?
No
Clean Up
Program Done
Transmit Program DONE Command (0x5E)
Check Busy?
No
Delay 200 µs
Yes
Transmit Read Busy Flag (0xF0) or Read Status Register Command (0x3C)
Yes
Busy?
No
Transmit Read Status Register Command (0x3C)
DONE Set?
No
Clean Up
Yes
5


5
Write Security Bit?
No
Yes
Transmit Program SECURITY (0xCE) or SECURITY PLUS Command (0xCF)
Check Busy?
No
Delay 200 µs
Yes
Transmit Read Busy Flag (0xF0) or Read Status Register Command (0x3C)
Busy?
Yes
No
Program OTP Fuses?
Transmit Program OTP Command (0xF9)
Transmit Read Busy Flag (0xF0) or Read Status Register Command (0x3C)
Busy?
Yes
No
Verify OTP Fuses?
No
Yes
Transmit Read OTP Command (0xFA)
OTP Fuses OK?
No
Clean Up
Yes
6



1. Refresh was successful if Status Register BUSY bit = 0, DONE bit = 1, and Configuration Check Status bits = 000. See TN1246, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide for complete Read Status Register command details.


Clean Up
Transmit Erase Flash CF/UFM/FR Command (0x0E)
Transmit Read Busy Flag (0xF0) or Read Status Register Command (0x3C)
Note: Do Not use fixed delays for the Flash erase sequence.
Yes
Busy?
No
Transmit Refresh Command (0x79)
Done
Note: The MachXO2 Configuration Flash/UFM/Feature Row memories are erased.

MachXO2 Programming Commands

Table 14-22. MachXO2 sysCONFIG Programming Commands

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Read Device ID [IDCODE_PUB]	0xE0	00 00 00	N/A	YY YY YY YY	YY characters represent the device-specific ID code
Enable Configuration Interface (Transparent Mode) [ISC_ENABLE_X]	0x74	08 00 00[1]	N/A	N/A	Enable the Configuration Logic for device programming in transparent mode.[1]
Enable Configuration Interface (Offline Mode) [ISC_ENABLE]	0xC6	08 00 00[1]	N/A	N/A	Enable the Configuration Logic for device programming in Offline mode.[1]
Read Busy Flag [LSC_CHECK_BUSY]	0xF0	00 00 00	N/A	YY	Bit 1 0 7 Busy Ready
Read Status Register [LSC_READ_STATUS]	0x3C	00 00 00	N/A	YY YY YY YY	Bit 1 0 12 Busy Ready 13 Fail OK
Erase [ISC_ERASE]	0x0E	0Y 00 00	N/A	N/A	Y = Memory space to erase Y is a bitwise OR Bit 1=Enable 17 Erase Feature Row 18 Erase Configuration Flash 19 Erase UFM
Erase UFM [LSC_ERASE_TAG]	0xCB	00 00 00	N/A	N/A	Erase the UFM sector only.
Reset Configuration Flash Address [LSC_INIT_ADDRESS]	0x46	00 00 00	N/A	N/A	Set Page Address pointer to the beginning of the Configuration Flash sector
Set Address [LSC_WRITE_ADDRESS]	0xB4	00 00 00	M0 00 PP PP	N/A	Set the Page Address pointer to the Flash page specified by the least significant 14 bits of the PP PP field. The 'M' field defines the Flash memory space to access. Field 0x0 0x4 M Configuration Flash UFM
Program Page [LSC_PROG_INCR_NV]	0x70	00 00 01	YY * 16	N/A	Program one Flash page. Can be used to program the Configuration Flash, or UFM.
Reset UFM Address [LSC_INIT_ADDR_UFM]	0x47	00 00 00	N/A	N/A	Set the Page Address Pointer to the beginning of the UFM sector
Program UFM Page [LSC_PROG_TAG]	0xC9	00 00 01	YY * 16	N/A	Program one UFM page
Program USERCODE [ISC_PROGRAM_USERCODE]	0xC2	00 00 00	YY * 4	N/A	Program the USERCODE.
Read USERCODE [USERCODE]	0xC0	00 00 00	N/A	YY * 4	Retrieves the 32-bit USERCODE value
Write Feature Row [LSC_PROG_FEATURE]	0xE4	00 00 00	YY * 8	N/A	Program the Feature Row bits
Read Feature Row [LSC_READ_FEATURE]	0xE7	00 00 00	N/A	YY * 8	Retrieves the Feature Row bits
Write FEABITS [LSC_PROG_FEABITS]	0xF8	00 00 00	YY * 2	N/A	Program the FEA bits
Read FEABITS [LSC_READ_FEABITS]	0xFB	00 00 00	N/A	YY * 2	Retrieves the FEA bits
Read Flash [LSC_READ_INCR_NV]	0x73	M0 PP PP	N/A	See "Reading Flash Pages" on page 14-47	Retrieves PPPP count pages. Only the least significant 14 bits of PP PP are used. The 'M' field must be set based on the configuration port being used to read the Flash memory. 0x0 I^2C 0x1 JTAG/SSPI/WB

Table 14-22. MachXO2 sysCONFIG Programming Commands (Continued)

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Read UFM Flash [LSC_READ_UFM]	0xCA	M0 PP PP	N/A	See "Reading Flash Pages" on page 14-47	Retrieves PPPP count UFM pages. Only the least significant 14 bits of PP PP are used for the page count. The 'M' field must be set based on the configuration port being used to read the UFM. 0x0 I^2C 0x1 JTAG/SSPI/WB
Program DONE [ISC_PROGRAM_DONE]	0x5E	00 00 00	N/A	N/A	Program the DONE status bit enabling SDM
Program OTP Fuses [LSC_PROG_OTP]	0xF9	00 00 00	UCFSUCFS	N/A	Makes the selected memory space One Time Programmable. Matching bits must be set in unison to activate the OTP feature. Bit 1 0 0, 4 SRAM OTP SRAM Writable 1, 5 Feature Row OTP Feature Row Writable 2, 6 CF OTP CF Writable 3, 7 UFM OTP UFM Writable
Read OTP Fuses [LSC_READ_OTP]	0xFA	00 00 00	N/A	UCFSUCFS	Read the state of the One Time Programmable fuses. Bit 1 0 0, 4 SRAM OTP SRAM Writable 1, 5 Feature Row OTP Feature Row Writable 2, 6 CF OTP CF Writable 3, 7 UFM OTP UFM Writable
Disable Configuration Interface [ISC_DISABLE]	0x26	00 00	N/A	N/A	Exit Offline or Transparent programming mode. The command causes the MachXO2 to reconfigure when leaving Offline mode. The Configuration SRAM must be cleared prior to transmitting the Disable Configuration Interface command.
Bypass [ISC_NOOP]	0xFF	FF FF FF	N/A	N/A	No Operation and Device Wakeup
Refresh [LSC_REFRESH]	0x79	00 00	N/A	N/A	Force the MachXO2 to reconfigure. Transmitting a REFRESH command reconfigures the MachXO2 in the same fashion as asserting PROGRAMN.
Program SECURITY [ISC_PROGRAM_SECURITY]	0xCE	00 00 00	N/A	N/A	Program the Security bit (Secures CFG Flash sector).[2]
Program SECURITY PLUS [ISC_PROGRAM_SECPLUS]	0xCF	00 00 00	N/A	N/A	Program the Security Plus bit (Secures CFG and UFM Sectors).[2]
Read TraceID code [UIDCODE_PUB]	0x19	00 00 00	N/A	YY*8	Read 64-bit TraceID.

1. Transmit the command opcode and first two operand bytes when using the I^2C port. The final operand byte must not be transmitted.
2. SECURITY and SECURITY PLUS commands are mutually exclusive.

Reading Flash Pages

Reading the Configuration Flash and the User Flash Memory pages requires a specific procedure. The Configuration Flash and UFM pages are accessible from any of the MachXO2's configuration ports. The JTAG, Slave SPI, and the WISHBONE configuration ports all behave identically when performing read operations. The I^2C port requires a modified access protocol. A high-level representation of the data flow, by port, is shown in Figure 14-19.

All ports start the read process in the same way, by sending a Read Flash/Read UFM Flash command. The MachXO2 begins the read process once the command byte has been accepted by the configuration logic. The Page Address Pointer determines the first page returned from the MachXO2. For the first returned page to be valid (e.g. for single-page read operations), a Retrieval delay of 240ns must be observed. The Retrieval delay time is from the end of the Command byte transmission to the end of the first Operand byte transmission See Figure 14-18. Note that for slower interface clock rates, 240ns may be consumed entirely by the normal transmission of the first Operand and no additional delay may be necessary.

Figure 14-18. Retrieval Delay Timing Requirement for Single-Page Reads

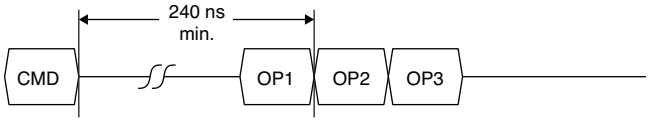
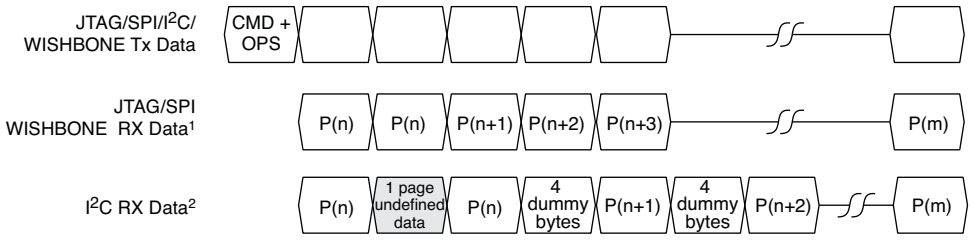


Figure 14-19. Flash Page Command and Data Sequence



Notes:

1. JTAG/SSPI must transmit data in order to read data back. The data sent by the JTAG/SSPI master is not specified (i.e. don't care).

2. The I^2C must use RESTART between sending the CMD and reading the data. (Issuing a STOP terminates a CMD and resets the I^2C state machine.)

CMD + OPS = Read Flash or Read UFM Flash command byte + 3 operand bytes.

Figure 14-19 shows a multiple page read sequence. The Read Page, or Read UFM Page command is transmitted to the MachXO2. As can be seen in Figure 14-19, all interfaces return the page at the Page Address Pointer immediately. For single-page read operations, all configuration ports are allowed to terminate the read immediately following the transfer of the final byte of the first page. The I^2C interface differs only in the Read Flash/Read UFM Flash operand bytes.

Reading more than one page requires special handling. The multiple page read duplicates the page selected by the Page Address Pointer. The result of this behavior is that the page count must be one greater than the desired number of pages. For example, reading two pages requires the page count supplied in the Read Flash/Read UFM Flash command to be assigned a value of 3. If the Page Address Pointer is 0000, the MachXO2 will return three pages, Page 0, Page 0, and Page 1.

The I^2C interface has additional overhead when reading Flash memory pages. Reviewing Figure 14-19 shows how the data is presented during a multiple page read request. When the page count is three, and the Page Address Pointer is 0000, the I^2C interface will return Page 0, 16 undefined bytes, Page 0, 4 dummy bytes, and Page 1. Reading the final four dummy bytes is optional.

References

- MachXO2 Family Data Sheet
- TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices
- TN1207, Using TraceID in MachXO2 Devices

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
May 2011	01.1	Added Appendix A and Appendix B.
May 2011	01.2	Corrected I^2C Function for the yyyxxxxx11 slave address in the Slave Addresses for I^2C Ports table.
August 2011	01.3	Added User SPI during transparent programming caution.
		Added external SPI address for dual boot option.
February 2012	01.4	Document status changed from Advance to Final.
		Updated document with new corporate logo.
June 2012	02.0	Major update, including: • Updated Programming algorithm • Added Feature Row discussion • Improved coverage of configuration port management
July 2012	02.1	Clarified SECURITY/SECURITY PLUS in Figure 14-16, MachXO2 Flash Memory Programming Flow, and Table 14-21, MachXO2 sysCONFIG Programming Commands.
		Added Figure 14-17, Retrieval Delay Timing Requirement for Single-Page Reads.
		Clarified Retrieval delay in Figure 14-18, Flash Page Command and Data Sequence.
		Added missing 'Program DONE' step in Flash Memory Programming Flow.
September 2012	02.2	Updated TransFR operation.
		Enhanced programming flow chart.
		Updated PROGRAMN configuration pin information.
		Added details about MUX_CONFIGURATION_PORTS.
October 2012	02.3	Added restriction: Primary port can be used as Configuration/UFM port or as User port, but not both.
April 2013	02.4	MachXO2 sysCONFIG Programming Commands table – Updated the Write Data field for the Set Address command.
		MachXO2 Flash Memory Programming Flow diagram – Added footnote.
		Added information on configuring from an external SPI Flash.
		Added Internal Flash Memory in Definition of Terms.
		Updated CFGUFM configuration preference information.
		Updated sysCONFIG Preferences figure.
		MachXO2 Feature Row Elements table – Updated contents and added the HW Default State (Erased) column.

Using TraceID in MachXO2 Devices

February 2012 Technical Note TN1207

Introduction

Design theft has caused many companies to explore methods to insure that their designs and intellectual property (IP) is protected or made less prone to blatant copying.

Design theft occurs when a design is copied in part or in whole, then designed into a cheaper competing product.

The MachXO2™ PLD family has a feature called TraceID for securing original design and IP. TraceID is a unique, 64-bit code that is programmed during manufacturing of the device, thus linking a specific design to a specific device. This ensures that only the original product manufacturer who ordered a specific device has access to it.

Why is TraceID Important?

TraceID can be used to prevent overbuilding and cloning of user designs. Overbuilding occurs when a contract manufacturer builds more products than the original company has approved. These extra products are, in turn, sold through other channels for profit without the knowledge or consent of the original company. Cloning is the act of making exact copies of a product and selling them under a different name at a lower price (thus reducing the OEM profit). These practices will cause OEMs to lose money not only from lost sales and lower margins, but also from unseen support costs such as failure analysis.

The MachXO2 TraceID feature can be used to prevent both overbuilding and cloning.

How Does TraceID Work?

TraceID is a unique, 64-bit device identification tracking number which is stored in the feature row of the device. The 64 bits contain the following unique information:

- Wafer lot number
- Wafer number within the lot
- Die X location
- Die Y location
- User-defined design-specific code

The MachXO2 TraceID register format is shown in Figure 15-1.

Figure15-1. TraceID Register

63 56	55 0
User-Defined Code (8 bits)	Factory-Programmed — Wafer Lot, Wafer Number, Die X Location, Die Y Location (56 bits)

The most significant eight bits are the user-defined design-specific code. These eight bits are read and write accessible. The remaining 56 bits are read-only and are programmed at the time the device is manufactured by Lattice. The 8-bit user-defined code plus the 56-bit factory-programmed portion together guarantee that every device will have a unique TraceID. This uniqueness provides OEMs greater control over how many of their products are introduced in the market and the ability to detect false products.

How to Program User-Defined Code of the TraceID

Lattice design software can be used to set a specific 8-bit user-defined code in the TraceID register. The TRACE_ID_BINARY preference must be set to a chosen value in the LPF file.

In the LPF file, set the TraceID value using the following format:

```
TRACEID "<8-bit value>"
```

Below is an example:

```
TRACEID "01101001"
```

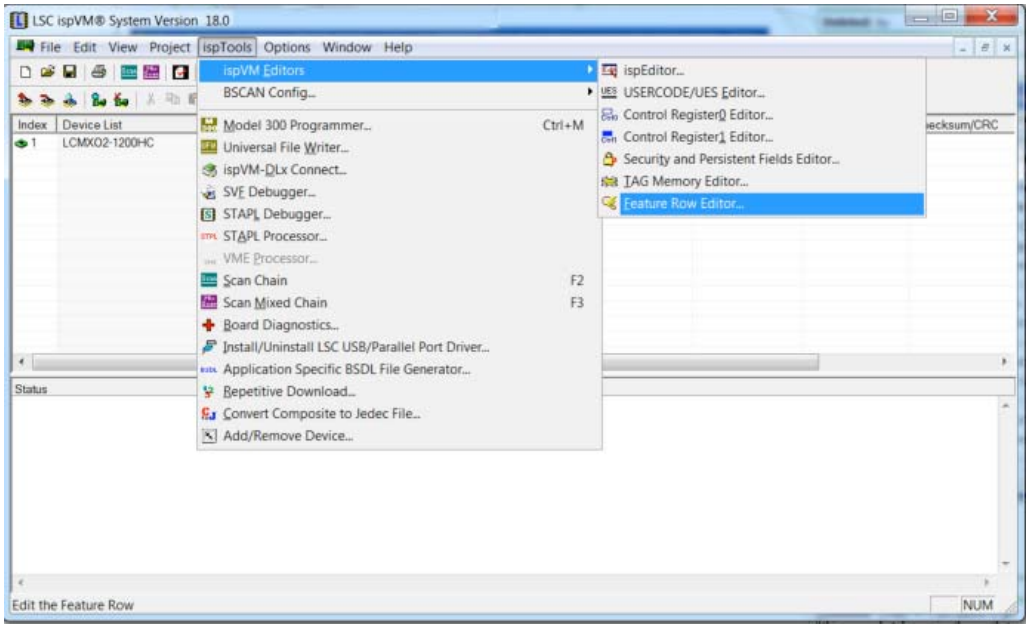
When a programming file is created, the 8-bit TraceID value is embedded in the JED file feature row. During programming, the TraceID registers in the device are updated with the one in the JED. The default value for the user defined code is "00000000".

Accessing the TraceID Register

The TraceID value in the MachXO2 device can be read using the internal WISHBONE port or externally through the JTAG, SSPI or I^2C ports. For SSPI, I^2C and JTAG interfaces, the UIDCODE_PUB command must be used to read the TraceID register. The OPCODE for the UIDCODE_PUB command is "00011001".

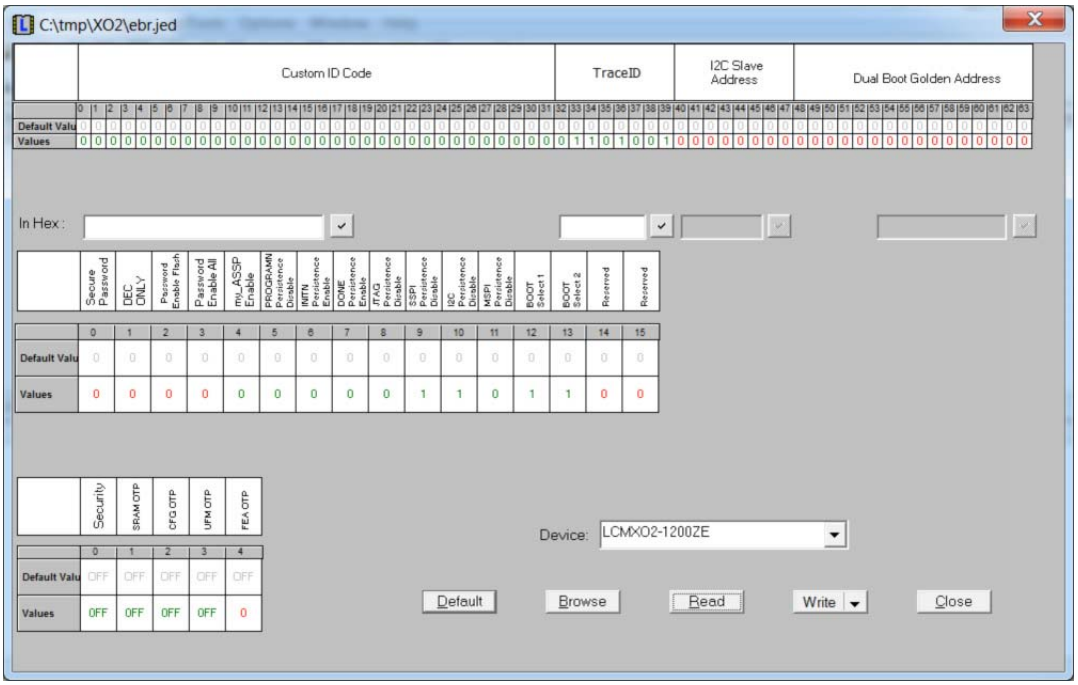
The TraceID value can also be read using the ispVM™ programming tool. From the main menu select **ispTools > ispVM Editors > Feature Row Editor**.

Figure15-2. ispVM System



The window shown in Figure 15-3 will open. From here, the Feature Row which includes the TraceID, can be read from and written to.

Figure15-3. Feature Row Editor



TraceID Access Through the JTAG Port

All MachXO2 devices have a dedicated JTAG port, which is compliant with the IEEE 1149.1 and IEEE 1532 specifications. The JTAG port has access to all configuration logic resources including the TraceID. To read the TraceID using JTAG, shift the command (0x19h) into the instruction register then read the 64-bit TraceID out of the data register.

TraceID Access Through the WISHBONE Slave Interface

The WISHBONE Slave interface of the EFB module enables designers to access the TraceID directly from the PLD core logic. The WISHBONE bus signals are utilized by a WISHBONE host that designers can implement using the general purpose PLD resources. In addition to the WISHBONE bus signals, an interrupt request output signal is brought to the PLD fabric.

The WISHBONE interface communicates to the configuration logic through a set of data, control and status registers. Table 15-1 shows the register names and their functions. These registers are a subset of the EFB register map. The detail of the WISHBONE slave interface pins, EFB register map, and WISHBONE register definition can be found in TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

Table15-1. WISHBONE Registers

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

When using the WISHBONE bus interface, the opcodes, operand and data are written to the CFGTXDR register. This is required only when communicating with the configuration logic inside the MachXO2 device. The TraceID can be accessed via the WISHBONE interface by writing the opcode and operand into the CFGTXDR register. The TraceID information can then be read from the CFGRXDR register.

The opcode to access the TraceID is 0x19h and the operand is 0x000000h.

TraceID Access Through the Slave SPI Port

All MachXO2 devices have a dedicated Slave SPI port, which can be used to perform read operations to the TraceID. The configuration SPI port is shared with the hardened SPI core of the EFB module. Asserting the configuration SN (select) pin will cause the SPI port to transition its service from user mode to configuration mode. The TraceID can be accessed using the SPI port by following the command sequence described below.

1. Pull down the configuration SN select pin (SPI Slave Select)
2. Send 8'h19 command from the external SPI master
3. Send 24-bit operand 24'h000000
4. Receive TraceID from SPI slave in next 64 SCLK cycle
5. Pull up configuration SN select pin

The complete sequence is shown in Figures 15-4 and 15-5.

Figure15-4. TraceID Read via SPI

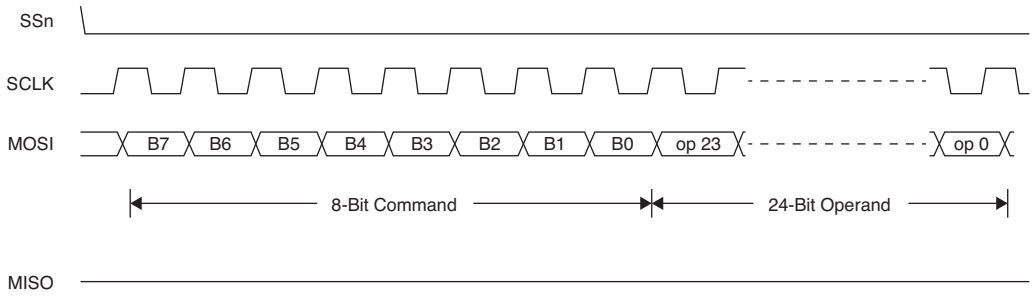
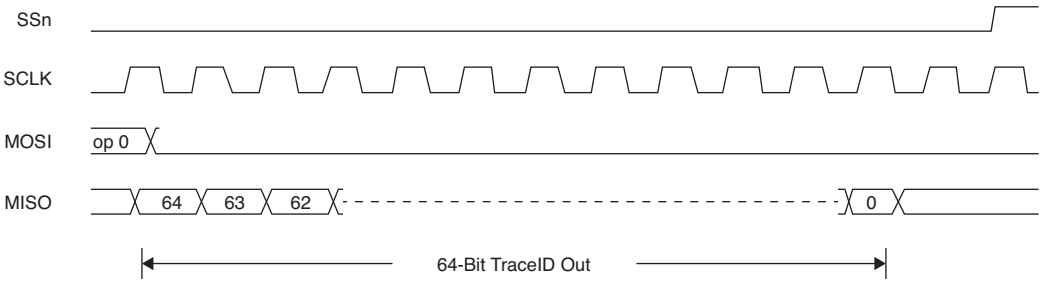


Figure15-5. TraceID Read via SPI, Continued



TraceID Access Through the I²C Port

All MachXO2 devices have a dedicated I²C port, which can be used to perform read operations to the TraceID. The configuration I²C port is shared with the hardened I²C primary core of the EFB module. Addressing the I²C primary port with the configuration address will change the port service from user mode with the WISHBONE interface to configuration mode. The pin locations of the configuration I²C port are pre-assigned in all MachXO2 devices.

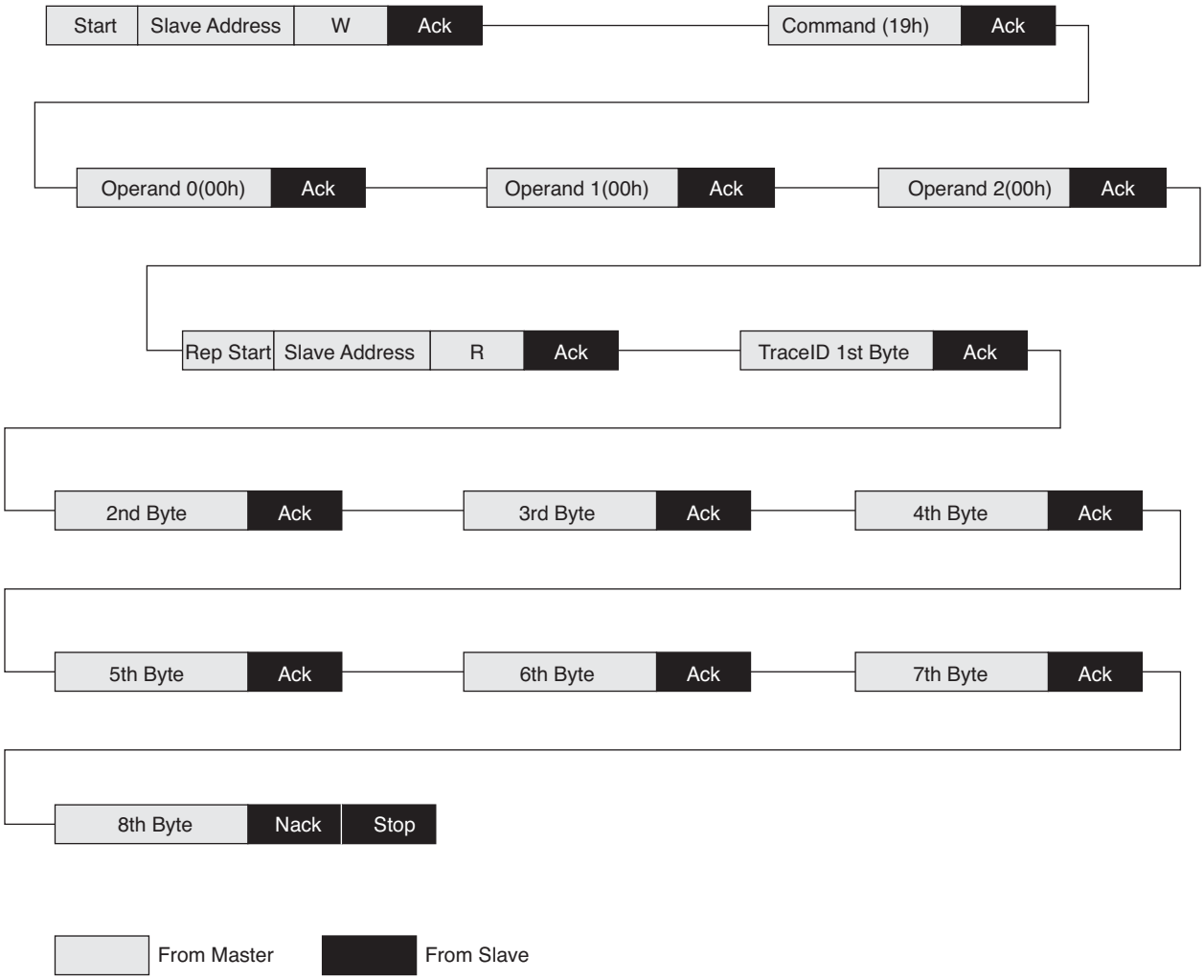
There is one address byte required since 7-bit addresses are used. The last bit of the address byte is the read/write bit and should always be set according to the required operation. This 7-bit I²C address is 1000000 (80h) which is the default address. The read sequence uses a repeated start condition during the sequence to avoid bus release

during communication. For 10-bit addressing the I²C slave address will be 10'b1111000000. To read the TraceID via the I²C bus, follow the steps below.

1. Send start condition.
2. Send default slave address (8'h80) and write command.
3. Send the 8-bit command 8'h19.
4. Send the 24-bit operand 24'h000000 in three single-byte transfers.
5. Send repeated start.
6. Send the slave address and read command.
7. Read the first byte of the TraceID and send ack.
8. Read the second to seventh bytes of the TraceID and send ack for each byte read.
9. Read the last TraceID byte and send nack.
10. Send the stop command.

Table 15-6 shows the TraceID read via I²C.

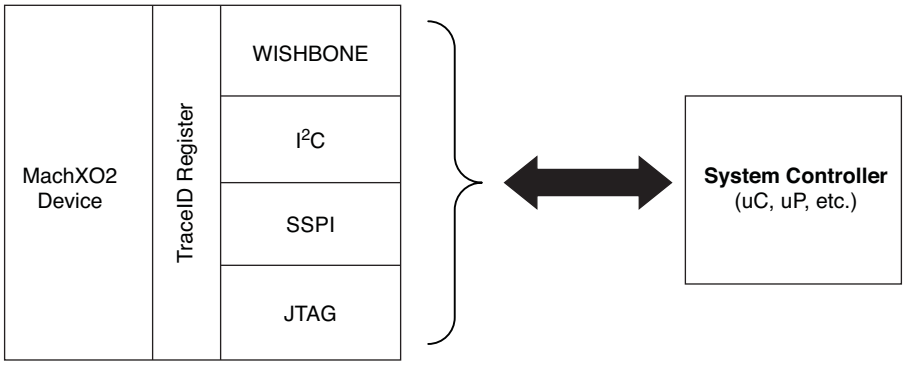
Figure15-6. TraceID Read via I²C



Example Uses of TraceID

TraceID can used to validate that the MachXO2 device or the system in general, as authorized by the OEM.

Figure15-7. Example System-level Use of TraceID



One way of implementing this is to use the system controller, either a microcontroller or microprocessor, to access the TraceID register (via WISHBONE, I^2C, SPI or JTAG interfaces) and compare it against a list of approved TraceID device tables. If the TraceID matches the approved device list, the system can continue to function as intended.

In cases where the read TraceID does not match with the approved device list, the system controller can choose to log the event and take one of the following actions:

- Stall – Stop working
- Continue with limited functionality – Partial operation of system
- Erase or destroy integral data in the system – Erase the boot ROM, Flash memory, register tables, etc.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
February 2012	01.1	Document status changed from advance to final.
		Updated document with new corporate logo.
		Added the following new sections: • TraceID Access Through the JTAG Port • TraceID Access Through the WISHBONE Slave Interface • TraceID Access Through the Slave SPI Port • TraceID Access Through the I^2C Port

MachXO2 SED Usage Guide

February 2013 Technical Note TN1206

Introduction

Memory errors can occur when high-energy charged particles alter the stored charge in a memory cell in an electronic circuit. The phenomenon first became an issue in DRAM, requiring error detection and correction for large memory systems in high-reliability applications. As device geometries have continued to shrink, the probability of memory errors in SRAM has become significant for some systems. Designers are using a variety of approaches to minimize the effects of memory errors on system behavior.

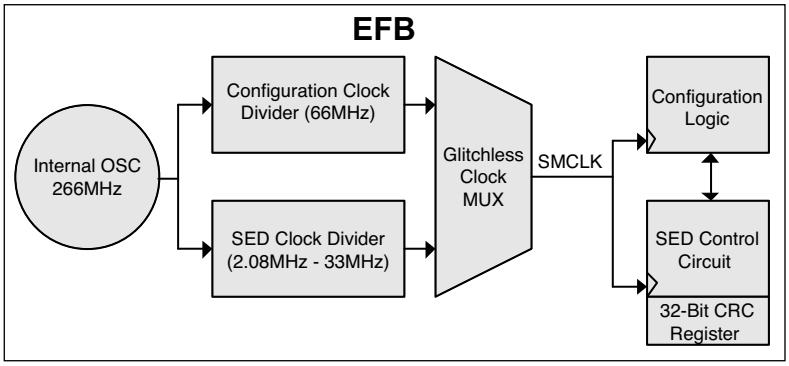
SRAM-based PLDs store logic configuration data in SRAM cells. As the number and density of SRAM cells in an PLD increase, the probability that a memory error will alter the programmed logical behavior of the system increases. A number of approaches have been taken to address this issue, but most involve Intellectual Property (IP) cores that the user instantiates into the logic of their design, using valuable resources and possibly affecting design performance. The MachXO2™ devices have a hardware implemented SED circuit which can be used to detect SRAM errors and allow them to be corrected.

This document describes the hardware-based SRAM CRC Error Detect (SED) approach taken by Lattice Semiconductor for MachXO2 PLDs.

SED Overview

The SED hardware in the MachXO2 devices is part of the Embedded Functional Block (EFB) consists of an access point to the PLD's Configuration Logic, a Controller Circuit, and a 32-bit register to store the CRC for a given bitstream (see Figure 17- 1). The SED hardware reads serial data from the PLD's Configuration memory and calculates a CRC. The data that is read, and the CRC that is calculated, does not include EBR memory or PFUs used as RAM. The calculated CRC is then compared with the expected CRC that was stored in the 32-bit register. If the CRC values match it indicates that there has been no configuration memory corruption, but if the values differ an error signal is generated.

Figure 1-1. System Block Diagram



Note that the calculated CRC is based on the particular arrangement of configuration memory for a particular design. Consequently, the expected CRC results cannot be specified until after the design is placed and routed. The Lattice Diamond® or ispLEVER® bitstream generation software analyzes the configuration of a placed and routed design and updates the 32-bit SED CRC register contents during bitstream generation.

The following sections describe the MachXO2 SED implementation and flow.

SED Limitations

SED should only be run when once Vcc reaches the data sheet Vcc minimum recommend level. In addition, clock frequencies of greater than 33.33 MHz for the SED are not supported.

The clock (SMCLK) of the SED circuit is shared with the Configuration Logic. As a result, the SED module interacts with several EFB functions with the following results:

- If the EFB or Configuration Logic is accessed while the SED circuit is running:
  - The current SED cycle will be terminated:
    - When the SED circuit is terminated there will be a delay of two SMCLK cycles before EFB or Configuration Logic can be accessed. This is a result of the SMCLK transferring clock from the SED Clock to the Configuration Clock domain. The two SMCLK cycles are defined by the slower SED clock.
    - When the SED circuit is terminated the SEDDONE will remain low, SEDERR will remain low, and SEDINPROG resets from high to low.
    - The EFB or Configuration Logic access which interacts with the SED circuit is defined as:
      - The following commands issued through the JTAG port or WISHBONE interface:
        - LSC_REFRESH
        - ISC_ENABLE
        - ISC_ENABLE_X
        - All IEEE 1532 instructions
        - ISC_DISABLE
    - Primary I^2C Configuration Logic slave address match
    - SPI Configuration Logic chip select being asserted
- The PROGRAMN pin detection logic requires the minimal low period be longer than six SMCLK cycles. If the SED circuit is running the six SMCLK cycles are defined by the SED clock.

SED Operating Modes

For MachXO2 devices there are two operating modes available for SED:

- Standard mode allows the design to control when the SED is run and to test the error detection operation.
- One-shot operation is used to run the SED once when the device is first configured to ensure that the configuration matches the desired configuration.

Both operations perform a single cycle which checks the CRC of all the bits in the SRAM except the EBR and RAM memory. Standard mode is activated using the SEDFA primitive while the One-Shot operation is activated using the SEDFB primitive. These primitives are described in the next section.

If an error is detected during an SED operation, the user can choose one of two corrective actions to take. One is to “Do Nothing” and the other is to initiate an on-demand user reconfiguration by pulling the PROGRAMN pin low. This can be done from another device or from an output of the MachXO2 device as shown in Figure 1-4.

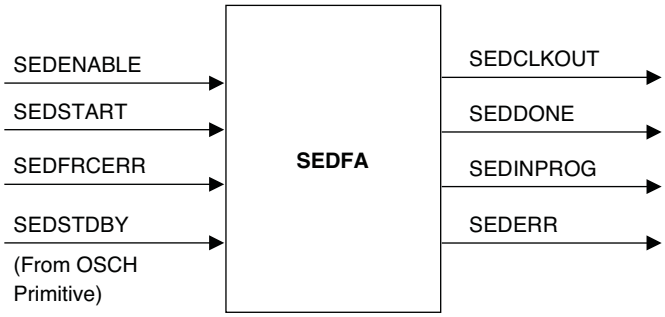
The PROGRAMn pin detection logic requires the minimal low period be longer than 6 SMCLK cycles. When error detection is actively enabled, the PROGRAMn pin minimal low period will be 6 SEDCLK cycles, since the active SMCLK switched to SEDCLK as mentioned previously. This will behave differently then normal operation where the SMCLK is operating at full speed. After booting, the SED function block will behave according to the new configuration programming.

Standard SED

The Standard SED operation can be used by instantiating the SEDFA primitive which is shown in Figure 1-2. The primitive port definitions are listed in Table 1-1. See the Port Descriptions section of this document for more detailed information about each of the ports.

If the Standard SED is done with the “R1” version of the MachXO2 devices, the first time the check is run a false error will be reported. If a second check is done the correct result will be reported. The “R1” versions of the MachXO2 devices have an “R1” suffix at the end of the part number, similar to LCMXO2-1200ZE-1TG144CR1. For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

Figure 1-2. SEDFA Primitive Symbol

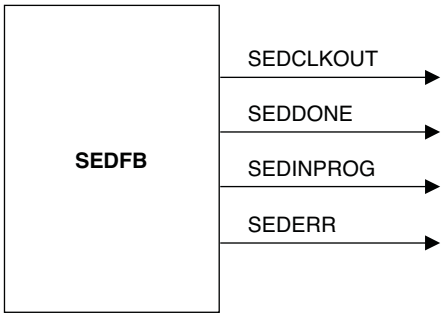


One-Shot SED

The One-Shot SED operation can be used by instantiating the SEDFB primitive which is shown in Figure 1-3. The definitions of the ports for the SEDFB primitive are shown in Table 1-1. See the Port Descriptions section of this document for more detailed information about each of the ports.

The One-Shot SED is NOT supported on the “R1” version of the MachXO2 devices because a false error will always be reported. The “R1” versions of the MachXO2 devices have an “R1” suffix at the end of the part number, similar to LCMXO2-1200ZE-1TG144CR1.

Figure 1-3. SEDFB Primitive Symbol



Signal Descriptions

Table1-1. SEDFA Primitive Port Definitions

Signal Name	Direction	Active	Description
SEDENABLE	Input	High	SRAM CRC enable
SEDSTART	Input	Rising Edge	Start SRAM CRC cycle
SEDFRCERR	Input	Rising Edge	Force an SRAM CRC error flag
SEDSTDBY	Input	High	SRAM CRC disable while in standby mode
SEDCLKOUT	Output	N/A	Output clock
SEDDONE	Output	High	SRAM CRC cycle is complete
SEDINPROG	Output	High	SRAM CRC cycle is in progress
SEDERR	Output	High	SRAM CRC error flag

SED Clock Driver

The SED circuitry is driven by the MachXO2 internal oscillator when using either the SEDFA or SEDFB primitive. The maximum frequency supported is 33.25 MHz.

The MachXO2 internal oscillator can be used for several functions in the device including Configuration, SED and as an internal user clock. The frequency of the oscillator output can be set differently for each of these different uses. The settings available for the SED clock are shown in the table below. When using of the SED, the internal oscillator frequency is specified using the SED_CLK_FREQ parameter.

Table1-2. SED Internal Oscillator Supported Frequency Settings

2.08	4.16	8.31	16.63
2.15	4.29	8.58	17.73
2.22	4.43	8.87	19.00
2.29	4.59	9.17	20.46
2.38	4.75	9.50	22.17
2.46	4.93	9.85	24.18
2.56	5.12	10.23	26.60
2.66	5.32	10.64	29.56
2.77	5.54	11.08	33.25
2.89	5.78	11.57	
3.02	6.05	12.09	
3.17	6.33	12.67	
3.33	6.65	13.30	
3.50	7.00	14.00	
3.69	7.39	14.78	
3.91	7.82	15.65	

SED Attributes

There are three attributes that can be used with the SED primitives and these are shown in Table 1-3. Usage examples for these attributes can be found in the Sample Code section of this document. Currently the SEDFB primitive does not support the three attributes listed.

Table1-3. SED Attributes

Attribute Name	Attribute Type	Description
SED_CLK_FREQ	String	Specifies the clock frequency when used with SEDFA primitive.
DEV_DENSITY	String	Specifies the device density for use by the simulation model.
CHECKALWAYS	String	Reserved for future use.

The SED_CLK_FREQ attribute is used to specify the clock frequency. The SEDFA primitive uses the MachXO2 internal oscillator as the clock source. The available settings are those shown in Table 1-3. If a value other than those shown in the table is used the software will issue an error message and exit from the MAP process.

The DEV_DENSITY attribute is used to specify the device density for the MachXO2 simulation model. If the DEV_DENSITY attribute is not specified the default value of 1200L will be used. The allowable values for the DEV_DENSITY attribute are:

256L, 640L, 1200L, 2000L, 4000L, 7000L, 640U, 1200U, or 2000U

The CHECKALWAYS attribute is not supported at this time.

Port Descriptions

SEDENABLE

SEDENABLE is a level-sensitive signal which enables SED checking when high. When this signal is low, the SED hardware is disabled. This can be tied high in a design if desired.

SEDSTART

SEDSTART is the signal which starts the SED process. The rising edge of the SEDSTART signal will cause the SED cycle to start if SEDENABLE is high. The SEDSTART signal must remain high until the SED process has completed. If SEDSTART goes low during the SED cycle the process will be terminated without asserting SEDDONE or SEDERR.

SEDFRCERR

SEDFRCERR is used to force the SED process to return an error indication on the SEDERR signal. This is typically done to test the logic associated with the SEDERR output. The rising edge of the SEDFRCERR signal is detected by the SED hardware and latched in by the rising edge of the SED Clock Driver signal. The SEDFRCERR should be latched high while the SED is active for an error indication to be returned. The recommended use is for the user logic to drive the SEDFRCERR signal from low to high once the rising edge of the SEDINPROG signal is detected.

SEDSTDBY

The SEDSTDBY port is provided on the SEDFA primitive only and must be connected to the SEDSTDBY output port on OSCH component. This signal is provided for simulation support of the STDBY function which can be used to turn off the internal oscillator. When the STDBY function turns off the internal oscillator the SEDFA block will no longer operate because its clock source has been turned off. If the user does not connect this signal on the SEDFA primitive the SED will still function the same way in the hardware but may not match the simulation results when STDBY is used.

SEDCLKOUT

SEDCLKOUT is a gated version of the SED Clock Driver signal to the SED block. SEDCLKOUT is gated by SEDENABLE. This signal can be used to synchronize the inputs to the SED block or the outputs from the SED block.

SEDDONE

SEDDONE is an output which indicates that SED checking has completed a cycle. This signal is an active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT. SEDDONE will be reset by a low SEDSTART signal.

SEDINPROG

SEDINPROG is an output which indicates that SED checking is in progress. This signal is an active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT. SEDINPROG will go high one clock cycle following SEDSTART going high.

SEDERR

SEDERR is an output which indicates that SED checking has completed a cycle with an error. This signal is an active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT. SEDERR will be reset by a low SEDSTART signal.

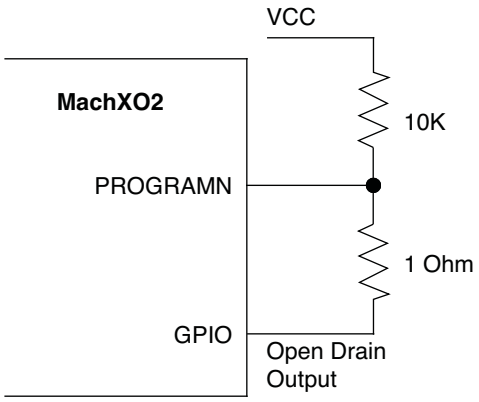
SED Flow

The general SED flow once VCC reaches the data sheet Vcc minimum recommend level is as follows.

1. User logic sets SEDENABLE high. This signal may be tied high if desired.
2. User logic sets SEDSTART high and holds it high for the duration of the SED cycle. SEDINPROG goes high. If SEDDONE or SEDERR are already high they will be driven low.
3. SED starts reading back data from the configuration SRAM.
4. SED finishes checking. SEDERR is updated, SEDINPROG goes low, SEDDONE goes high and another SED cycle is started by asserting SEDSTART. When SEDSTART is asserted the SEDERR signal will be reset.
5. If SEDERR is driven high it can be reset by reconfiguring the PLD.
6. SEDENABLE goes low when/if the user specifies, and SED is no longer in use.

The preferred action to take when an error is detected is to reconfigure the PLD. Reconfiguration can be accomplished by driving the PROGRAMN pin low. This can be done by externally connecting a GPIO pin to PROGRAMN.

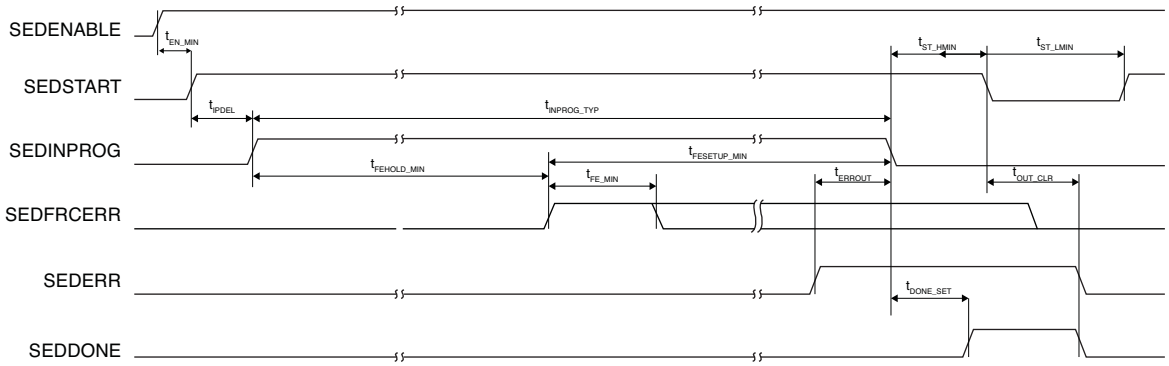
Figure 1-4. Example Schematic



Note: The 1 Ohm resistor shown allows a user to recover from a bad program which always pulls the GPIO pin low in the MachXO2 device. If this type of pattern is loaded into the MachXO2, the device will always be held in the re-configuration state and is not able to communicate or be erased to clear the error condition. To recover from this condition, remove the resistor and reprogram the device, then replace resistor.

Timing Diagram for SED Operation

Figure 1-5. Timing Diagram for SED Operation



Where:

Parameter	Value	Units
t_{EN_MIN}	0	SEDCLK
t_{IPDEL}	2	SEDCLK
t_{FEHOLD_MIN}	92	SEDCLK
$t_{FESETUP\_MIN}$	5	SEDCLK
t_{FE_MIN}	2	SEDCLK
t_{ERROUT}	1	SEDCLK
t_{ST_HMIN}	0	SEDCLK
t_{ST_LMIN}	1	SEDCLK
t_{OUT_CLR}	2	SEDCLK
t_{DONE_SET}	0	SEDCLK

SED Run Time

The amount of time needed to perform an SED check depends on the density of the device and the frequency of the SED Clock Driver signal. There will also be some overhead time for calculation, but it is fairly short in comparison. An approximation of the time required can be found by using the following formula:

(Maxbits / 8) / SED Clock Driver Frequency = Time (ms)

Maxbits is in KBits and depends on the density of the PLD (see Table 1-4). The SED Clock Driver signal frequency is shown in MHz. Time is in milliseconds. The SED checking in the MachXO2 device reads 8 bits (1 byte) on each SED clock cycle.

For example, for a design using a MachXO2 with 4,000 look-up tables and an SED Clock Driver frequency of 7 MHz:

(972 KBits / 8) / 7.0 MHz = 17.4 ms

In this example, SED checking will take approximately 17.4 ms.

Note that the internal oscillator is used to generate the SED Clock Driver signal and its frequency can vary by ±5.5%.

Table1-4. SED Run Time

Density[1]	XO2-256	XO2-640	XO2-640U	XO2-1200	XO2-1200U	XO2-2000	XO2-2000U	XO2-4000	XO2-7000	Units
	94K	191K	360K	360K	535K	535K	972K	972K	1534K	
33.3 MHz	0.35	0.72	1.35	1.35	2.00	2.00	3.64	3.64	5.75	ms
3.05 MHz	3.84	7.82	14.8	14.8	21.9	21.9	39.9	39.9	62.9	ms

1. Density refers to the number of configuration bits in the device.

Sample Code

The following simple example code shows how to instantiate the SED primitive. In the example the SED is always enabled and will be run when the “sed_start” signal is driven high. The outputs of the SED hardware are available to route to the PLD output pins or for use in another module. Note that the SADFA primitive is part of ispLEVER 8.1 SP1 or later and Diamond 1.1 or later.

SED VHDL Examples

VHDL SEDFA

```
COMPONENT SEDFA
   GENERIC(
      SED_CLK_FREQ  : string  := "3.5";
      CHECKALWAYS   : string  := "DISABLED";
      DEV_DENSITY   : string  := "1200L");
--"256L","640L","1200L","2000L","4000L","7000L", "640U", "1200U", "2000U"

PORT(
          SEDENABLE    :       in      STD_LOGIC;
          SEDSTART     :       in      STD_LOGIC;
          SEDFRCERR    :       in      STD_LOGIC;
          SEDSTDBY     :       in      STD_LOGIC;
          SEDERR       :       out     STD_LOGIC;
          SEDDONE      :       out     STD_LOGIC;
          SEDINPROG    :       out     STD_LOGIC;
          SEDCLKOUT    :       out     STD_LOGIC);
END COMPONENT;

attribute SED_CLK_FREQ : string ;
attribute SED_CLK_FREQ of SEDinst0 : label is "13.30";
attribute CHECKALWAYS : string ;
attribute CHECKALWAYS of SEDinst0 : label is "DISABLED" ;
attribute DEV_DENSITY : string ;
attribute DEV_DENSITY of SEDinst0 : label is "1200L" ;


SEDinst0:  SEDFA
-- synthesis translate_off
      GENERIC MAP ( SED_CLK_FREQ => “13.30”;
           CHECKALWAYS => "DISABLED" ;
           DEV_DENSITY => "1200L" )
-- synthesis translate_on
PORT MAP  (SEDENABLE => '1',
        SEDSTART => sed_start,
        SEDFRCERR => '0',
```

```
        SEDSTDBY => sed_stdby,
        SEDERR => sed_err,
        SEDDONE => sed_done,
        SEDINPROG => sed_active,
        SEDCLKOUT => sed_clkout);
```

SED Verilog Examples

Verilog SEDFA

```
module SEDFA (SEDENABLE, SEDSTART, SEDFRCERR, SEDSTDBY, SEDERR, SEDDONE, SEDINPROG,
SEDCLKOUT);

input   SEDENABLE, SEDSTART, SEDFRCERR, SEDSTDBY;
output  SEDERR, SEDDONE, SEDINPROG, SEDCLKOUT;

parameter SED_CLK_FREQ = "3.5";
parameter CHECKALWAYS = "DISABLED";
parameter DEV_DENSITY = "1200L";
//"256L","640L","1200L","2000L","4000L","7000L","640U", "1200U", and "2000U"


endmodule
```

Verilog SEDFA Primitive Instantiation

```
//  instantiate SEDFA primitive module with parameter passing to SEDFA module
SEDFA # (.SED_CLK_FREQ("4.75"), .DEV_DENSITY("1200L"))

sedfa_tst (.SEDENABLE(1'b1), .SEDSTART(sed_start), .SEDFRCERR(1'b0), .SEDSTDBY(),
          .SEDERR(sed_err),  .SEDDONE(sed_done),  .SEDINPROG(sed_active),  .SEDCLK-
OUT(sed_clkout) );
```

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.
January 2011	01.1	Updated for ultra-high I/O (“U”) devices.
May 2011	01.2	Changed “SED” to “SRAM CRC Error Detection” throughout the document.
		Added Limitations section.
		Replaced Basic SED Mode and Hardware Description sections with Operating Modes section and added description of the primitives.
		Updated supported frequencies in Table 17-2, SRAM CRC Error Detection Internal Oscillator Supported Frequency Settings.
		Updated Port descriptions and Flow sections.
		Corrected Run Time calculations and updated Table 17-4, SRAM CRC Error Detection Run Time.
		Updated Sample Code section.
June 2011	01.3	Document title changed from “MachXO2 Soft Error Detection (SED) Usage Guide” to “MachXO2 SRAM CRC Error Detection Usage Guide”.
		Clarified attribute usage for SEDFB primitive.
		Clarified run time calculation section.
July 2011	01.4	Updated Standard SRAM CRC Error Detection text section with information about migration from MachXO2-1200-R1 to Standard (N-1) devices.
February 2012	01.5	Updated document with new corporate logo.
August 2012	01.6	Updated SRAM CRC Error Detection Limitations and SRAM CRC Error Detection Operating Modes sections.
January 2013	01.7	Defined the EFB or Configuration Logic which interacts with the SRAM CRC Error Detection circuit.
		Added timing diagram for SED operation.
		Changed “SRAM CRC Error Detection” to “SED” throughout the document.
February 2013	01.8	Removed requirements for holding user logic in a steady state.

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

April 2013 Technical Note TN1246

Introduction

This reference guide supplements TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2™ Devices Usage Guide which explains the software usage. In this document you will find:

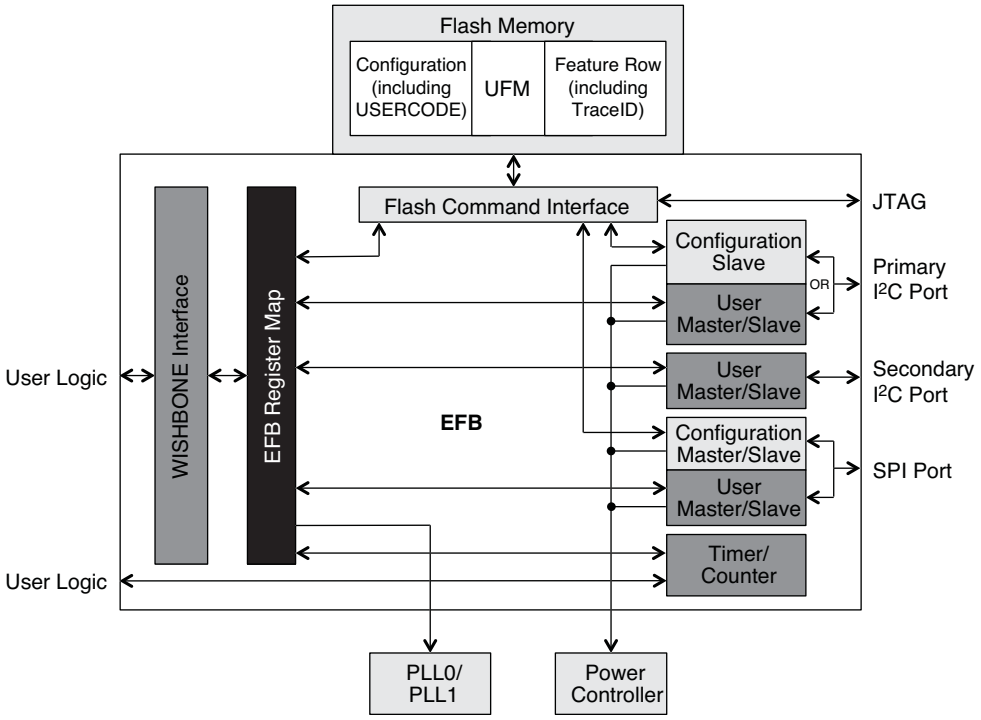
- WISHBONE Protocol
- EFB Register Map
- Command Sequences
- Examples

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 I^2C 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 I^2C and SPI

Figure 17-1 shows the EFB architecture and the interface to the FPGA core logic.

Figure 17-1. Embedded Function Block (EFB)



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 17-1 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 17-1. EFB Register Map

Address (Hex)	Hardened Function
0x00-0x1F	PLL0 Dynamic Access1
0x20-0x3F	PLL1 Dynamic Access1
0x40-0x49	I^2C Primary
0x4A-0x53	I^2C Secondary
0x54-0x5D	SPI
0x5E-0x6F	Timer/Counter
0x70-0x75	Flash Memory (UFM/Configuration)
0x76-0x77	EFB Interrupt Source

1. 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 TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide, for details on PLL configuration registers and recommended usage.

Address spaces that are not defined in Table 17-1 are invalid and will result in non-deterministic results. It is the responsibility of the designer to ensure valid addresses are presented to the EFB WISHBONE slave interface.

WISBONE 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 17-2 shows the supported WISHBONE bus signals between the FPGA core and the EFB. Table 17-2 provides a detailed definition of the supported signals.

Figure 17-2. WISHBONE Bus Interface Between the FPGA Core and the EFB Module

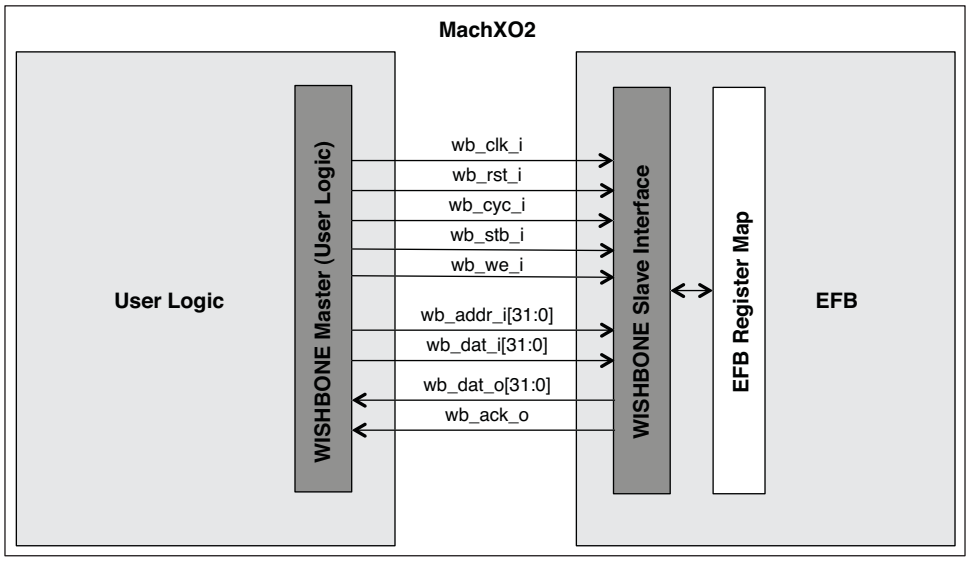


Table 17-2. WISHBONE Slave Interface Signals of the EFB Module

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.
wb_rst_i	Input	1	Active-high, synchronous reset signal that will only reset the WISHBONE interface logic. This signal will not affect the contents of any registers. It will only affect ongoing bus transactions. Wait 1us 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 will simply respond by asserting wb_ack_o signal. It is the responsibility of the user to stay within the valid address range.

WISHBONE Write Cycle

Figure 17-3 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 that will be written to the EFB block, wb_dat_i[7:0] data lines
- Asserts the write enable wb_we_i signal, indicating a write cycle
- Asserts the wb_cyc_i to indicate the start of the cycle
- Asserts the wb_stb_i, selecting a specific slave module

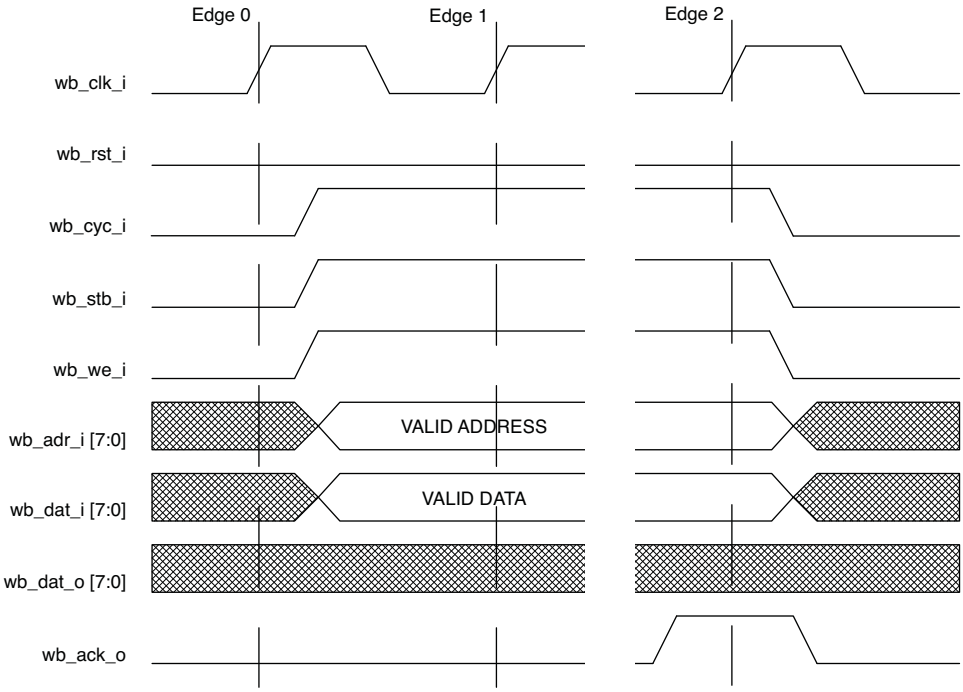
On clock Edge 1, the EFB WISHBONE Slave decodes the input signals presented by the master. During this cycle:

- 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 17-3. WISHBONE Bus Write Operation



WISHBONE Read Cycle

Figure 17-4 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 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

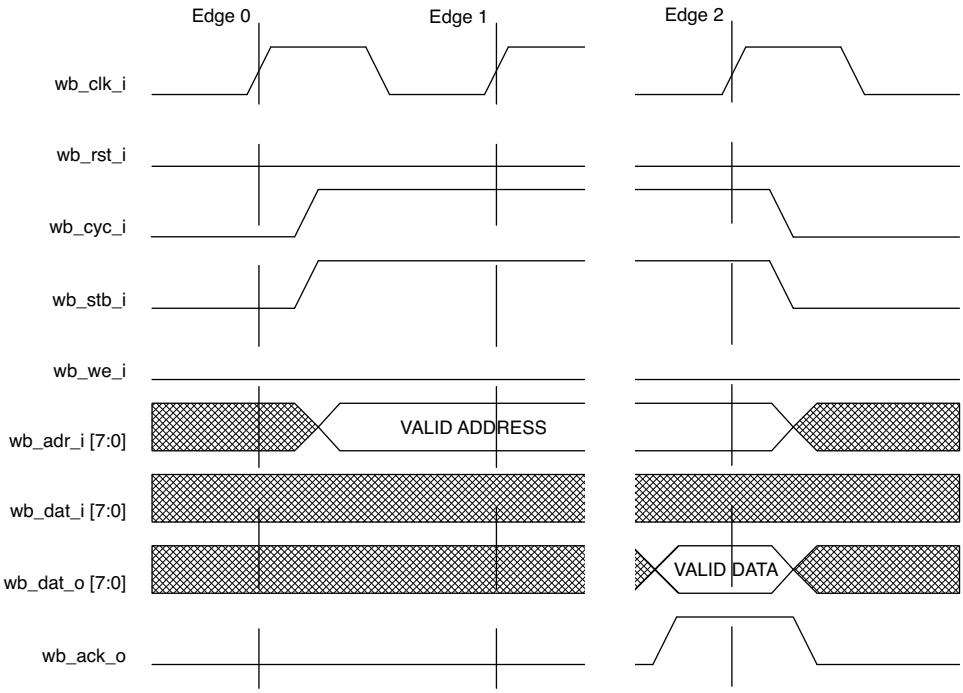
On clock Edge 1, the EFB WISHBONE slave decodes the input signals presented by the master. The following occurs during this cycle:

- 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

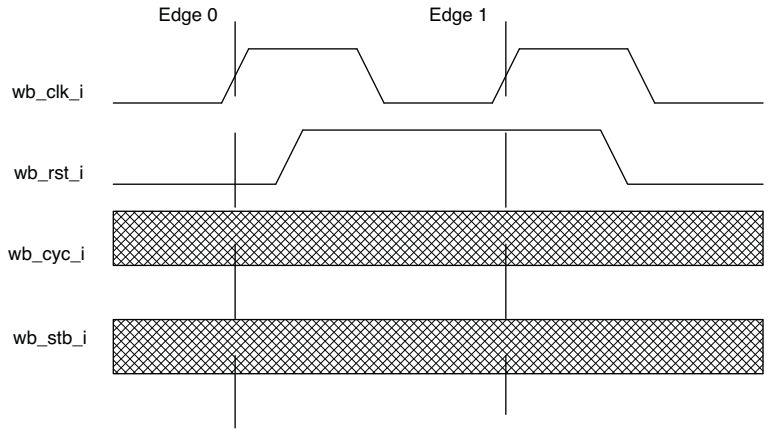
Figure 17-4. WISHBONE Bus Read Operation



WISHBONE Reset Cycle

Figure 17-5 shows the waveform of the synchronous wb_rst_i signal. Asserting the reset signal will only reset the WISHBONE interface logic. This signal will not affect the contents of any registers in the EFB register map. It will only affect ongoing bus transactions.

Figure 17-5. EFB WISHBONE Interface Reset



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

Hardened I²C IP Cores

I²C is a widely used two-wire serial bus for communication between devices on the same board. Every MachXO2 device contains two hardened I²C IP cores designated as the "Primary" and "Secondary" I²C IP cores. Either of the two cores can be operated as an I²C Master or as an I²C Slave. The difference between the two cores is that the Primary core has pre-assigned I/O pins while the ports of the secondary core can be assigned by designers to any general purpose I/O. In addition, the Primary I²C core can be used for accessing the User Flash Memory (UFM) and for programming the Configuration Flash. However, the Primary I²C port cannot be used for both UFM/Config access and user functions in the same design.

I²C Registers

Both I²C cores communicate with the EFB WISHBONE interface through a set of control, command, status and data registers. Table 17-3 shows the register names and their functions. These registers are a subset of the EFB register map.

Table 17-3. I²C Registers

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

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

Table 17-4. I²C Control (Primary/Secondary)

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

Note: A write to this register will cause the I²C 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 Figure 17-14)

- 00: 300ns
- 01: 150ns
- 10: 75ns
- 11: 0ns

Table 17-5. I²C Command (Pri/Sec)

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

STA — Generate START (or Repeated START) condition (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. The I²C cores support a “wait state” or clock stretching from the slave, meaning the slave can enforce a wait state if it needs time to finish the task. Bit CKSDIS disables the clock stretching if desired by the user. In this case, the overflow flag must be monitored. For Master operations, set this bit to ‘0’. Clock stretching will be used by the MachXO2 EFB I²C Slave during both ‘read’ and ‘write’ operations (from the Master perspective) when I²C Command Register bit CKSDIS=0.

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

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

- 0: Enabled
- 1: Disabled

Table 17-6. I²C Clock Prescale 0 (Primary/Secondary)

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

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

Table 17-7. I²C 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[1]	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	—	R/W	R/W

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

I2C_PRESCALE[9:0] — I²C Clock Prescale value. A write operation to I2CBR [9:8] will cause an I²C core reset. The WISHBONE clock frequency is divided by (I2C_PRESCALE*4) to produce the Master I²C clock frequency supported by the I²C bus (50KHz, 100KHz, 400KHz).

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 I²C core is configured in the I²C tab of the EFB GUI. The calculation is based on the WISHBONE Clock Frequency and the I²C Frequency, both entered by the user. The digital value of the divider is programmed in the 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 I²C bus frequencies.

Table 17-8. I²C Transmit Data Register (Primary/Secondary)

I2C_1_TXDR / I2C_2_TXDR								0x44/0x4E
Bit	7	6	5	4	3	2	1	0
Name	I2C_Transmit_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] — I²C Transmit Data. This register holds the byte that will be transmitted on the I²C bus during the Write Data phase. Bit 0 is the LSB and will be transmitted last. When transmitting the slave address, Bit 0 represents the Read/Write bit.

Table 17-9. I²C Status (Primary/Secondary)

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

1. These bits exhibit 0.5 SCK period latency before valid in R1 devices. For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

TIP — Transmit In Progress. The current data byte is being transferred. Note that the TIP flag will suffer one-half SCL cycle latency right after the START condition because of the signal synchronization. Also note that this bit could be high after configuration wake-up and before the first valid I²C transfer start (when BUSY is low), and it is not indicating byte in transfer, but an invalid indicator.

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

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

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

RARC — Received Acknowledge. An acknowledge response from the addressed slave (during master write) or from receiving master (during master read) was received.

- 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

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 of Receiver is not ready

TROE — Transmitter/Receiver Overrun Error or NACK received. A transmit or receive overrun error has occurred (dependent upon master/slave mode and SRW status), or a No Acknowledge was received (only when RARC also set). This bit is capable of generating an interrupt.

- 1: Transmitter or Receiver Overrun detected or NACK received
- 0: Normal

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

1: General Call Received in slave mode
0: Normal

Figure 17-6. I²C General Call Data Register (Primary/Secondary)

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

I2C_ GC _Data[7:0] — I²C General Call Data. This register holds the second (command) byte of the General Call transaction on the I²C bus.

Table 17-10. I²C Receive Data Register (Primary/Secondary)

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

I2C_ Receive _Data[7:0] — I²C Receive Data. This register holds the byte captured from the I²C bus during the Read Data phase. Bit 0 is LSB and was received last.

Table 17-11. I²C Interrupt Status (Primary/Secondary)

I2C_1_IRQ / I2C_2_ IRQ								0x48/0x52
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)				IRQARBL	IRQTRRDY	IRQTROE	IRQHGC
Default	—	—	—	—	—	—	—	—
Access	—	—	—	—	R/W	R/W	R/W	R/W

IRQARBL — Interrupt Status for Arbitration Lost.
When enabled, indicates ARBL was asserted. Write a '1' to this bit to clear the interrupt.
1: Arbitration Lost Interrupt
0: No interrupt

IRQTRRDY — Interrupt Status for Transmitter or Receiver Ready.
When enabled, indicates TRRDY was asserted. Write a '1' to this bit to clear the interrupt.
1: Transmitter or Receiver Ready Interrupt
0: No interrupt

IRQTROE — Interrupt Status for Transmitter/Receiver Overrun or NACK received.
When enabled, indicates TROE was asserted. Write a '1' to this bit to clear the interrupt.
1: Transmitter or Receiver Overrun or NACK received Interrupt
0: No interrupt

IRQHGC — Interrupt Status for Hardware General Call Received.
When enabled, indicates HGC was asserted. Write a '1' to this bit to clear the interrupt.

1: General Call Received in slave mode Interrupt
0: No interrupt

Table 17-12. I²C Interrupt Enable (Primary/Secondary)

I2C_1_ IRQEN / I2C_2_IRQEN								0x49/0x53
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)				IRQARBLEN	IRQTRRDYEN	IRQTROEEN	IRQHGCEN
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	R/W	R/W	R/W	R/W

IRQARBLEN — Interrupt Enable for Arbitration Lost
1: Interrupt generation enabled
0: Interrupt generation disabled

IRQTRRDYEN — Interrupt Enable for Transmitter or Receiver Ready
1: Interrupt generation enabled
0: Interrupt generation disabled

IRQTROEEN — Interrupt Enable for Transmitter/Receiver Overrun or NACK Received
1: Interrupt generation enabled
0: Interrupt generation disabled

IRQHGCEN — Interrupt Enable for Hardware General Call Received
1: Interrupt generation enabled
0: Interrupt generation disabled

Figure 17-7 shows a flow diagram for controlling Master I²C reads and writes initiated via the WISHBONE interface. The following sequence is for the Primary I²C but the same sequence applies to the Secondary I²C.

Figure 17-7. I²C Master Read/Write Example (via WISHBONE)

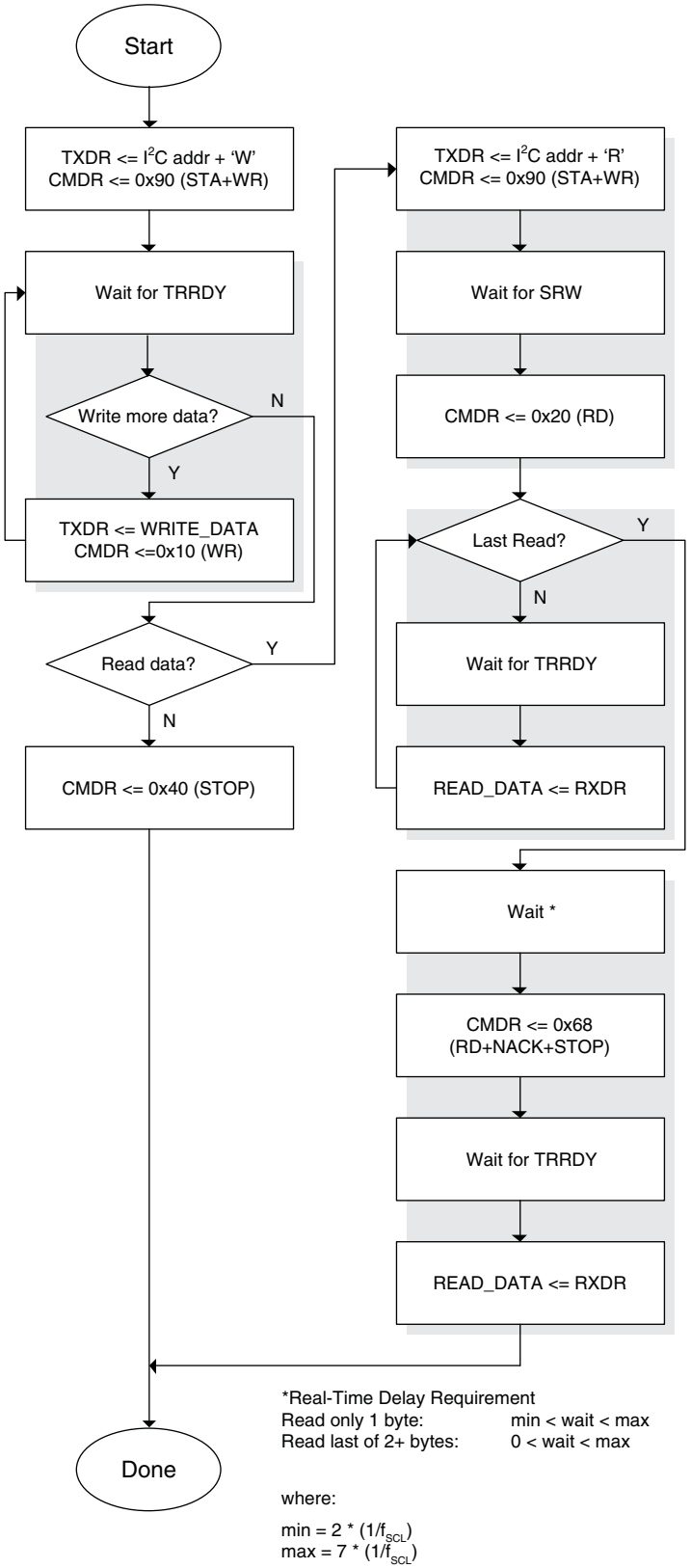
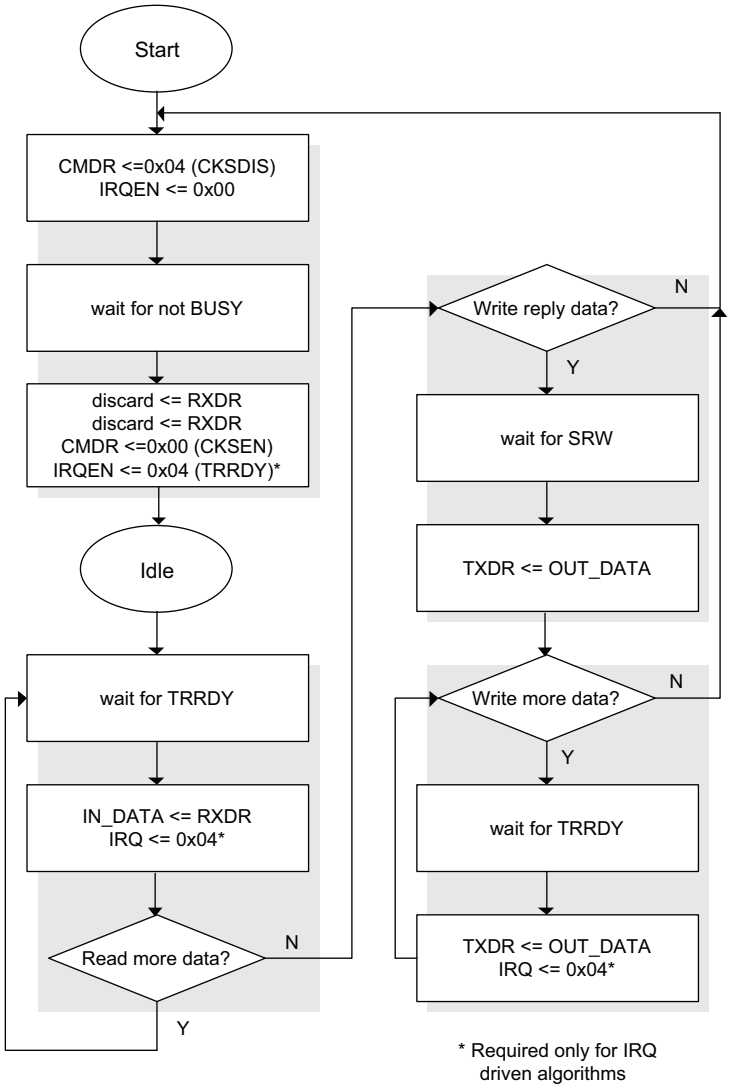


Figure 17-8 shows a flow diagram for reading and writing from an I²C Slave device via the WISHBONE interface. The following sequence is for the Primary I²C but the same sequence applies to the Secondary I²C.

Figure 17-8. I²C Slave Read/Write Example (via WISHBONE)



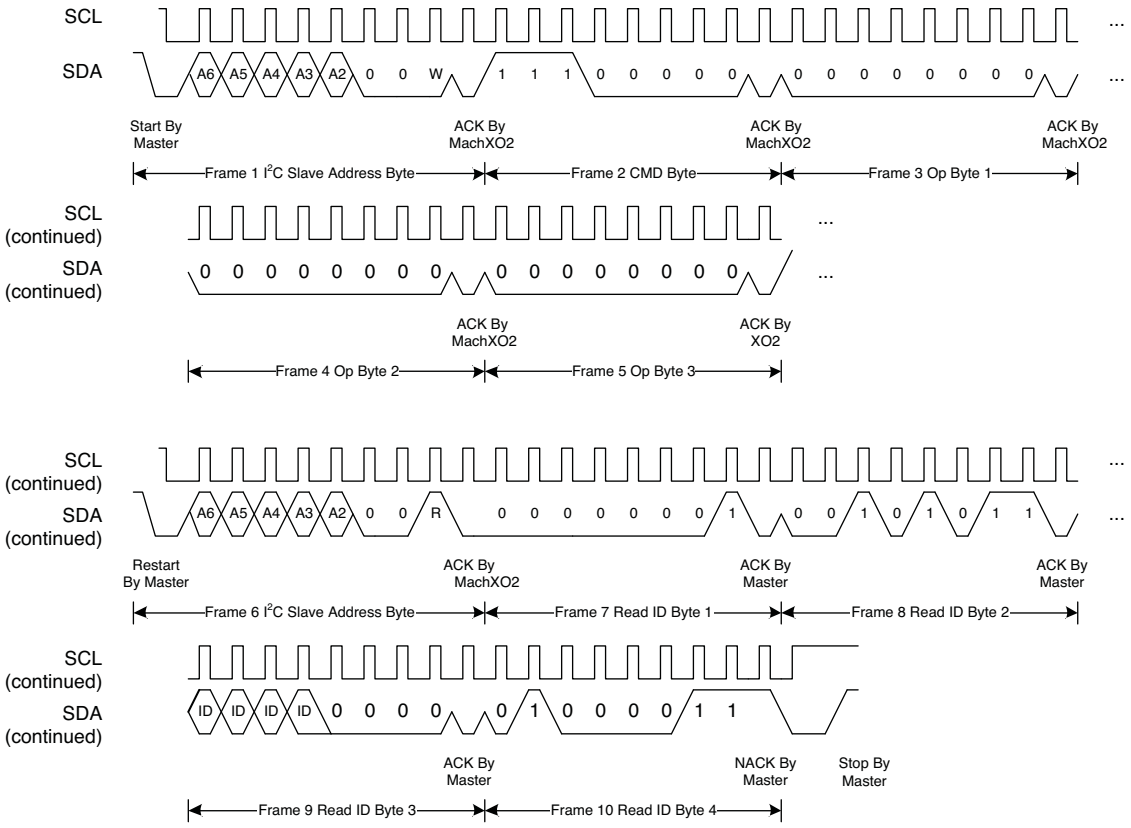
I²C Framing

Each command string sent to the I²C EFB port must be correctly "framed" using the protocol defined for each interface. In the case of I²C, the protocol is well known and defined by the industry as shown below.

Table 17-13. Command Framing Protocol, by Interface

Interface	Pre-op (+)	Command String	Post-op (-)
I²C	Start	(Command/Operands/Data)	Stop

Figure 17-9. I²C Read Device ID Example



I²C Functional Waveforms

Figure 17-10. EFB Master – I²C Write

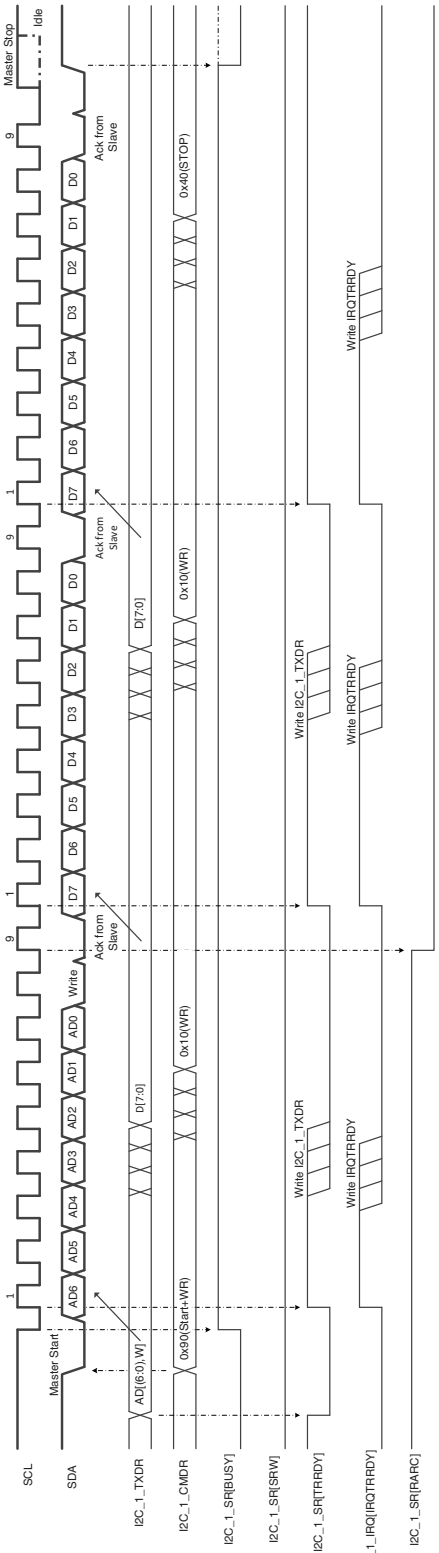


Figure 17-11. EFB Master – I²C Read

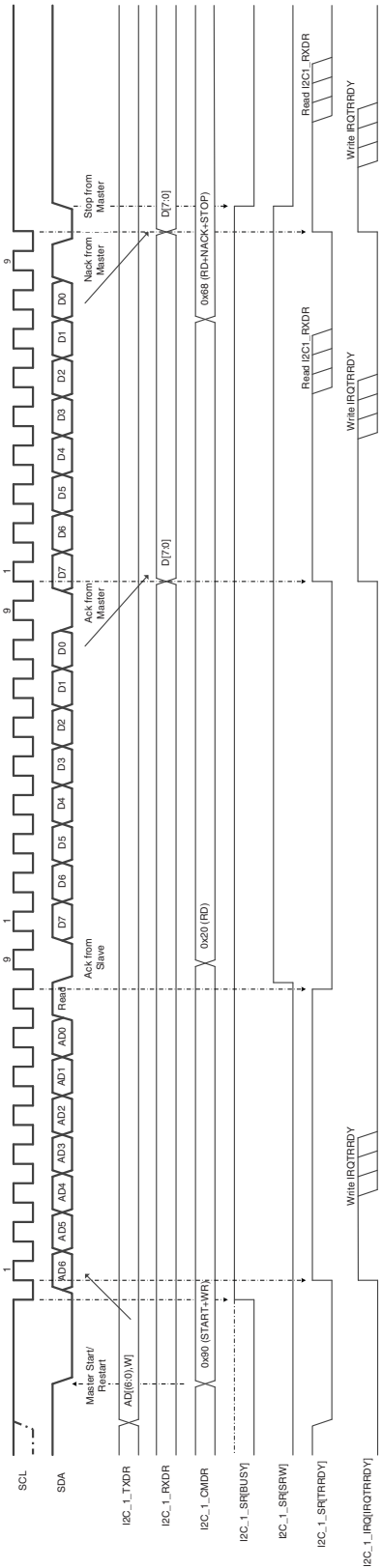


Figure 17-12. EFB Slave – I²C Write

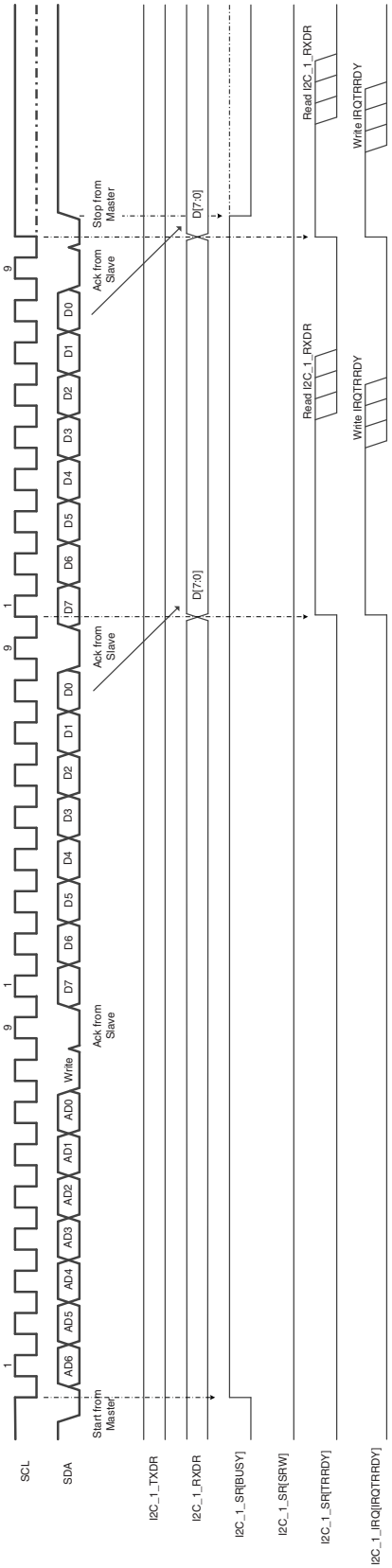
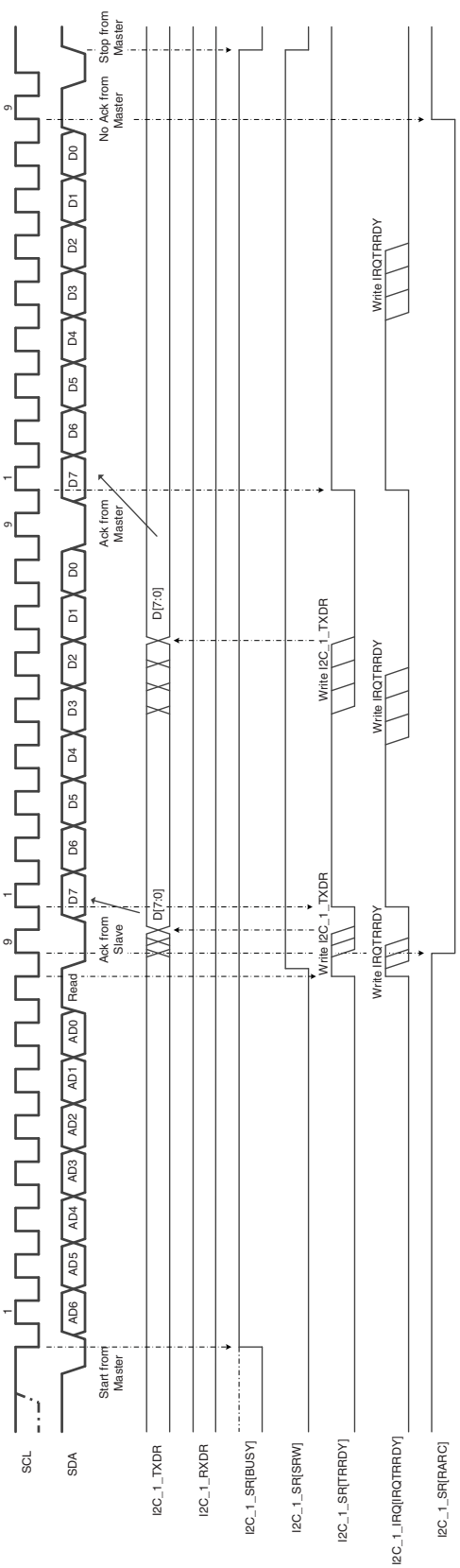
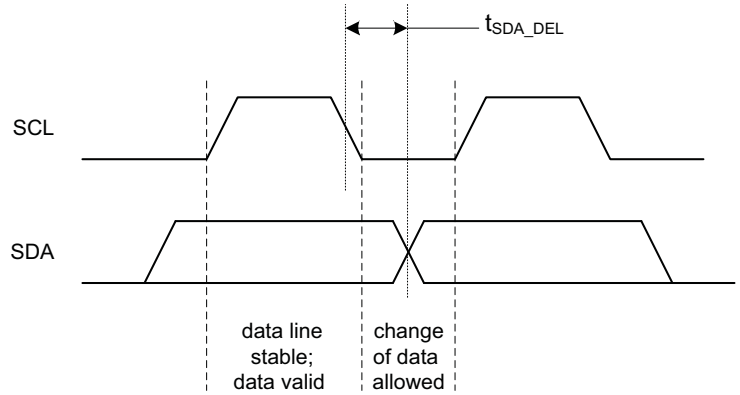


Figure 17-13. EFB Slave – I²C Read



I²C Timing Diagram

Figure 17-14. I²C Bit Transfer Timing



I²C Simulation Model

The I²C EFB Register Map translation to the MachXO2 EFB software simulation model is provided in below.

Table 17-14. I²C Primary Simulation Mode

I²C Primary Register Name	Register Size/Bit Location	Register Function	Address I²C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_1_CR	[7:0]	Control	0x40	Read/Write	i2ccr1[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2CEN	7				i2c_en	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
GCEN	6				i2c_gcen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
WKUPEN	5				i2c_wkupen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
SDA_DEL_SEL[1:0]	[3:2]				sda_del_sel	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_1_CMDR	[7:0]	Command	0x41	Read/Write	i2ccmdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
STA	7				i2c_sta	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
STO	6				i2c_sto	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
RD	5				i2c_rd	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
WR	4				i2c_wt	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
ACK	3				i2c_nack	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
CKSDIS	2				i2c_cksdis	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_1_BR0	[7:0]	Clock Pre-scale	0x42	Read/Write	i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_PRESCALE[7:0]	[7:0]				i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_1_BR1	[7:0]	Clock Pre-scale	0x43	Read/Write	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_PRESCALE[9:8]	[1:0]				i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_1_TXDR	[7:0]	Transmit Data	0x44	Write	i2ctxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_Transmit_Data[7:0]	[7:0]				i2ctxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_1_SR	[7:0]	Status	0x45	Read	i2csr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/

Table 17-14. I²C Primary Simulation Mode (Continued)

I²C Primary Register Name	Register Size/Bit Location	Register Function	Address I²C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
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/
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/
IRQARBLEN	3				i2csr_1st_irqena_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDYEN	2				i2csr_1st_irqena_2	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTROEEN	1				i2csr_1st_irqena_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQHGCEN	0				i2csr_1st_irqena_0	../efb_top/efb_pll_sci_inst/u_efb_sci/

Table 17-15. I²C Secondary Simulation Model

I²C Secondary Register Name	Register Size/Bit Location	Register Function	Address I²C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_2_CR	[7:0]	Control	0x4A	Read/Write	i2ccr1[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2CEN	7				i2c_en	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
GCEN	6				i2c_gcen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
WKUPEN	5				i2c_wkupen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
SDA_DEL_SEL[1:0]	[3:2]				sda_del_sel	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_CMDR	[7:0]	Command	0x4B	Read/Write	i2ccmdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
STA	7				i2c_sta	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/

Table 17-15. I²C Secondary Simulation Model (Continued)

I²C Secondary Register Name	Register Size/Bit Location	Register Function	Address I²C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
STO	6				i2c_sto	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
RD	5				i2c_rd	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
WR	4				i2c_wt	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
ACK	3				i2c_nack	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
CKSDIS	2				i2c_cksdis	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_BR0	[7:0]	Clock Pre-scale	0x4C	Read/Write	i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_PRESCALE[7:0]	[7:0]				i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_BR1	[7:0]	Clock Pre-scale	0x4D	Read/Write	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_PRESCALE[9:8]	[1:0]				i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/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_irqsts_3, i2csr_2nd_irqsts_2, i2csr_2nd_irqsts_1, i2csr_2nd_irqsts_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQARBL	3				i2csr_2nd_irqsts_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDY	2				i2csr_2nd_irqsts_2	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTROE	1				i2csr_2nd_irqsts_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQHGC	0				i2csr_2nd_irqsts_0	../efb_top/efb_pll_sci_inst/u_efb_sci/
I2C_2_IRQEN	[7:0]	IRQ Enable	0x53	Read/Write	{1'b0, 1'b0, 1'b0, 1'b0, i2csr_2nd_irqena_3, i2csr_2nd_irqena_2, i2csr_2nd_irqena_1, i2csr_2nd_irqena_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQARBLEN	3				i2csr_2nd_irqena_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDYEN	2				i2csr_2nd_irqena_2	../efb_top/efb_pll_sci_inst/u_efb_sci/

Table 17-15. I²C Secondary Simulation Model (Continued)

I²C Secondary Register Name	Register Size/Bit Location	Register Function	Address I²C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
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/

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.

SPI Registers

The SPI core communicates with the WISHBONE interface through a set of control, command, status and data registers. Table 17-16 shows the register names and their functions. These registers are a subset of the EFB register map.

Table 17-16. 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 17-17. 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 will cause 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 17-18. SPI Control 1

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

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

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

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

WKUPEN_USER Wake-up Enable via User. Enables the SPI core to send a wake-up signal to the on-chip Power Controller to wake the part from Standby mode when the User 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 Figures 17-25 through 17-28.

0: Transmit data on the MCLK edge defined by SPICR2[CPOL] and SPICR2[CPHA]
1: Transmit data ½ MCLK earlier than defined by SPICR2[CPOL] and SPICR2[CPHA]

Table 17-19. 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 will cause 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.

Note: 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 AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

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 (e.g. 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 will be 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 will be 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 Figure 17-19.

0: Normal Slave SPI operation
1: Lattice proprietary Slave Dummy Byte Response Enabled

Note: 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 — 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 will abort a transmission in progress and force the SPI system into idle state. Refer to Figures 17-25 through 17-28.

0: Active-high clocks selected. In idle state SCK is low.
1: Active-low clocks selected. In idle state SCK is high.

CPHA — SPI Clock Phase. Selects the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. Refer to Refer to Figures 17-25 through 17-28.

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

Note: When CPHA=1, the user must explicitly place a pull-up or pull-down on SCK pad corresponding to the value of CPOL (e.g. 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.

LSBF — LSB-First. LSB appears first on the SPI interface. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. Refer to

Figures 17-25 through 17-28.

Note: This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7.

0: Data is transferred most significant bit (MSB) first
1: Data is transferred least significant bit (LSB) first

Table 17-20. SPI Clock Prescale

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

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 will cause a SPI core reset. DIVIDER must be >= 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 17-21. 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 will cause the SPI core to reset.

Table 17-22. 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 that will be transmitted on the SPI bus. Bit 0 in this register is LSB, and will be 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 17-25. This timing requires at least one protocol dummy byte be included for all slave SPI read operations.

Table 17-23. 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 AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

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 was 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) was 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 17-24. 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 was received last when LSBF=0 or first when LSBF=1.

Table 17-25. 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] was 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] was asserted. Write a '1' to this bit to clear the interrupt.
1: SPI Receive Ready Interrupt
0: No interrupt

IRQROE Interrupt Status for Receive Overrun Error.
When enabled, indicates ROE was asserted. Write a '1' to this bit to clear the interrupt.
1: Receive Overrun Error Interrupt
0: No interrupt

IRQMDF Interrupt Status for Mode Fault.
When enabled, indicates MDF was asserted. Write a '1' to this bit to clear the interrupt.
1: Mode Fault Interrupt
0: No interrupt

Table 17-26. 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 Interrupt Enable for SPI Transmit Ready.
1: Interrupt generation enabled
0: Interrupt generation disabled

IRQRRDYEN	Interrupt Enable for SPI Receive Ready 1: Interrupt generation enabled 0: Interrupt generation disabled
IRQROEEN	Interrupt Enable for Receive Overrun Error 1: Interrupt generation enabled 0: Interrupt generation disabled
IRQMDFEN	Interrupt Enable for Mode Fault 1: Interrupt generation enabled 0: Interrupt generation disabled

Figure 17-15 shows a flow diagram for controlling Master SPI reads and writes initiated via the WISHBONE interface.

Figure 17-15. SPI Master Read/Write Example (via WISHBONE) – Production Silicon

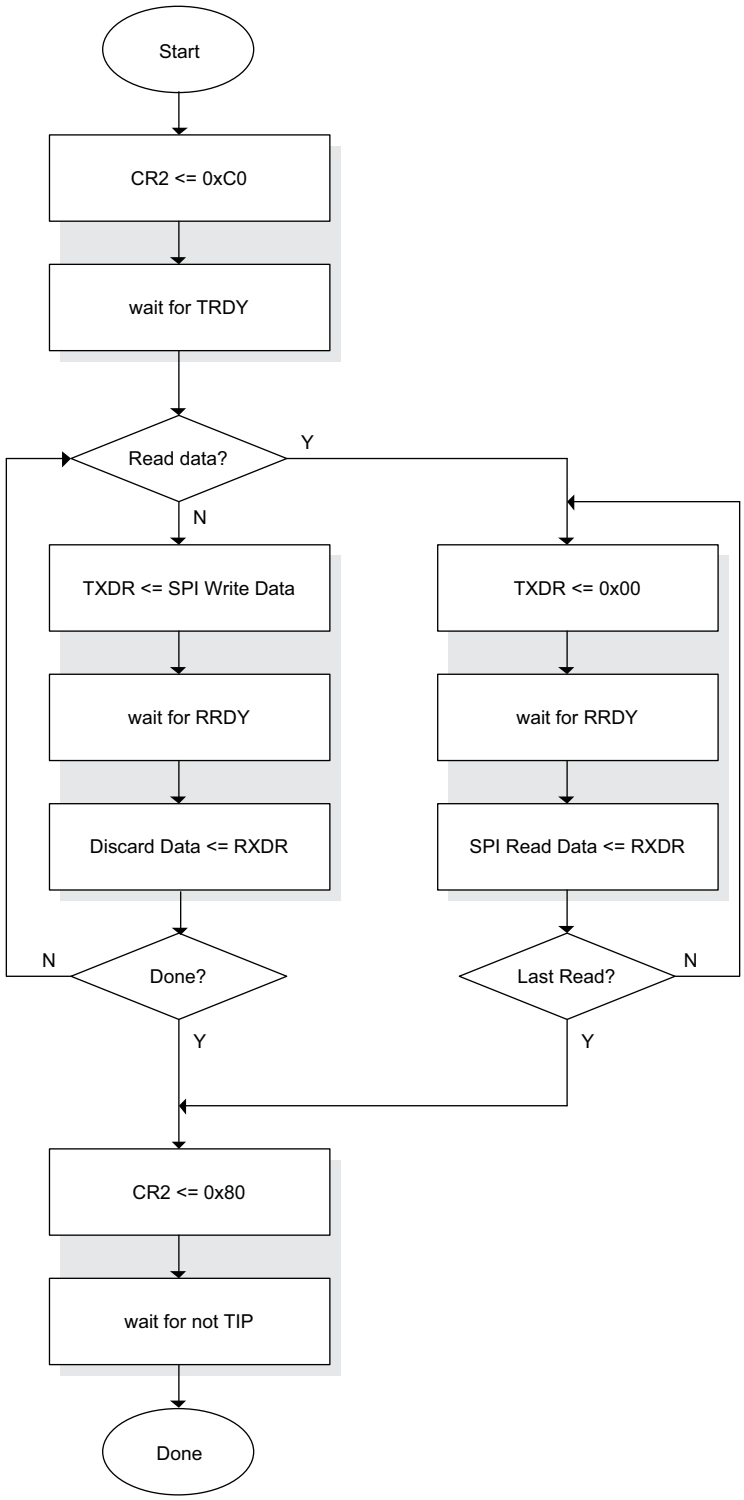
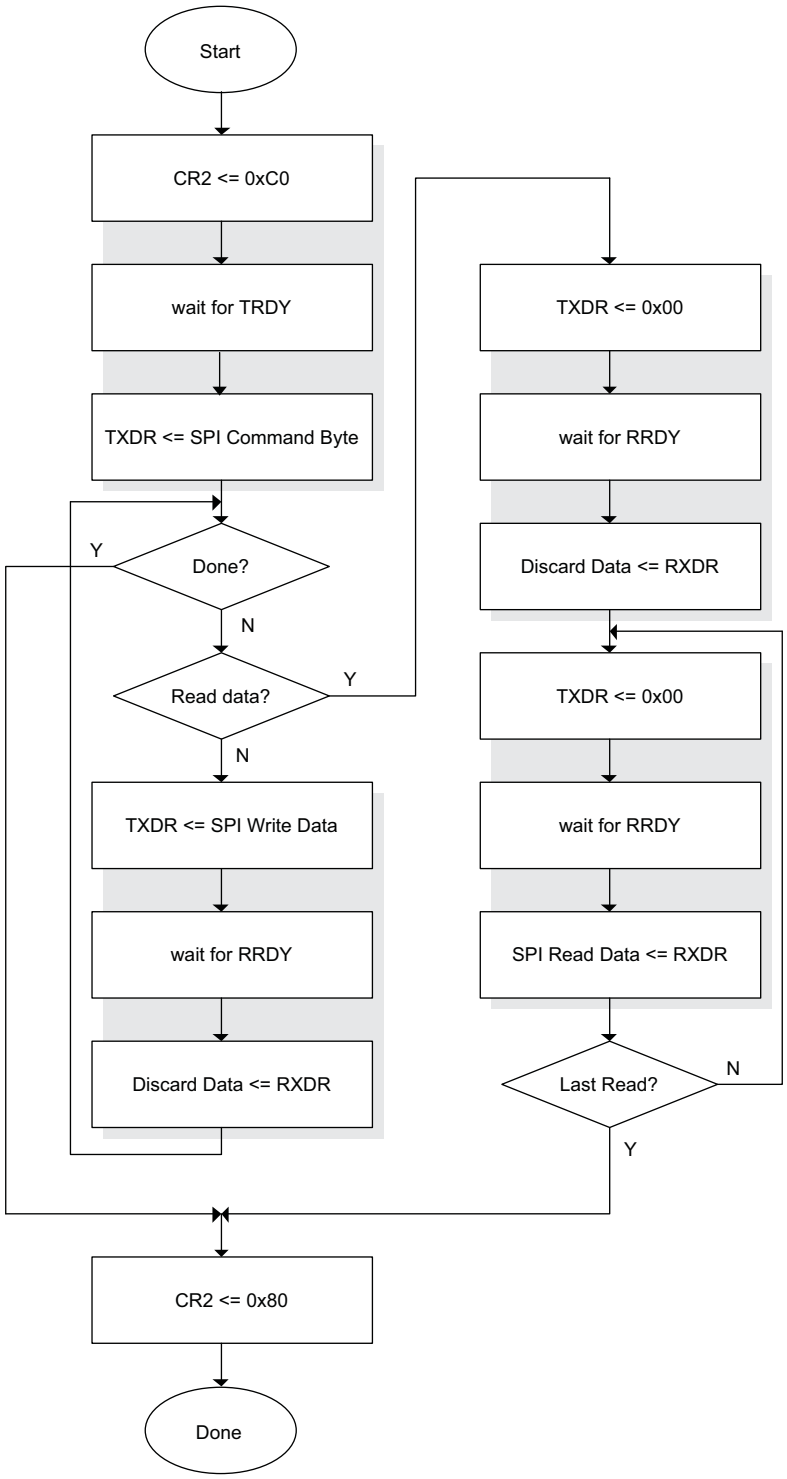


Figure 17-16. SPI Master Read/Write Example (via WISHBONE) – R1 Silicon



Note: For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

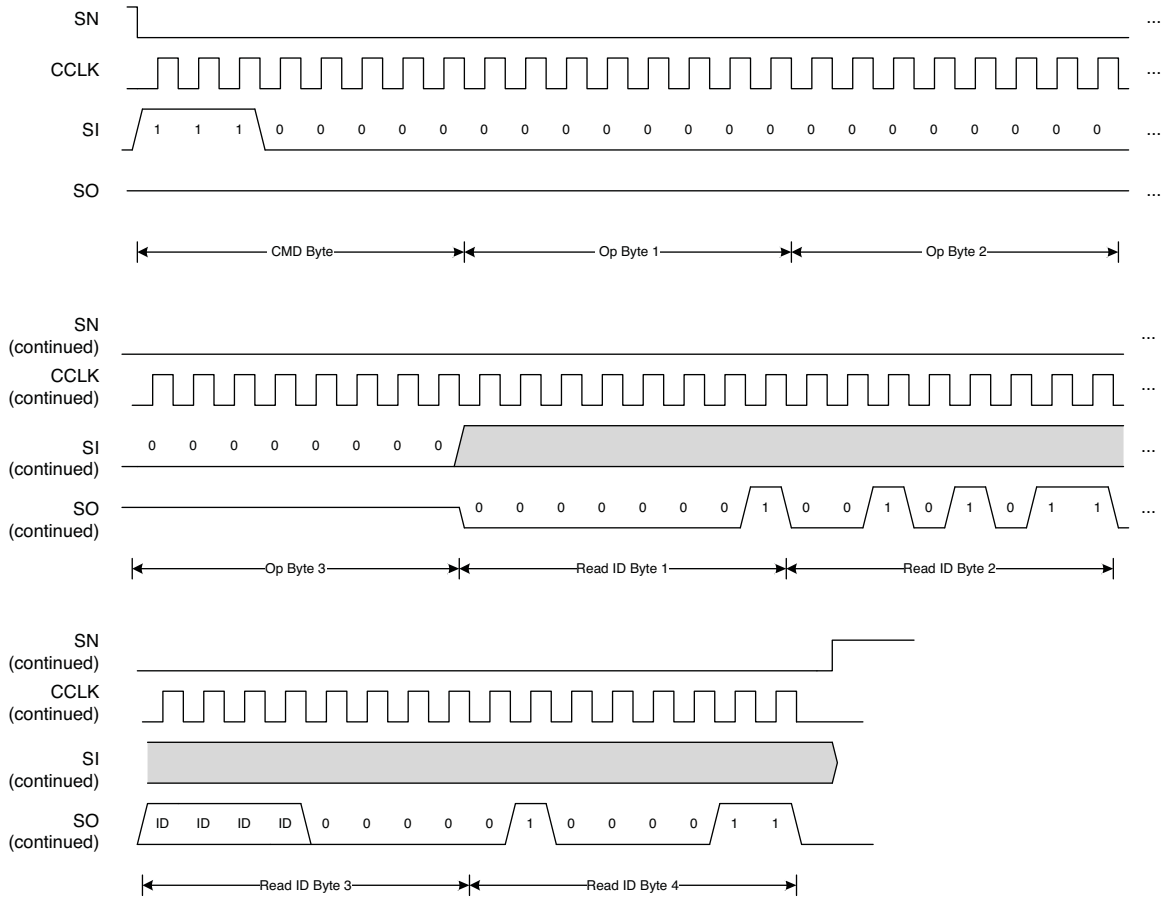
SPI Framing

Each command string sent to the SPI EFB port must be correctly 'framed' using the protocol defined for each interface. In the case of SSPI the protocol is well known and defined by the industry as shown below:

Table 17-27. Command Framing Protocol, by Interface

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

Figure 17-17. SSPI Read Device ID Example



SPI Functional Waveforms

Figure 17-18. Fully Specified SPI Transaction

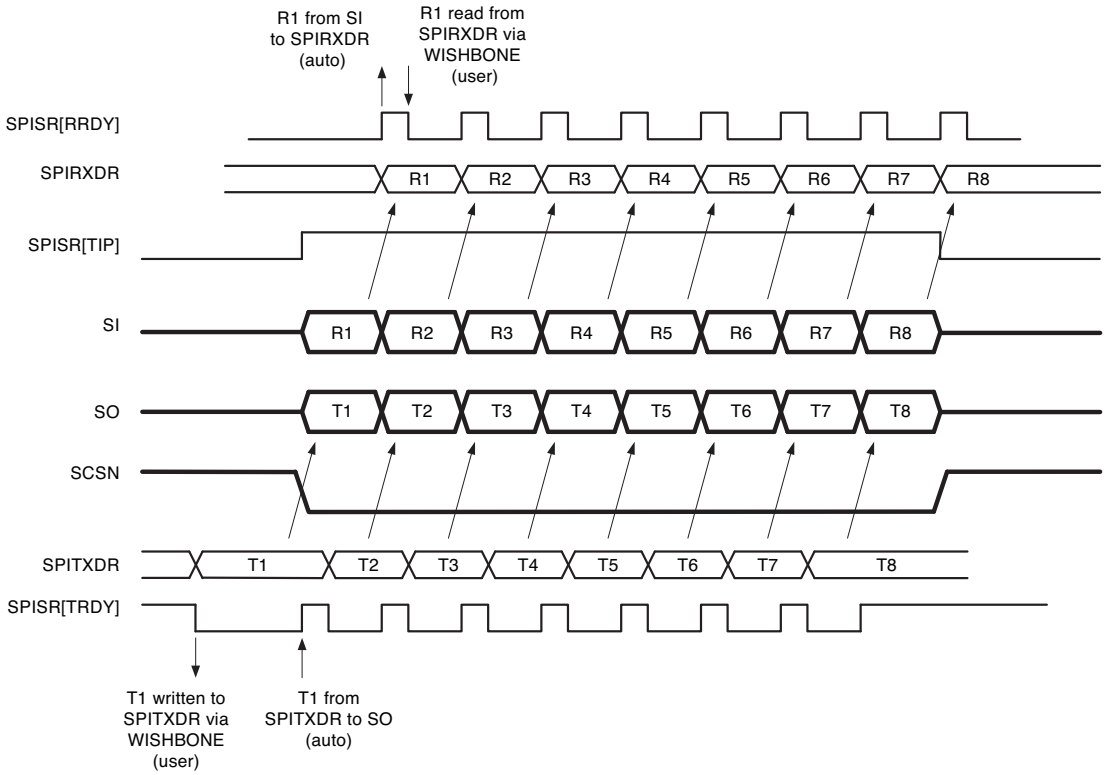
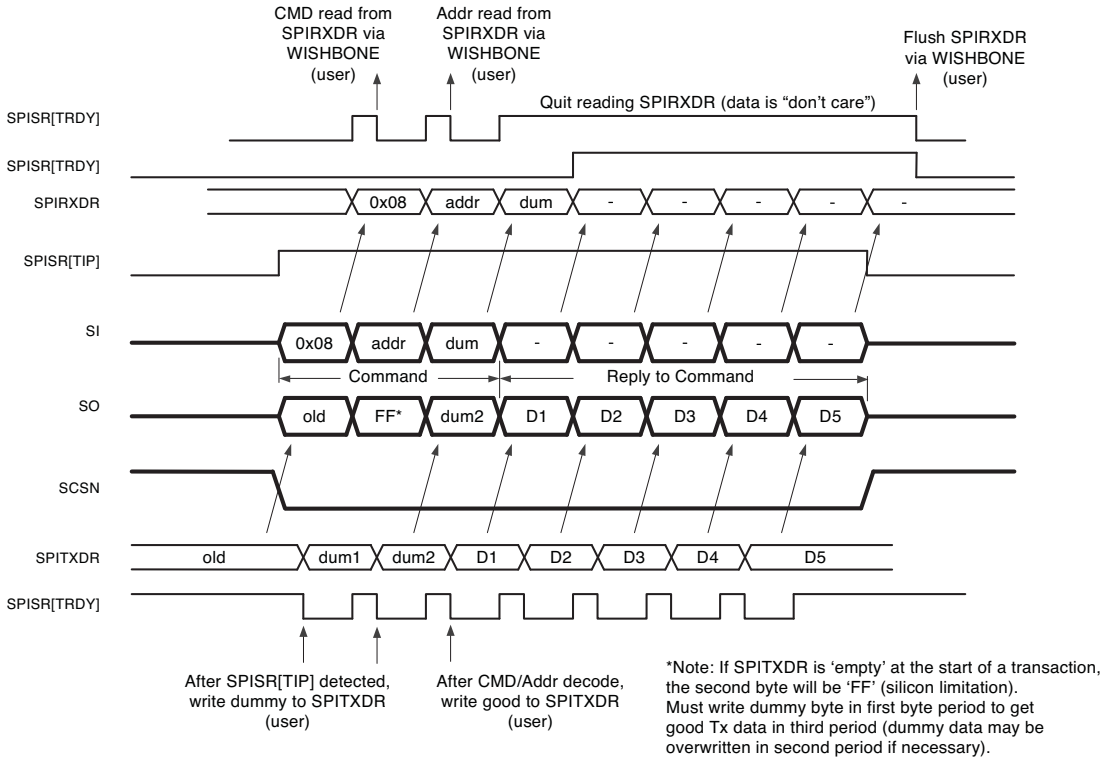


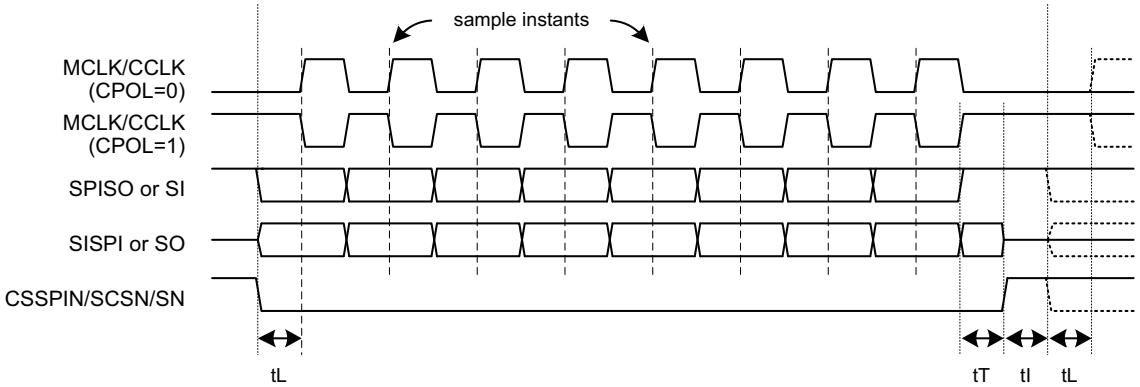
Figure 17-19. Minimally Specified SPI Transaction



*Note: If SPITXDR is ‘empty’ at the start of a transaction, the second byte will be ‘FF’ (silicon limitation). Must write dummy byte in first byte period to get good Tx data in third period (dummy data may be overwritten in second period if necessary).

SPI Timing Diagrams

Figure 17-20. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)

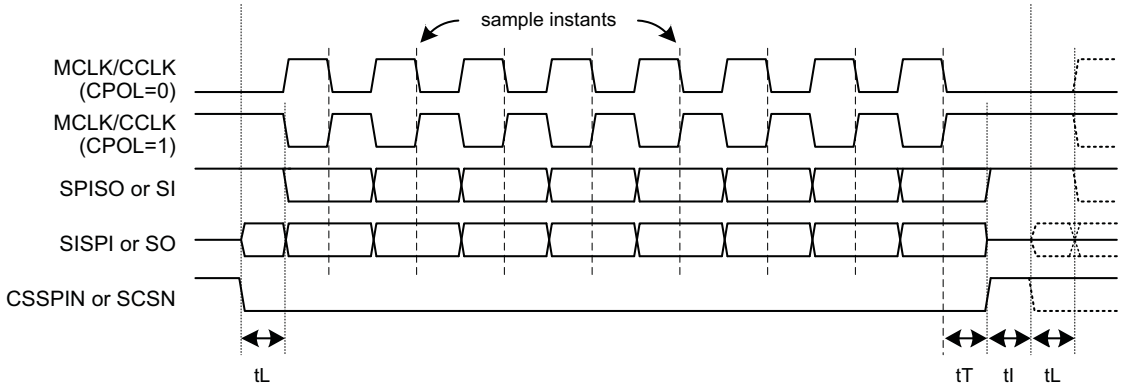


MSB first (LSBF=0):	MSB	bit6	bit5	bit4	bit3	bit2	bit1	LSB
LSB first (LSBF=1):	LSB	bit1	bit2	bit3	bit4	bit5	bit6	MSB

tL = TLead_XCNT
tT = TTrail_XCNT
tL = Tidle_XCNT

*Note: MachXO2 SPI configuration modes only support CPHA = CPOL = LSBF = TXEDGE = 0

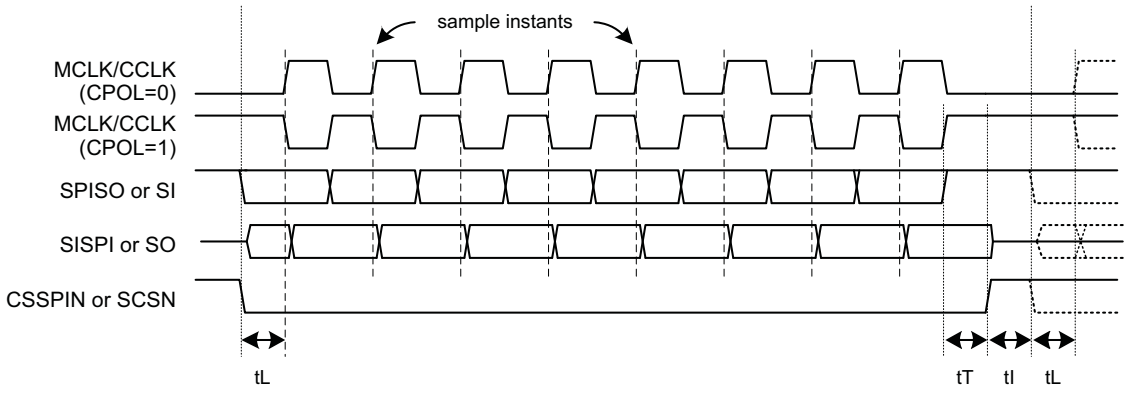
Figure 17-21. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)



MSB first (LSBF=0):	MSB	bit6	bit5	bit4	bit3	bit2	bit1	LSB
LSB first (LSBF=1):	LSB	bit1	bit2	bit3	bit4	bit5	bit6	MSB

tL = TLead_XCNT
tT = TTrail_XCNT
tL = Tidle_XCNT

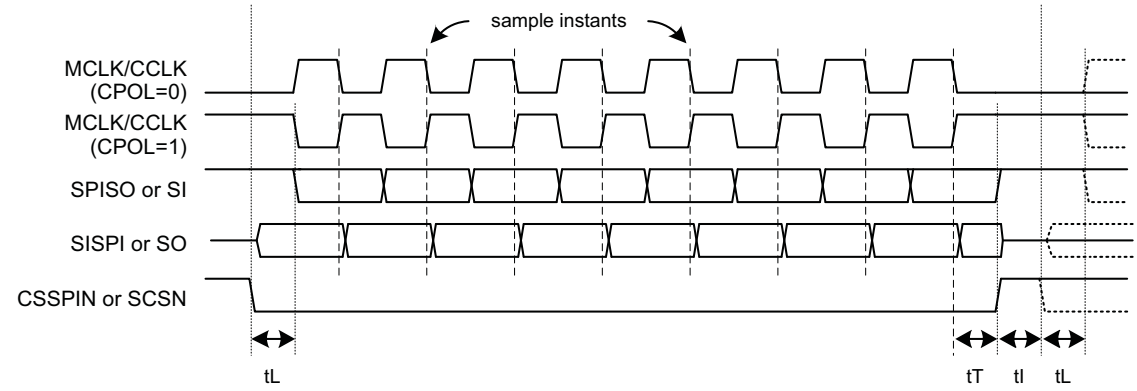
Figure 17-22. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)



MSB first (LSBF=0):	MSB	bit6	bit5	bit4	bit3	bit2	bit1	LSB
LSB first (LSBF=1):	LSB	bit1	bit2	bit3	bit4	bit5	bit6	MSB

tL = TLead_XCNT
tT = TTrail_XCNT
tL = Tidle_XCNT

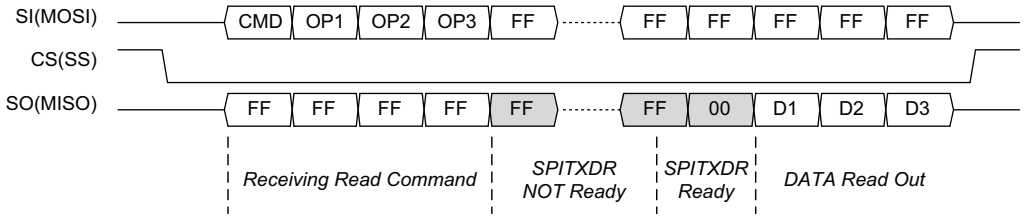
Figure 17-23. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)



MSB first (LSBF=0):	MSB	bit6	bit5	bit4	bit3	bit2	bit1	LSB
LSB first (LSBF=1):	LSB	bit1	bit2	bit3	bit4	bit5	bit6	MSB

tL = TLead_XCNT
tT = TTrail_XCNT
tL = Tidle_XCNT

Figure 17-24. Slave SPI Dummy Byte Response (SPICR2[SDBRE]) Timing



SPI Simulation Model

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

Table 17-28. SPI Simulation Model

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

Table 17-28. SPI Simulation Model

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/

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.

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 17-29 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 17-29. 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
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 17-30. 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 will update 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 17-31. 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 will not be 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 17-32. Timer/Counter Set Top Counter Value 0

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

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

Table 17-33. Timer/Counter Set Top Counter Value 1

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

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 17-34. Timer/Counter Set Compare Counter Value 0

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

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

Table 17-35. Timer/Counter Set Compare Counter Value 1

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

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 bits [7:0] of the compare value. TCOCRSET1 register holds the upper eight bits [15:8] of the compare value.

Table 17-36. 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_clki. Due to the clock domain crossing, there is a one-clock uncertainty when the reset is effective. This bit has higher priority then WBPAUSE.
- 0: Disabled
- 1: Enabled

WBPAUSE — Pause the 16-bit counter
- 1: Pause
- 0: Normal

Table 17-37. 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 17-38. 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 17-39. 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 17-40. 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 17-41. 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 17-42. 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 17-43. 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 17-44. 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 17-45. Timer/Counter Status Register

TCSR								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 17-46. 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 was 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 was asserted. Write a '1' to this bit to clear the interrupt. Note the interrupt line is asserted for only 1 clock cycle.
1: Compare Match Flag Interrupt
0: No interrupt

IRQOVF — Interrupt Status for Overflow Flag.
When enabled, indicates OVF was asserted. Write a '1' to this bit to clear the interrupt. Note the interrupt line is asserted for only 1 clock cycle.
1: Overflow Flag Interrupt
0: No interrupt

Table 17-47. 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

Timer Counter Simulation Model

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

Table 17-48. Timer/Counter Simulation Mode

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

Table 17-48. Timer/Counter Simulation Mode (Continued)

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

Flash Memory (UFM/Configuration) Access

Designers can access the Flash Memory Configuration Logic interface using the JTAG, SPI, I^2C, or WISHBONE interfaces. The MachXO2 Flash Memory consists of two sectors:

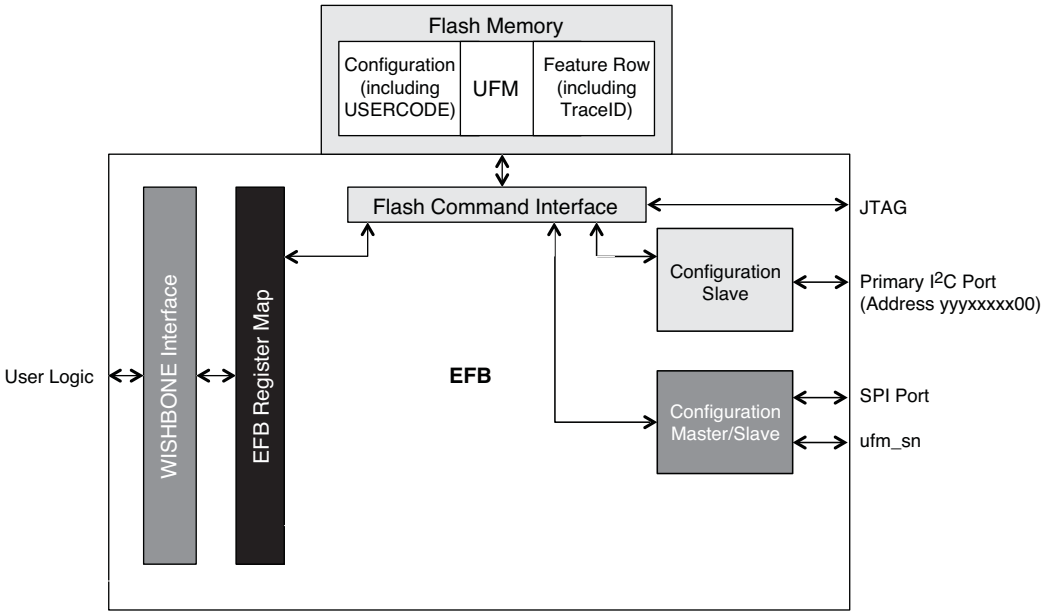
- User Flash Memory (UFM)
  - MachXO2-640 and higher density devices provide one sector of User Flash Memory (UFM).
- Configuration
  - Configuration consists of two sectors Configuration Flash and the Feature Row.

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

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 17-25 illustrates the interfaces to the UFM and Configuration Memory sectors.

Figure 17-25. 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 will be 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 will temporarily disable certain features of the device including:

- *Power Controller*
- *GSR*
- *Hardened User SPI port*

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

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 I^2C port must be enabled. For more details, refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

The WISHBONE Interface communicates to the Configuration Logic through a set of data, control and status registers. Table 17-49 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 17-49. 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
CFGRXDR	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 17-50. 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. See “WISHBONE Framing” on page 50 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 17-51. 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 will be written to the Flash Memory (UFM/Configuration) logic. Bit 0 is LSB.

Figure 17-26. 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
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 was enabled. This port has priority over the I^2C and WISHBONE ports and will pre-empt 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^2C Active. Indicates the I^2C port has started actively communicating with the Configuration Logic while WBCE was enabled. This port has priority over the WISHBONE ports and will pre-empt any existing, and prohibit any new WISHBONE transaction. This bit is capable of generating an interrupt.

1: I^2C port active
0: I^2C port not active

Table 17-52. Flash Memory (UFM/Configuration) Receive Data

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

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

Table 17-53. 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 was 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 was 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 was 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 was 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 was asserted. Write a '1' to this bit to clear the interrupt.
1: Slave SPI Active Interrupt
0: No interrupt

IRQI2CACT Interrupt Status for I^2C Active.
When enabled, indicates I2CACT was asserted. Write a '1' to this bit to clear the interrupt.

1: I^2C Active Interrupt
0: No interrupt

Table 17-54. 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 I^2C Active
1: Interrupt generation enabled
0: Interrupt generation disabled

Table 17-55. 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 17-56. 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 I^2C 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 I^2C Block. Use I2C_1_ IRQ for further source resolution.
- 1: A bit is set in register I2C_1_ IRQ
- 0: No interrupt

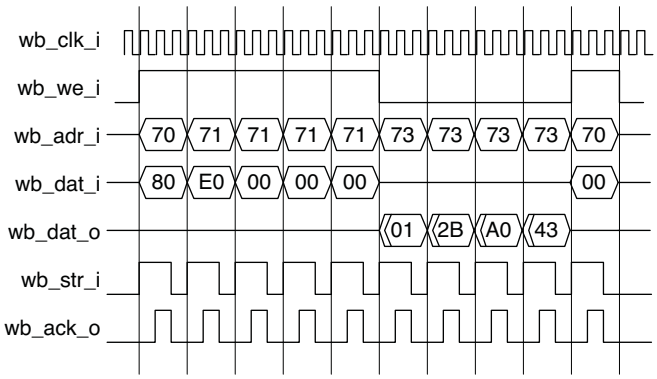
WISHBONE Framing

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 17-57. Command Framing Protocol, by Interface

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

Figure 17-27. WISHBONE Read Device ID Example (-1200 HC Device)



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.

Command Summary by Application

Table 17-58. 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^2C 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 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 17-59. 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[1]	0x79	LSC_REFRESH	Launch boot sequence (same as toggling PROGRAMN pin).
STANDBY	0x7D	LSC_DEVICE_CTRL	Triggers the Power Controller to enter or wake from standby mode
Bypass	0xFF	ISC_NOOP	Null operation.
Enable Configuration Interface (Transparent Mode)	0x74	ISC_ENABLE_X	Enable Transparent Configuration Flash access – All user I/Os (except the hardened user SPI and primary user I^2C 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.)

Table 17-59. Configuration Flash (Sector 0) Commands (Continued)

Command Name	Command MSB LSB	SVF Command Name	Description
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 pages to read. Address Register is post-incremented.
Erase	0x0E	ISC_ERASE	Erase the Config Flash, Done bit, Security bits and USERCODE
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
Program FEABITs	0xF8	LSC_PROG_FEABITS	Program the FEA bits

1. The Refresh commands are not supported by the software simulation model.

Table 17-60. 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^2C 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.

Command Descriptions by Command Code

Table 17-61. 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 I^2C address, sysCONFIG port persistence, Boot mode, OTP, etc.)
- 0: No action
- 1: Erase

s: Erase SRAM
- 0: No action
- 1: Erase

Notes: Poll the BUSY bit (or wait, see Table 17-93) 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 17-62. 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 17-63. 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 (e.g. 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 I^2C port operation.

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

Table 17-64. 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 IxEE Exxx xxxx xxFB xxCD xxxx xxxx

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

D bit 8 Flash or SRAM Done Flag
When C = 0 SRAM Done bit has been programmed
- D = 1 Successful Flash to SRAM transfer
- D = 0 Failure in the Flash to SRAM transfer

When C=1 Flash Done bit has been programed
- D = 1 Programmed
- D = 0 Not Programmed

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

B bit 12: Busy Flag (1 = busy)

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

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

EEE bits[25:23]: Configuration Check Status
000: No Error
001: ID ERR
010: CMD ERR
011: CRC ERR
100: Preamble ERR
101: Abort ERR
110: Overflow ERR
111: SDM EOF

(all other bits reserved)

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

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

Example: 0x3C 00 00 00
Read 4-byte Status Register e.g. 0x00 00 20 00 (fail flag set)

Table 17-65. 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 17-66. 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 17-67. 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 200us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

Table 17-68. Program Configuration Flash (0x70)

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
	x		Y	70	00 00 01	W	16B	16 bytes UFM write data

Example: 0x70 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.

Operands (0x00 00 00) are equivalent to (0x00 00 01).

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

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

Table 17-69. Read Configuration Flash (0x73) (WISHBONE/SPI)

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

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

*Operand: 0001 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 num_pages > 1

**Data Size: (num_pages * 16) bytes

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

***Examples: 0x73 10 00 01
0 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 (4 pages total)

Note: The maximum speed which one page of data (num_page=1) can be read through the WISHBONE is 36 MHz. There is no restriction on SPI speeds besides the port limitations.

Table 17-70. Read Configuration Flash (0x73) (I^2C/WISHBONE/SPI)

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

Note: This applies when Configuration Flash is read through I^2C, WISHBONE or SPI

*Operand: 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 num_pages > 1

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

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

***Examples: 0x73 00 00 01
0 bytes dummy followed by 1 page CFG data (16 bytes total)

0x73 00 00 04
Read 2 pages dummy, followed by three (1 page CFG data, followed by 4 bytes dummy) (5 pages and 12 bytes total)

Note: The maximum speed which one page of data (num_page=1) can be read through the WISHBONE is 36 MHz. There is no restriction on I^2C and SPI speeds besides the port limitations.

Table 17-71. 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	—	—	—

Notes: This command is required to enable modification of the UFM, configuration Flash, or non-volatile registers (NVR). Terminate this command with command 0x26 followed by command 0xFF.

Exercising this command will temporarily disable certain features of the device, notably GSR, user SPI port, primary user I^2C port and Power Controller. These features are restored when the command is terminated.

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

Example: 0x74 08 00 00
Enable UFM/configuration interface for change access

Table 17-72. 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 will Launch Boot sequence

Must have only two operands

After completing the Refresh command (e.g. SPI SN deassertion or I^2C stop), further bus accesses are prohibited for the duration of $t_{REFRESH}$. Violating this requirement will cause 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^2C port may be left in an undefined (non-idle) state. The likely hood of this condition is design and route dependent. To positively return the Secondary I^2C port to the idle state, write a value of 0x40 to register I2C_2_CMDR via WISHBONE immediately after device reset is released. This will cause a short low-pulse on SCK as the hard-block signals a STOP on the bus then returns to the idle state. Failure to manually return the Secondary I^2C port to the idle state may result in an I^2C bus lock-up condition. Normal I^2C activity can be commenced without additional delay

Table 17-73. 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. I^2C has the following ways:
   a. Primary I^2C Configuration port – Address match to the I^2C Configuration address (No other settings required)

b. Primary or Secondary I^2C User port – Address match the I^2C 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 TN1198, Power Estimation and Management for MachXO2 Devices.

Table 17-74. 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 17-75. 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 17-76. 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 200us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

Table 17-77. Enable Configuration Interface (Offline) (0xC6))

UFM	CFG	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
	x			C6	08 00 00	—	—	—

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 will tri-state all user I/Os (except persisted sysCONFIG ports). User logic ceases to function. UFM remains accessible.

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

Table 17-78. 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 200us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

Table 17-79. Read UFM (0xCA) (WISHBONE/SPI)

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

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

where: pp..pp: num_pages – Number of UFM pages to read

**Data Size (num_pages * 16) bytes

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

***Examples: 0xCA 10 00 01
Read 0 bytes dummy followed by 1 page UFM data (16 bytes total)

0xCA 10 00 04
Read 1 page dummy followed by 3 pages UFM data (4 pages total)

Note: The maximum speed which one page of data (num_page=1) can be read using WISHBONE and no wait states is 16.6MHz. 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 TN1204, MachXO2

Programming and Configuration Usage Guide. SPI transactions in MachXO2 always meet the minimum retrieval delay requirement. No special timing is necessary for SPI.

Table 17-80. Read UFM (0xCA) (WISHBONE/SPI/I²C)

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

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

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

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

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

***Examples: 0xCA 00 00 01

Read 0 bytes dummy followed by 1 page UFM data (16 bytes total)

0xCA 00 00 04

Read 2 pages dummy followed by 3 (1 page UFM data, followed by 4 bytes dummy) (5 pages total and 12 bytes)

Note: The maximum speed which one page of data (num_page=1) can be read using WISHBONE and no wait states is 16.6MHz. 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 TN1204, MachXO2 Programming and Configuration Usage Guide. SPI and I²C transactions in MachXO2 always meet the minimum retrieval delay requirement. No special timing is necessary for SPI or I²C.

Table 17-81. 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 17-93) 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 17-82. 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 200us) 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.

Table17-83. 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 200us) 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 17-84. 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 17-85

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

Table 17-85. 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 17-86. 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 17-85

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 17-87. 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 I^2C 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 once it has been program the first time.

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

Program Feature Row with User I^2C address set to 1, default user TraceID string, Custom ID code of 12 34

Table 17-88. 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 I^2C 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 17-89. 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, e.g. 0x80 (busy flag set)

Table 17-90. 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: I^2C 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 once they have been programmed the first time.

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

Table 17-91. 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
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: I^2C 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 17-92. 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

Interface to Configuration Flash

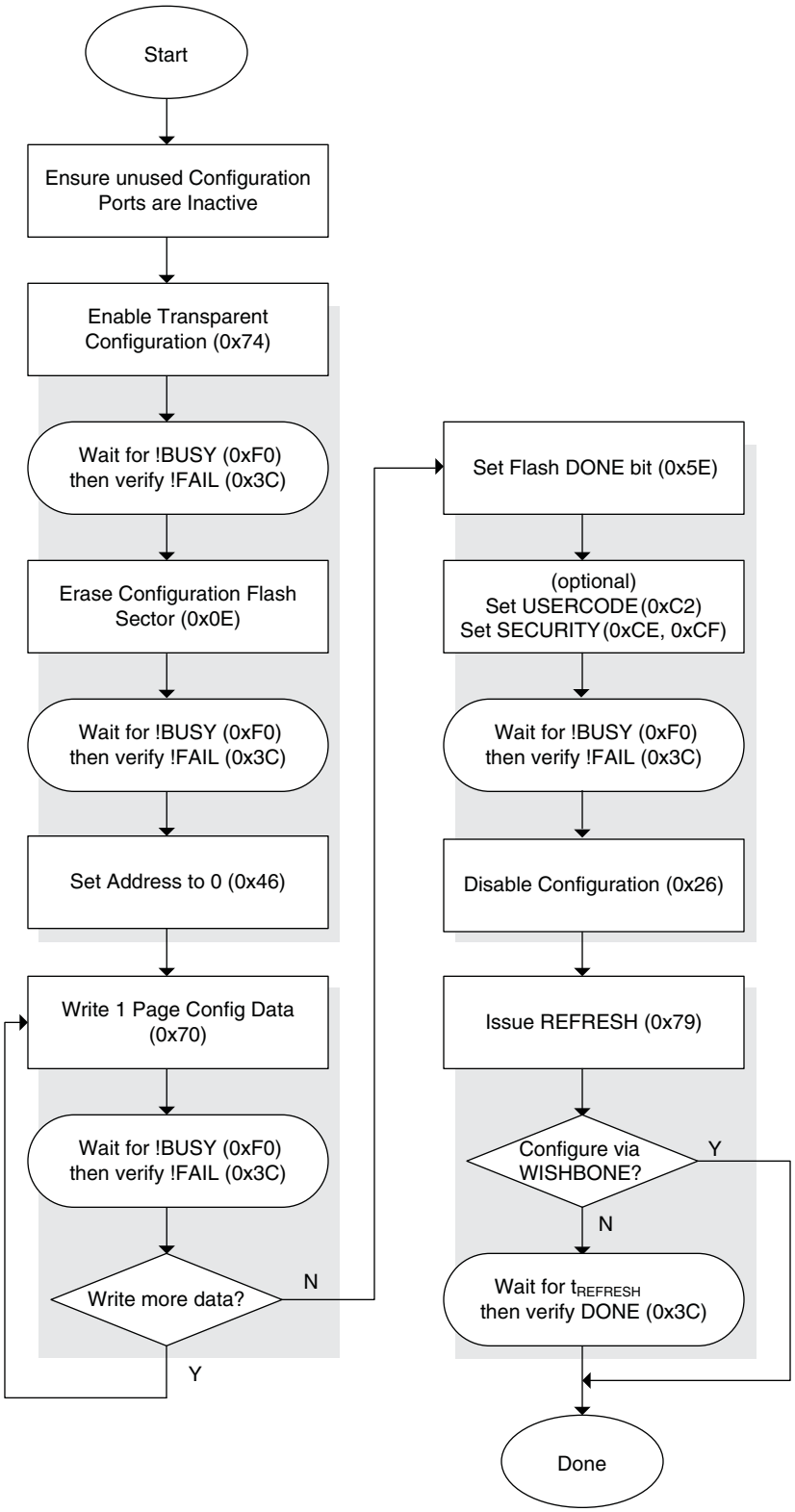
The WISHBONE interface of the EFB module allows a WISHBONE host to access the configuration resources of the MachXO2 devices. This can be particularly useful for reading data from configuration resources such as USERCODE and TraceID. Most importantly, this feature allows users to update the Configuration 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 Tables 17-49 to 17-57 for information on these registers.

Figure 17-28 shows a basic flow diagram for implementing a Configuration Flash Update initiated via any of the sysCONFIG ports (I^2C, SPI, or WISHBONE).

For detailed information on MachXO2 programming and configuration, see TN1204, MachXO2 Programming and Configuration Usage Guide.

Figure 17-28. Basic Configuration Flash Update Example



Flash Memory Erase and Program Performance

Table 17-93. Flash Memory (UFM/Configuration) Performance in MachXO2 Devices[1]

		MachXO2 -256	MachXO2 -640	MachXO2 -640U	MachXO2 -1200	MachXO2 -1200U	MachXO2 -2000	MachXO2 -2000U	MachXO2 -4000	MachXO2 -7000
CFG Erase	Min.	400	600	800	800	1100	1100	1800	1800	2800
	Max.	700	1100	1400	1400	1900	1900	3100	3100	4800
CFG Program	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	Min.	—	300	400	400	500	500	600	600	900
	Max.	—	600	700	700	900	900	1000	1000	1600
UFM Program	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

1. All times are averages, in (ms). SRAM erase times are < 0.1ms.

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 (us): tEPV = tE + tP + tV

where:
tE = tECFG + tEUFM[1]
tP = 0.2us * number of Pages to program[2]
tV = (8 * number of Pages programmed) * BusEff * tBUSCLK

Table 17-94. E/P/V Calculation parameters

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

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

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 (e.g. 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 17-95. 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 5us 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 200us if not polling)
		+			
6	W	C9	00 00 01	10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	Write UFM Page 1 Data (Note: Address automatically incremented)
		-			
		+			
7	W	3C	00 00 00	—	Poll Configuration Status Register
	R			xx xx bx xx	
		-			(poll until Busy Flag clear, or wait 200us if not polling)
		+			
8	W	26	00 00	—	Disable Configuration Interface
		-			
		+			
9	W	FF	—	—	Bypass (NOP)
		-			

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, discussed in "WISHBONE Framing" on page 50.

Table 17-96. 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 5us 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)
		-			

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, discussed in "WISHBONE Framing" on page 50.

Table 17-97. 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 5us if not polling)
		+			
3	W	47	00 00 00	—	Init UFM address to 0000
		-			
		+			
4	W	CA	10 00 03		Read two pages of UFM data, after one page of dummy bytes.3
	R			xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	
		-			
		+			
5	W	26	00 00	—	Disable Configuration Interface
		-			
		+			
6	W	FF	—	—	Bypass (NOP)
		-			

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.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
June 2012	01.0	Initial release.
August 2012	01.1	Timer/Counter Control 1 table – Corrected names of four LSBs.
		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.
		Read Feature Row (0xE7) table – Updated CMD (Hex) column.
		Read FEABITs (0xFB) table – Removed example below table.
		Read USERCODE (0xC0) table – Data Size column updated. EN Required" value changed from "N" to "Y/N" and example text updated.
		Updated Timer/Counter Control 0 table and Timer/Counter Control 1 table.
		Updated Basic Configuration Flash Update Example diagram.
		Device ID Table – Updated Device Name column.
		Read Status Register (0x3C) table – Updated Data Format column.
		Verify Device ID Code (0xE2) table – "EN Required" value changed from "N" to "Y" and example text updated.
October 2012	01.2	Added restriction: Primary port can be used as Configuration/UFM port or as a user port, but not both.
		Added restriction: Primary I^2C port is unavailable while in ISC_ENABLE_X (transparent) configuration access mode.
April 2013	01.3	Read Configuration Flash (0x73) (I^2C/WISHBONE/SPI) table – Corrected table title.
		Read Feature Row (0xE7) table – Updated Data Format in the table and description.
		Updated information in the I^2C Master Read/Write Example (via WISHBONE) figure.
		Updated examples in the Read UFM (0xCA) (WISHBONE/SPI/I^2C) table.
		Added note: SECURITY and SECURITY PLUS commands are mutually exclusive.
		Added Erase/Program/Verify time calculation example.
		Updated (decreased) the maximum WISHBONE clock rate for page reads from 36MHz to 16.6MHz.
		Corrected BUSY wait times (1000ns -> 200ns) in Write Two UFM Pages table.
		Updated Basic Configuration Flash Update Example, changed "Wait for !BUSY" to "Wait for tREFRESH" in last step.
		Added Wait for tREFRESH caution to Refresh command description.
		Clarified Secondary I^2C non-idle reset issue after REFRESH.

MachXO2 Hardware Checklist

September 2012 Technical Note TN1208

Introduction

When designing complex hardware using the MachXO2™ PLD, designers must pay special attention to critical hardware configuration requirements. This technical note steps through these critical hardware requirements related to the MachXO2 device. This document does not provide detailed step-by-step instructions but gives a high-level summary checklist to assist in the design process.

The MachXO2 ultra-low power, instant-on, non-volatile PLDs are available in three versions – ultra low power (ZE) and high performance (HC and HE) devices. HC devices have an internal linear voltage regulator which supports external VCC supply voltages of 3.3V or 2.5V. ZE and HE devices only accept 1.2V as the external VCC supply voltage. With the exception of power supply voltage, all three types of devices (ZE, HC and HE) are functionally and pin compatible with each other.

This technical note assumes that the reader is familiar with the MachXO2 device features as described in the MachXO2 Family Data Sheet.

The critical hardware areas covered in this technical note include:

- Power supplies as they relate to the MachXO2 supply rails and how to connect them to the PCB and the associated system
- Configuration and how to connect the configuration mode selection for proper power up configuration
- Device I/O interface and critical signals

Important: Users should refer to the following documents for detailed recommendations.

- TN1068, Power Decoupling and Bypass Filtering for Programmable Devices
- TN1198, Power Estimation and Management for MachXO2 Devices
- TN1202, MachXO2 sysIO Usage Guide
- TN1203, Implementing High-Speed Interfaces with MachXO2 Devices
- TN1204, MachXO2 Programming and Configuration Usage Guide
- TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices

Power Supply

The VCC and VCCIO0 power supplies determine the MachXO2 internal “power good” condition. These supplies need to be at a valid and stable level before the device can become operational. In addition, there are VCCIO1-5 supplies that power the remaining I/O banks. Table 18-1 shows the power supplies and the appropriate voltage levels for each.

Refer to the MachXO2 Family Data Sheet for more information on the voltage levels.

Table 18-1. Power Supply Description and Voltage Levels

Supply	Voltage (Nominal Value)	Description
VCC	1.2V	Core power supply for 1.2V devices (ZE and HE)
	2.5V/3.3V	Core power supply for 2.5V/3.3V devices (HC)
VCCIOx	1.2V to 3.3V	Power supply pins for I/O Bank x. There are up to five I/O banks.

Power Estimation

Once the MachXO2 device density, package and logic implementation is decided, power estimation can be performed using the Power Calculator tool which is provided as part of the Lattice Diamond® design software. While performing power estimation the user should keep two specific goals in mind.

1. Power supply budgeting should be considered based on the maximum of the power-up in- rush current, configuration current or maximum DC and AC current for a given system environmental condition.
2. The ability of the system environment and MachXO2 device packaging to support the specified maximum operating junction temperature.

By determining these two criteria, system design planning can take the MachXO2 power requirements into consideration early in the design phase.

This is explained in TN1198, Power Estimation and Management for MachXO2 Devices.

Configuration Considerations

MachXO2 devices contain two types of memory, SRAM and Flash. SRAM is volatile memory and contains the active configuration. Flash is non-volatile memory that provides on-chip storage for the SRAM configuration data.

The MachXO2 includes multiple programming and configuration interfaces:

- 1149.1 JTAG
- Self download
- Slave SPI
- Master SPI
- Dual Boot
- I^2C
- WISHBONE bus

For ease of prototype debugging it is recommended that every PCB should have easy access to the programming and configuration pins.

For a detailed description of the programming and configuration interfaces please refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

The use of external resistors is always needed if the configuration signals are being used to handshake with other devices. Pull-up and pull-down resistor (4.7K) recommendations on different configuration pins are listed below.

Table 18-2. Default State of the sysCONFIG Pins

Pin Name	Pin Function (Configuration Mode)	Pin Direction (Configuration Mode)	Default Function (User Mode)
PROGRAMN	PROGRAMN	Input with weak pull-up	PROGRAMN
INITN	I/O	I/O with weak pull-up	User-defined I/O
DONE	I/O	I/O with weak pull-up	User-defined I/O
MCLK/CCLK	SSPI	Input with weak pull-up, external pull-up	User-defined I/O
SN	SSPI	Input with weak pull-up, external pull-up	User-defined I/O
SI/SPISI	SSPI	Input	User-defined I/O
SO/SOSPI	SSPI	Output	User-defined I/O
CSSPIN	I/O	I/O with weak pull-up	User-defined I/O
SCL	I2C	Bi-Directional open drain, external pull-up	User-defined I/O
SDA	I2C	Bi-Directional open drain, external pull-up	User-defined I/O
TDI	TDI	Input with weak pull-up	TDI
TDO	TDO	Output with weak pull-up	TDO
TCK	TCK	Input	TCK
TMS	TMS	Input with weak pull-up	TMS
JTAGENB	I/O	Input with weak pull-down	I/O

PROGRAMN Initial Power Considerations

The MachXO2 PROGRAMN is permitted to become a general purpose I/O. The PROGRAMN only becomes a general purpose I/O after the configuration bitstream is loaded. When power is applied to the MachXO2 the PROGRAMN input performs the PROGRAMN function. It is critical that any signal input to the PROGRAMN have a high-to-low transition period that is longer than the VCC (min) to INITN rising edge time period. Transitions faster than this time period prevent the MachXO2 from becoming operational. Refer to the description of PROGRAMN in TN1204, MachXO2 Programming and Configuration Usage Guide.

Pin-out Considerations

The MachXO2 PLDs support many applications with high-speed interfaces. These include various rule-based pin-outs that need to be understood prior to the implementation of the PCB design. The pin-out selection must be completed with an understanding of the interface building blocks of the FPGA fabric. These include IOLOGIC blocks such as DDR, clock resource connectivity, and PLL usage. Refer to TN1203, Implementing High-Speed Interfaces with MachXO2 Devices for rules pertaining to these interface types.

True-LVDS Output Pin Assignments

True-LVDS outputs are on the top bank (Bank 0) of the MachXO2-1200 and higher density devices. When using the LVDS outputs, a 2.5V or 3.3V supply needs to be connected to the Bank 0 VCCIO supply rails. Refer to TN1202, MachXO2 sysIO Usage Guide for more information on this.

HSTL, SSTL and Referenced LVCMOS Pin Assignments

The externally-referenced I/O standards (HSTL and SSTL) and internally referenced LVCMOS require an external reference voltage. Each I/O bank supports one reference voltage (VREF). Any I/O in the bank can be configured as the input reference voltage pin. This pin is a regular I/O if it is not used as a reference voltage input. The VREF pin(s) should get the highest priority for pin assignment. The input reference voltage can also be generated internally from the VREF generator. Again, there is one VREF generator per bank and its programmable settings include OFF, 45% of VCCIO, 50% of VCCIO, and 55% of VCCIO. Programming of the internal VREF generator and the external VREF pin cannot be set at the same time for a particular bank since there is only one VREF per bank.

PCI Clamp Pin Assignment

PCI clamps are available on the bottom I/O bank (Bank 2) of the MachXO2-1200 and higher density devices. When the system design calls for PCI clamp, these pins should be assigned to I/O Bank 2. For the clamp characteristic, refer to the IBIS buffer models either on the Lattice web site or in the Lattice Diamond design software.

Checklist

	MachXO2 Hardware Checklist Item	OK	N/A
1	Power Supply		
1.1	Core Supply VCC at 1.2V		
1.2	Core Supply VCC at 2.5V or 3.3V		
1.3	I/O power supply VCCIO 0-5 at 1.2V to 3.3V		
1.4	Power Estimation		
2	Configuration		
2.1	Configuration options		
2.2	Pull-up on PROGRAMN, INITN, DONE		
2.3	Pull-up on SPI mode pins		
2.4	Pull-up on I^2C mode pins		
2.5	JTAG default logic levels		
2.6	PROGRAMN high-to-low transition time period is larger than the VCC (min) to INITN rising edge time period		
3	I/O pin assignment		
3.1	True LVDS pin assignment considerations		
3.2	HSTL, SSTL and referenced LVCMOS pin assignment considerations		
3.3	PCI clamp requirement considerations		

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2011	01.0	Initial release.
June 2012	01.1	Updated document with new corporate logo.
		Added external pull-up requirement on SPI signals and updated Configuration Considerations section.
September 2012	01.2	Added PROGRAMN Initial Power Considerations section.
		Added item 2.6 to the Checklist table.

Section III. MachXO2 Family Handbook Revision History

MachXO2 Family Handbook Revision History

Revision History

Date	Handbook Revision Number	Change Summary
November 2010	01.0	Initial release.
January 2011	01.1	MachXO2 Family Data Sheet updated to version 01.1.
		Technical note TN1199 updated to version 01.1.
		Technical note TN1200 updated to version 01.1.
		Technical note TN11201 updated to version 01.1.
		Technical note TN1202 updated to version 01.1.
		Technical note TN1203 updated to version 01.1.
		Technical note TN1205 updated to version 01.1.
		Technical note TN1206 updated to version 01.1.
April 2011	01.2	MachXO2 Family Data Sheet updated to version 01.2.
		Technical note TN1199 updated to version 01.2.
		Technical note TN1200 updated to version 01.2.
		Technical note TN1202 updated to version 01.2.
		Technical note TN1203 updated to version 01.2.
May 2011	01.3	MachXO2 Family Data Sheet updated to version 01.3.
		Technical note TN1198 updated to version 01.1.
		Technical note TN1204 updated to version 01.2.
		Technical note TN1206 updated to version 01.2.
June 2011	01.4	Technical note TN1205 updated to version 02.0.
June 2011	01.5	Technical note TN1199 updated to version 01.3.
July 2011	01.6	Technical note TN1199 updated to version 01.4.
		Technical note TN1202 updated to version 01.3.
		Technical note TN1203 updated to version 01.3.
		Technical note TN1205 updated to version 01.5.
		Technical note TN1206 updated to version 01.4.
August 2011	01.7	MachXO2 Family Data Sheet updated to version 01.4.
		Technical note TN1204 updated to version 01.3.
		Technical note TN1205 updated to version 02.2.
August 2011	01.8	MachXO2 Family Data Sheet updated to version 01.5.
		Technical note TN1199 updated to version 01.5.
September 2011	01.9	Technical note TN1205 updated to version 02.3.
October 2011	02.0	Technical note TN1205 updated to version 02.4.
February 2012	02.1	MachXO2 Family Data Sheet updated to version 01.6.
		Technical note TN1199 updated to version 01.6.
		Technical note TN1205 updated to version 02.9.
February 2012	02.2	MachXO2 Family Data Sheet updated to version 01.7.
		Technical note TN1198 updated to version 01.2.
		Technical note TN1205 updated to version 03.0.

Date	Handbook Revision Number	Change Summary
February 2012	02.2 (cont.)	Technical note TN1202 updated to version 01.4.
		Technical note TN1203 updated to version 01.4.
		Technical note TN1201 updated to version 01.2.
		Technical note TN1199 updated to version 01.7.
		Technical note TN1204 updated to version 01.5.
		Technical note TN1207 updated to version 01.1.
		Technical note TN1200 updated to version 01.3.
		Technical note TN1206 updated to version 01.5.
February 2012	02.3	Technical note TN1205 updated to version 03.1.
March 2012	02.4	MachXO2 Family Data Sheet updated to version 01.8.
		Removed TN1200, MachXO2 Density Migration, from the handbook.
May 2012	02.5	Technical note TN1199 updated to version 01.8.
June 2012	02.6	Technical note TN1204 updated to version 02.0.
		Technical note TN1205 updated to version 04.0.
		New technical note TN1246 added to handbook.
July 2012	02.7	Technical note TN1204 updated to version 02.1.
August 2012	02.8	Technical note TN1246 updated to version 01.1.
August 2012	02.9	Technical Note TN1199 updated to version 01.9.
August 2012	03.0	Technical Note TN1199 updated to version 02.0.
		Added Technical Note TN1208, MachXO2 Hardware Checklist.
August 2012	03.1	Technical Note TN1199 updated to version 02.1.
August 2012	03.2	Technical note TN1206 updated to version 01.6.
September 2012	03.3	Technical Note TN1204 updated to version 02.2.
		Technical Note TN1208 updated to version 01.2.
September 2012	03.4	Technical Note TN1199 updated to version 02.2.
October 2012	03.5	Technical Note TN1204 updated to version 02.3.
		Technical Note TN1205 updated to version 04.1.
		Technical Note TN1246 updated to version 01.2.
October 2012	03.6	Technical Note TN1198 updated to version 01.3.
February 2013	03.7	MachXO2 Family Data Sheet updated to version 02.0.
		Technical Note TN1198 updated to version 01.4.
		Technical Note TN1206 updated to version 01.8.
May 2013	03.8	Technical Note TN1202 updated to version 01.5.
		Technical Note TN1203 updated to version 01.5.
		Technical Note TN1204 updated to version 02.4.
		Technical Note TN1246 updated to version 01.3.

Note: For detailed revision changes, please refer to the revision history for each document.