MachXO3 sysCLOCK PLL Design and User Guide

Technical Note

FPGA-TN-02058-1.5

January 2022
Disclaimers

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

Acronyms in This Document.............................................................................................................6
1. Introduction..................................................................................................................................7
2. Clock/Control Distribution Network................................................................................................7
3. MachXO3L/LF Top Level View .....................................................................................................8
4. Primary Clocks ...........................................................................................................................9
5. Dynamic Clock Mux (DCMA).........................................................................................................9
  5.1. DCMA Primitive Definition .......................................................................................................9
  5.2. DCMA Declaration in VHDL Source Code .............................................................................10
  5.3. DCMA Usage with Verilog Source Code ................................................................................10
6. Dynamic Clock Control (DCCA) ....................................................................................................11
  6.1. DCCA Primitive Definition ......................................................................................................11
  6.2. DCCA Declaration in VHDL Source Code .............................................................................11
  6.3. DCCA Usage with Verilog Source Code ................................................................................12
7. Edge Clocks ................................................................................................................................13
  7.1. Edge Clock Bridge ..................................................................................................................13
8. ECLKBRIDGECS Primitive Definition .......................................................................................14
  8.1. ECLKBRIDGECS Declaration in VHDL Source Code .............................................................14
  8.2. ECLKBRIDGECS Usage with Verilog Source Code .................................................................15
9. Edge Clock Synchronization (ECLKSYNCA) ..............................................................................16
  9.1. ECLKSYNCA Primitive Definition ........................................................................................16
  9.2. ECLKSYNCA Declaration in VHDL Source Code .................................................................16
  9.3. ECLKSYNCA Usage with Verilog Source Code ....................................................................17
10. Clock Dividers (CLKDIVC).........................................................................................................18
  10.1. CLKDIVC Primitive Definition .............................................................................................18
  10.2. CLKDIVC Declaration in VHDL Source Code ......................................................................19
  10.3. CLKDIVC Usage with Verilog Source Code .........................................................................19
11. sysCLOCK PLL ..........................................................................................................................20
  11.1. Functional Description ..........................................................................................................20
  11.1.1. PLL Divider Blocks ............................................................................................................20
  11.2. PLL Features .......................................................................................................................21
  11.2.1. Standby Mode ....................................................................................................................21
  11.2.2. Fractional-N synthesis ......................................................................................................21
  11.2.3. WISHBONE Ports .............................................................................................................21
  11.3. PLL Inputs and Outputs .........................................................................................................21
  11.3.1. CLKI Input .......................................................................................................................21
  11.3.2. CLKFB Input .....................................................................................................................22
  11.3.3. RST Input ..........................................................................................................................22
  11.3.4. RESETM Input ...................................................................................................................22
  11.3.5. RESETC Input ...................................................................................................................22
  11.3.6. RESETD Input ...................................................................................................................23
  11.3.7. ENCLKOP Input ................................................................................................................23
  11.3.8. ENCLKOS Input .................................................................................................................23
  11.3.9. ENCLKOS2 Input .............................................................................................................23
  11.3.10. ENCLKOS3 Input ............................................................................................................24
  11.3.11. STDBY Input ..................................................................................................................24
  11.3.12. PHASESEL Input ............................................................................................................24
  11.3.13. PHASEDIR Input .............................................................................................................24
  11.3.14. PHASESTEP Input .........................................................................................................24
  11.3.15. CLKOP Output ...............................................................................................................24
  11.3.16. CLKOS Output ...............................................................................................................24
  11.3.17. CLKOS2 Output .............................................................................................................25
  11.3.18. CLKOS3 Output .............................................................................................................25
11.3.19. DPHSRC Output .................................................................................................................. 25
11.3.20. LOCK Output .................................................................................................................... 25
11.3.21. WISHBONE Ports ............................................................................................................. 25
11.4. PLL Attributes ....................................................................................................................... 25
11.4.1. FIN ...................................................................................................................................... 25
11.4.2. CLKI_DIV, CLKFB_DIV, CLKOP_DIV, CLKOS_DIV, CLKOS2_DIV, CLKOS3_DIV ................. 26
11.4.3. FREQUENCY_PIN_CLKI, FREQUENCY_PIN_CLKOP, FREQUENCY_PIN_CLKOS, FREQUENCY_PIN_CLOS2,
       FREQUENCY_PIN_CLKOS3 ........................................................................................................ 26
11.4.4. Frequency Tolerance – CLKOP, CLKOS, CLKOS2, CLKOS3 ............................................ 26
12. MachXO3L/LF PLL Primitive Definition .................................................................................. 27
13. Dynamic Phase Adjustment ....................................................................................................... 29
14. Frequency Calculation ............................................................................................................... 31
15. Fractional-N Synthesis Operation .......................................................................................... 32
16. Low Power Features ................................................................................................................ 33
16.1. Dynamic Clock Enable .......................................................................................................... 33
16.2. Standby Mode ....................................................................................................................... 33
17. Configuring the PLL Using IPexpress ..................................................................................... 34
17.1. Configuration Tab .................................................................................................................. 34
17.2. Configuration Modes ........................................................................................................... 34
17.2.1. Frequency Mode ................................................................................................................ 34
17.2.2. Divider Mode: .................................................................................................................. 35
17.3. IPexpress Output .................................................................................................................. 37
17.4. Use of the Spreadsheet View .............................................................................................. 37
18. PLL Reference Clock Switch (PLLREFCS) .............................................................................. 38
18.1. PLLREFCS Declaration in VHDL Source Code ..................................................................... 38
18.2. PLLREFCS Usage with Verilog Source Code ....................................................................... 39
19. Internal Oscillator (OSCH) ..................................................................................................... 40
19.1. OSCH Primitive Definition ................................................................................................... 40
19.2. OSCH Declaration in VHDL Source Code ........................................................................... 41
19.3. OSCH Instantiation in Verilog Source Code ......................................................................... 42
Appendix A. Primary Clock Sources and Distribution ................................................................. 43
Appendix B. Edge Clock Sources and Connectivity ....................................................................... 45
Appendix C. Clock Preferences .................................................................................................... 47
Appendix D. PLL WISHBONE Bus Operation .............................................................................. 50
D.1. PLL Architecture .................................................................................................................... 50
Appendix E. MachXO3L/LF Device Usage with Lattice Diamond Design Software ...................... 59
Technical Support Assistance ...................................................................................................... 61
Revision History .......................................................................................................................... 62
**Figures**

Figure 3.1. MachXO3L/LF Clocking Structure .................................................................................8
Figure 5.1. DCMA Primitive Symbol .............................................................................................9
Figure 6.1. DCCA Primitive Symbol ............................................................................................ 11
Figure 8.1. ECLKBRIDGECS Primitive Symbol ............................................................................14
Figure 9.1. ECLKSYNCA Primitive Symbol ..................................................................................16
Figure 10.1. MachXO3L/LF Clock Divider ....................................................................................18
Figure 10.2. CLKDIVC Primitive Symbol .......................................................................................18
Figure 11.1. MachXO3L/LF PLL Block Diagram ..........................................................................20
Figure 11.2. RST and RESETM Timing Diagram ...........................................................................22
Figure 11.3. RESETC and RESETD Timing Diagram .......................................................................23
Figure 12.1. PLL Primitive Symbol ...............................................................................................27
Figure 13.1. GPLL VCO Phase Rotation Timing Diagram ...............................................................30
Figure 17.1. IPexpress Main Window for PLL Module .................................................................34
Figure 17.2. MachXO3L/LF PLL Configuration Tab ......................................................................35
Figure 17.3. Spreadsheet View Example .......................................................................................37
Figure 18.1. PLLREFCS Primitive Symbol ....................................................................................38
Figure 19.1. OSCH Primitive Symbol ...........................................................................................40
Figure A.1. MachXO3L/LF Primary Clock Sources and Distribution ............................................43
Figure A.2. MachXO3L/LF Primary Clock Muxes .........................................................................43
Figure A.3. MachXO3L/LF Primary Clock Muxes – MachXO3L/LF-256 and MachXO3L/LF-640 44
Figure B.1. MachXO3L/LF Edge Clock Sources and Connectivity .............................................45
Figure B.2. MachXO3L/LF Edge Clock Bridge Sources and Connectivity ......................................46
Figure C.1. Timing Preferences/INPUT_SETUP, CLOCK_TO_OUT Preference Window ................49
Figure D.1. PLL CLKOP Output Section ......................................................................................51
Figure E.1. IPexpress Main Window for PLL Module Using Diamond ...........................................59
Figure E.2. Spreadsheet View for Clock Selection Using Diamond ...............................................60

**Tables**

Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers ...................................................7
Table 5.1. DCMA Primitive Port Definition ....................................................................................9
Table 6.1. DCCA Primitive Port Definition ..................................................................................11
Table 8.1. ECLKBRIDGECS Primitive Port Definition ...............................................................14
Table 9.1. ECLKSYNCA Primitive Port Definition ....................................................................16
Table 10.1. CLKDIVC Primitive Port Definition ...........................................................................18
Table 10.2. CLKDIVC Primitive Attribute Definition .................................................................19
Table 12.1. PLL Primitive Port Definition .....................................................................................27
Table 13.1. PHASESEL Signal Settings Definitions ......................................................................29
Table 13.2. PHASEDIR Signal Settings Definitions ......................................................................29
Table 17.1. User Parameters in the IPexpress User Interface .........................................................35
Table 18.1. PLLREFCS Primitive Port Definition ..........................................................................38
Table 19.1. OSCH Primitive Port Definition ................................................................................40
Table 19.2. OSCH Primitive Attribute Definition .........................................................................40
Table 19.3. OSCH Supported Frequency Settings .........................................................................41
Table D.1. PLL Data Bus Port Definitions ......................................................................................50
Table D.2. EFB WISHBONE Locations for PLL Registers ............................................................51
Table D.3. PLL Register Descriptions ............................................................................................53
# Acronyms in This Document

A list of acronyms used in this document.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKDIVC</td>
<td>Clock Dividers</td>
</tr>
<tr>
<td>DCMA</td>
<td>Dynamic Clock Mux</td>
</tr>
<tr>
<td>DCCA</td>
<td>Dynamic Clock Control</td>
</tr>
<tr>
<td>ECLKSYNCA</td>
<td>Edge Clock Synchronization Control</td>
</tr>
<tr>
<td>EFB</td>
<td>Embedded Function Block</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-Programmable Gate Array</td>
</tr>
<tr>
<td>IP</td>
<td>Intellectual Property</td>
</tr>
<tr>
<td>NVCM</td>
<td>Non-Volatile Configuration Memory</td>
</tr>
<tr>
<td>OSCH</td>
<td>Internal Oscillator</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase-locked Loop</td>
</tr>
<tr>
<td>PLLREFCS</td>
<td>PLL Reference Clock Switch</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
</tbody>
</table>
1. Introduction

The MachXO3™ devices support a variety of I/O interfaces such as display interfaces (7:1 LVDS) high speed DDR interfaces with gearing. In order to support applications that use these interfaces, the MachXO3L/LF 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 MachXO3L/LF 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 MachXO3L/LF device.

The number of PLLs, edge clocks, and clock dividers for each MachXO3L/LF device are listed in Table 1.1.

Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>XO3L/LF-640</th>
<th>XO3L/LF-1200</th>
<th>XO3L/LF-2100</th>
<th>XO3L/LF-4300</th>
<th>XO3L/LF-6900</th>
<th>XO3L/LF-9400</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of PLLs</td>
<td>General purpose PLLs</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Number of edge clocks</td>
<td>Edge clocks for high speed applications (top &amp; bottom sides)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Number of clock dividers</td>
<td>Clock dividers for DDR applications</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

2. Clock/Control Distribution Network

MachXO3L/LF 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. Other clock sources include clock input pins, internal nodes, PLLs, clock dividers, and the internal oscillator.
3. **MachXO3L/LF Top Level View**

A top level view of the major clocking resources for the MachXO3L/LF device is shown in Figure 3.1.
4. Primary Clocks

The MachXO3L/LF 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

5. Dynamic Clock Mux (DCMA)

The MachXO3L/LF 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.

5.1. DCMA Primitive Definition

The DCMA primitive can be instantiated in the source code of a design as defined in this section. **Figure 5.1** and **Table 5.1** show the DCMA definitions.

![Figure 5.1. DCMA Primitive Symbol](image)

**Table 5.1. DCMA Primitive Port Definition**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK0</td>
<td>I</td>
<td>Clock input port zero – this the default</td>
</tr>
<tr>
<td>CLK1</td>
<td>I</td>
<td>Clock input port one</td>
</tr>
</tbody>
</table>
| SEL       | I   | Select port  
|           |     | − SEL=0 for CLK0  
|           |     | − SEL=1 for CLK1 |
| DCMOUT    | O   | Clock output port |
5.2. DCMA Declaration in VHDL Source Code

Library Instantiation

library machXO3;
use machXO3.all;

Component Declaration

component DCMA
    portCLK0: 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);

5.3. 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));
6. Dynamic Clock Control (DCCA)

The MachXO3L/LF 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 it is not needed, and thus saves power.

6.1. DCCA Primitive Definition

The DCCA primitive can be instantiated in the source code of a design as defined in this section. Figure 6.1 and Table 6.1 show the DCCA definitions.

![DCCA Symbol](image)

**Figure 6.1. DCCA Primitive Symbol**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKI</td>
<td>I</td>
<td>Clock input</td>
</tr>
<tr>
<td>CE</td>
<td>I</td>
<td>Clock enable port</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– CE = 0 – disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– CE = 1 – enabled</td>
</tr>
<tr>
<td>CLKO</td>
<td>O</td>
<td>Clock output port</td>
</tr>
</tbody>
</table>

6.2. DCCA Declaration in VHDL Source Code

**Library Instantiation**

```vhdl
library machXO3;
use machXO3.all;
```

**Component Declaration**

```vhdl
component DCCA
port(CLKI: in std_logic;
    CE:in std_logic;
    CLKO: out std_logic);
end component;
```

**DCCA Instantiation**

```vhdl
I1: DCCA
port map(CLKI=> CLKI,
    CE => CE,
    CLKO=>CLKO);
end component;
```
6.3. DCCA Usage with Verilog Source Code

**Component Declaration**

```verilog
module DCCA (CLKI, CE, CLKO);
  input CLKI;
  input CE;
  output CLKO;
endmodule
```

**DCCA Instantiation**

```verilog
DCCA I1(.CLKI (CLKI),
  .CE (CE),
  .CLKO (CLKO));
```
7. Edge Clocks

There are two edge clock resources on the top and bottom sides of the device. 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. Edge Clock Sources and Connectivity for detailed information on the ECLK locations and connectivity.

The sources of edge clocks are:
- Dedicated clock pins
- PLL outputs
- Internal nodes

7.1. Edge Clock Bridge

The MachXO3L/LF 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 Implementing High-Speed Interfaces with MachXO3 Devices (FPGA-TN-02057).

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. Edge Clock Sources and Connectivity.
8. 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 8.1 and Table 8.1 show the ECLKBRIDGECS definitions.

![ECLKBRIDGECS Primitive Symbol](image)

Table 8.1. ECLKBRIDGECS Primitive Port Definition

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK0</td>
<td>I</td>
<td>Clock Input port zero – this the default.</td>
</tr>
<tr>
<td>CLK1</td>
<td>I</td>
<td>Clock Input port one</td>
</tr>
<tr>
<td>SEL</td>
<td>I</td>
<td>Select port</td>
</tr>
<tr>
<td>ECSOUT</td>
<td>O</td>
<td>Clock output port</td>
</tr>
</tbody>
</table>

8.1. ECLKBRIDGECS Declaration in VHDL Source Code

Library Instantiation

```vhdl
library machXO3;
use machXO3.all;
```

Component Declaration

```vhdl
component ECLKBRIDGECS
port (CLK0:in std_logic;
     CLK1: in std_logic;
     SEL: in std_logic;
     ECSOUT: out std_logic);
end component;
```

ECLKBRIDGECS Instantiation

```vhdl
I1: ECLKBRIDGECS
port map (CLK0=>CLK0,
          CLK1 => CLK1,
          SEL => SEL,
          ECSOUT => ECSOUT);
```
8.2. 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));
9. Edge Clock Synchronization (ECLKSYNCA)

The MachXO3L/LF 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.

9.1. ECLKSYNCA Primitive Definition

The ECLKSYNCA primitive can be instantiated in the source code of a design as defined in this section. Figure 9.1 and Table 9.1 show the ECLKSYNCA definitions.

![ECLKSYNCA Symbol](image)

Figure 9.1. ECLKSYNCA Primitive Symbol

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECLK</td>
<td>I</td>
<td>Clock Input port</td>
</tr>
<tr>
<td>STOP</td>
<td>I</td>
<td>Control signal to stop edge clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— STOP=0 Clock is active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— STOP=1 Clock is off</td>
</tr>
<tr>
<td>ECLKO</td>
<td>O</td>
<td>Clock output port</td>
</tr>
</tbody>
</table>

Table 9.1. ECLKSYNCA Primitive Port Definition

9.2. ECLKSYNCA Declaration in VHDL Source Code

```vhdl
Library Instantiation
library machXO3;
use machXO3.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);
```
9.3. ECLKSYNCA Usage with Verilog Source Code

**Component Declaration**

```verilog
class ECLKSYNCA (ECLKI, STOP, ECLKO);
input ECLKI;
input STOP;
output ECLKO;
endmodule
```

**ECLKSYNCA Instantiation**

```verilog
ECLKSYNCA I1 (.ECLKI (ECLKI),
               .STOP (STOP),
               .ECLKO (ECLKO));
```
10. Clock Dividers (CLKDIVC)

There are four clock dividers available in the MachXO3L/LF devices. 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 10.1.

![Clock Divider Block Diagram](image)

### 10.1. CLKDIVC Primitive Definition

The CLKDIVC primitive can be instantiated in the source code of a design as defined in this section. Figure 10.2, Table 10.1, and Table 10.2 show the CLKDIVC definitions.

![CLKDIVC Symbol](image)

#### Table 10.1. CLKDIVC Primitive Port Definition

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKI</td>
<td>I</td>
<td>Clock input</td>
</tr>
<tr>
<td>RST</td>
<td>I</td>
<td>Reset input - asynchronously forces all outputs low</td>
</tr>
<tr>
<td>ALIGNWD</td>
<td>I</td>
<td>Signal is used for word alignment.</td>
</tr>
<tr>
<td>CDIV1</td>
<td>O</td>
<td>Divide by 1 output port. When RST = 1 CDIV1 output does not toggle and stays either L or H</td>
</tr>
<tr>
<td>CDIVX</td>
<td>O</td>
<td>Divide by 2, 3.5 or 4 output port</td>
</tr>
</tbody>
</table>

© 2014-2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at [www.latticesemi.com/legal](http://www.latticesemi.com/legal). All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.
Table 10.2. CLKDIVC Primitive Attribute Definition

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSR</td>
<td>GSR Enable</td>
<td>ENABLED, DISABLED</td>
<td>DISABLED</td>
</tr>
<tr>
<td>DIV</td>
<td>CLK Divider</td>
<td>2.0, 3.5 or 4.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>

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 Implementing High-Speed Interfaces with MachXO3 Devices (FPGA-TN-02057).

10.2. CLKDIVC Declaration in VHDL Source Code

Library Instantiation

```
library machXO3;
use machXO3.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);
```

10.3. 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));
```
11. sysCLOCK PLL

The MachXO3L/LF PLL provides features such as clock injection delay removal, frequency synthesis, and phase adjustment. Figure 11.1 shows a block diagram of the MachXO3L/LF PLL.

![MachXO3L/LF PLL Block Diagram](image)

Figure 11.1. MachXO3L/LF PLL Block Diagram

11.1. Functional Description

11.1.1. 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 MachXO3 Family Data Sheet (FPGA-DS-02032). 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 MachXO3 Family Data Sheet (FPGA-DS-02032).
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.

Phase Alignment:
After the device has reached steady state operation after power-up, and after releasing RST and RESETM, the CLKOP and CLKOS outputs will be edge aligned (for related frequencies) when Phase Adjustment is set to 0 degrees. Under the same conditions, CLKOS2 and CLKOS3 will be aligned to CLKOP and CLKOS to within 1 VCO clock period.

Edge Trim Adjustment (Static Mode):
The CLKOP and CLKOS ports can be finely tuned with an edge trim adjustment feature.

11.2. PLL Features

11.2.1. Standby Mode
The MachXO3L/LF 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.

11.2.2. Fractional-N synthesis
The MachXO3L/LF 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.

11.2.3. WISHBONE Ports
The MachXO3L/LF 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 WISHBONE Bus Operation.

11.3. PLL Inputs and Outputs

11.3.1. 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.
The dedicated GPLL pins and PCLK pins located on the top and bottom sides provide direct connection to the PLL input. The PCLK pins located on the left and right sides use primary clock routing to connect to the PLL input pin.

11.3.2. 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.

11.3.3. 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 11.2 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.

11.3.4. 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 11.2 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 11.2. RST and RESETM Timing Diagram](image)

11.3.5. 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 11.3 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.

### 11.3.6. 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 11.3 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.

![RESETC and RESETD Timing Diagram](image)

**Figure 11.3.** RESETC and RESETD Timing Diagram

### 11.3.7. 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.

### 11.3.8. 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.

### 11.3.9. 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.
11.3.10. **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.

11.3.11. **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.

11.3.12. **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.

11.3.13. **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.

11.3.14. **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.

11.3.15. **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.

11.3.16. **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.
11.3.17. 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.

11.3.18. 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.

11.3.19. 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.

11.3.20. 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.

11.3.21. WISHBONE Ports
The WISHBONE parts are listed in Appendix D. PLL WISHBONE Bus Operation along with the description of how to use them. The WISHBONE ports are optional.

11.4. 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.

11.4.1. FIN
The input frequency can be any value within the specified frequency range based upon the divider settings.
11.4.2. 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.

11.4.3. FREQUENCY_PIN_CLKI, FREQUENCY_PIN_CLKOP, FREQUENCY_PIN_CLKOS,
        FREQUENCY_PIN_CLKOS2, FREQUENCY_PIN_CLKOS3

These input and output clock frequencies determine the divider values.

11.4.4. Frequency Tolerance – CLKOP, CLKOS, CLKOS2, CLKOS3

When the desired output frequency is not achievable, users may enter the frequency tolerance of the clock output.
12. MachXO3L/LF PLL Primitive Definition

The PLL primitive can be instantiated in the source code of a design as defined in this section. Figure 12.1 and Table 12.1 show the EHXPLLJ definitions.

![EHXPLLJ Diagram]

**Figure 12.1. PLL Primitive Symbol**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKI</td>
<td>I</td>
<td>Input clock to PLL</td>
</tr>
<tr>
<td>CLKFB</td>
<td>I</td>
<td>Feedback clock</td>
</tr>
<tr>
<td>PHASESEL[1:0]</td>
<td>I</td>
<td>Select which output is affected by Dynamic Phase adjustment ports.</td>
</tr>
<tr>
<td>PHASEDIR</td>
<td>I</td>
<td>Dynamic Phase adjustment direction.</td>
</tr>
<tr>
<td>PHASESTEP</td>
<td>I</td>
<td>Dynamic Phase step – toggle shifts VCO phase adjust by one step</td>
</tr>
<tr>
<td>LOADREG</td>
<td>I</td>
<td>Dynamic Phase Load – toggle loads divider phase adjustment values into PLL</td>
</tr>
<tr>
<td>CLKOP</td>
<td>O</td>
<td>Primary PLL output clock (with phase shift adjustment)</td>
</tr>
<tr>
<td>CLKOS</td>
<td>O</td>
<td>Secondary PLL output clock (with phase shift adjust)</td>
</tr>
<tr>
<td>CLKOS2</td>
<td>O</td>
<td>Secondary PLL output clock2 (with phase shift adjust)</td>
</tr>
<tr>
<td>CLKOS3</td>
<td>O</td>
<td>Secondary PLL output clock3 (with phase shift adjust)</td>
</tr>
<tr>
<td>LOCK</td>
<td>O</td>
<td>PLL LOCK, asynchronous signal. Active high indicates PLL is locked to input and feedback signals.</td>
</tr>
<tr>
<td>INTLOCK</td>
<td>O</td>
<td>PLL internal LOCK, asynchronous signal. Active high indicates PLL lock using internal feedback.*</td>
</tr>
<tr>
<td>REFCLK</td>
<td>O</td>
<td>Output of reference clock mux</td>
</tr>
<tr>
<td>DPHSRC</td>
<td>O</td>
<td>Dynamic phase source – ports or WISHBONE is active</td>
</tr>
<tr>
<td>STDBY</td>
<td>I</td>
<td>Standby signal to power down the PLL</td>
</tr>
<tr>
<td>PLLWAKESYNC</td>
<td>I</td>
<td>PLL wake-up sync. Enable PLL to switch from internal to user feedback path when the PLL wakes up.*</td>
</tr>
<tr>
<td>RST</td>
<td>I</td>
<td>PLL Reset without resetting the M-divider. Active high reset.</td>
</tr>
<tr>
<td>Port Name</td>
<td>I/O</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----</td>
<td>-------------</td>
</tr>
<tr>
<td>RESETM</td>
<td>I</td>
<td>PLL Reset - includes resetting the M-divider. Active high reset.</td>
</tr>
<tr>
<td>RESETC</td>
<td>I</td>
<td>Reset for CLKOS2 output divider only. Active high reset.</td>
</tr>
<tr>
<td>RESETD</td>
<td>I</td>
<td>Reset for CLKOS3 output divider only. Active high reset.</td>
</tr>
<tr>
<td>ENCLKOP</td>
<td>I</td>
<td>Clock Enable for CLKOP output</td>
</tr>
<tr>
<td>ENCLKOS</td>
<td>I</td>
<td>Clock Enable for CLKOS output – only available if CLKOS port is active</td>
</tr>
<tr>
<td>ENCLKOS2</td>
<td>I</td>
<td>Clock Enable for CLKOS2 output – only available if CLKOS2 port is active</td>
</tr>
<tr>
<td>ENCLKOS3</td>
<td>I</td>
<td>Clock Enable for CLKOS3 output – only available if CLKOS3 port is active</td>
</tr>
<tr>
<td>PLLCLK</td>
<td>I</td>
<td>PLL data bus clock input signal</td>
</tr>
<tr>
<td>PLLRST</td>
<td>I</td>
<td>PLL data bus reset. This resets only the data bus not any register values.</td>
</tr>
<tr>
<td>PLLSTB</td>
<td>I</td>
<td>PLL data bus strobe signal</td>
</tr>
<tr>
<td>PLLWE</td>
<td>I</td>
<td>PLL data bus write enable signal</td>
</tr>
<tr>
<td>PLLADDR [4:0]</td>
<td>I</td>
<td>PLL data bus address</td>
</tr>
<tr>
<td>PLLDATI [7:0]</td>
<td>I</td>
<td>PLL data bus data input</td>
</tr>
<tr>
<td>PLLDATO [7:0]</td>
<td>O</td>
<td>PLL data bus data output</td>
</tr>
<tr>
<td>PLLACK</td>
<td>O</td>
<td>PLL data bus acknowledge signal</td>
</tr>
</tbody>
</table>

*Note*: 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.
13. Dynamic Phase Adjustment

The MachXO3L/LF PLL supports dynamic phase adjustments through either the dynamic phase adjusts ports or the WISHBONE interface using the following method. The WISHBONE interface is covered in more detail in Appendix D. PLL WISHBONE Bus Operation.

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 user 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 13.1 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 13.1. PHASESEL Signal Settings Definitions

<table>
<thead>
<tr>
<th>PHASESEL[1:0]</th>
<th>PLL Output Shifted</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>CLKOS</td>
</tr>
<tr>
<td>01</td>
<td>CLKOS2</td>
</tr>
<tr>
<td>10</td>
<td>CLKOS3</td>
</tr>
<tr>
<td>11</td>
<td>CLKOP</td>
</tr>
</tbody>
</table>

The selected output clock phase will either be advanced or delayed depending upon the value of the PHASEDIR port or signal. Table 13.2 shows the PHASEDIR settings available. The PHASEDIR signal must be stable before the PHASESTEP signal is toggled.

Table 13.2. PHASEDIR Signal Settings Definitions

<table>
<thead>
<tr>
<th>PHASEDIR</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Delayed (lagging)</td>
</tr>
<tr>
<td>01</td>
<td>Advanced (leading)</td>
</tr>
</tbody>
</table>

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.

\[
\text{Step size} = \frac{45^\circ}{\text{Output Divider}}
\]

If the phase shift desired is larger than 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.

The timing diagram shown in Figure 13.1 describes the setup and hold timing requirements for PHASESEL[1:0] and PHASEDIR with respect to PHASESTEP, when dynamically changing the phase controls signals.
Figure 13.1. GPLL VCO Phase Rotation Timing Diagram
14. 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.

\[
\begin{align*}
\text{fOUT} &= \text{fIN} \times \frac{N}{M} \\
\text{fVCO} &= \text{fOUT} \times V \\
\text{fPFD} &= \text{fIN} / M = \text{fFB} / N
\end{align*}
\]

Where:

- \(f\text{OUT}\) is the output frequency.
- \(f\text{IN}\) is the input frequency.
- \(f\text{VCO}\) is the VCO frequency.
- \(f\text{PFD}\) is the PFD (Phase detector) Frequency.
- \(f\text{FB}\) is the Feedback signal Frequency.
- \(N\) is the feedback divider (integer value shown in the IPexpress user interface).
- \(M\) is the input divider (integer value shown in the IPexpress user interface).
- \(V\) is the output divider (integer value shown in the IPexpress user interface).

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. Otherwise, the equation (1) becomes more complex because the two dividers must be included.
15. Fractional-N Synthesis Operation

The MachXO3L/LF 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 user interface 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:

\[
N_{eff} = N + \frac{F}{65536}
\]  
(4)

Where:
- \(N\) is the integer Feedback divider (shown in the IPexpress user interface).
- \(F\) is the value entered into the Fractional-N synthesis box described above.

The output frequency is given by the equation:

\[
f_{OUT} = \frac{f_{IN}}{M} * N_{eff}
\]  
(5)

Where:
- \(f_{OUT}\) is the output frequency.
- \(f_{IN}\) is the input frequency.
- \(M\) is the input divider (shown in the IPexpress user interface).

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 MachXO3 Family Data Sheet (FPGA-DS-02032).
16. Low Power Features

The MachXO3L/LF 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.

16.1. 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 user interface, 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 user interface.

If an output is not enabled in the IPexpress user interface, the ports for that selected output signal will not be present in the module and that output will be inactive.

16.2. 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 user interface. 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 $\tau_{lock}$. When the PLL achieves lock to the external feedback signal, the LOCK signal will be asserted high to indicate that it has locked.
17. 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 17.1 shows the main window when the PLL is selected in Lattice Diamond®. For an example of the equivalent screen in Diamond, see Figure E.1.

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 17.2.

17.1. 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.

17.2. Configuration Modes

There are two modes that can be used to configure the PLL in the Configuration Tab: Frequency Mode and Divider Mode.

17.2.1. Frequency Mode

In this mode, you can enter 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. You 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.
17.2.2. Divider Mode:

In this mode, you can set the input frequency and the divider settings. You will choose the CLKOP divider value to maximize the frequency of the VCO within the acceptable range as specified in MachXO3 Family Data Sheet (FPGA-DS02032).

If the combination of entered values will result in an invalid PLL configuration, you will be prompted by a text box to change the value with a suggestion for the value that is out of range.

![Figure 17.2. MachXO3L/LF PLL Configuration Tab](image)

Table 17.1. User Parameters in the IPexpress User Interface

<table>
<thead>
<tr>
<th>User Parameter</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Mode</td>
<td>User enters desired CLKI and CLKOP frequency</td>
<td>ON/OFF</td>
<td>ON</td>
</tr>
<tr>
<td>Divider Mode</td>
<td>User enters desired CLKI frequency and divider settings</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>CLKI Frequency</td>
<td>Frequency</td>
<td>7 to 400 MHz</td>
<td>100 MHz</td>
</tr>
<tr>
<td>Divider</td>
<td>1 to 40</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User Parameter</td>
<td>Description</td>
<td>Range</td>
<td>Default</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>CLKFB</td>
<td>Feedback mode</td>
<td>CLKOP, CLKOS, CLKOS2, CLKOS3, INT_OP, INT_OS, INT_OS2, INT_OS3, UserClock</td>
<td>CLKOP</td>
</tr>
<tr>
<td></td>
<td>Fractional-N divider enable</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Fractional-N divider</td>
<td>0 to 65535</td>
<td>0</td>
</tr>
<tr>
<td>Output Port Selections</td>
<td>Dynamic phase ports</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Clock enable ports</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Standby ports</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>PLL Reset Options</td>
<td>Provide PLL reset</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Provide PLLM reset</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Provide CLKOS2 reset</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Provide CLKOS3 reset</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>Lock Settings</td>
<td>Provide PLL LOCK signal</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>PLL LOCK is “sticky”</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>WISHBONE Bus</td>
<td>Provide WISHBONE ports</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>CLKOP</td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Clock Divider (in Bypass mode only)</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Desired frequency</td>
<td>3.125 to 400 MHz</td>
<td>100 MHz</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Divider</td>
<td>1-128</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Actual frequency (read only)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Static phase shift (degrees)</td>
<td>0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>Rising edge trim</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Falling edge trim</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Delay multiplier</td>
<td>0, 1, 2, 4</td>
<td>0</td>
</tr>
<tr>
<td>CLKOS</td>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Clock divider (in Bypass mode only)</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Desired frequency</td>
<td>0.024 – 400 MHz</td>
<td>100 MHz</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Divider</td>
<td>1-128</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Actual frequency (read only)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Static phase shift (degrees)</td>
<td>0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>Rising edge trim</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Delay multiplier</td>
<td>0, 1, 2, 4</td>
<td>0</td>
</tr>
<tr>
<td>CLKOS2</td>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Clock divider (in Bypass mode only)</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Desired frequency</td>
<td>0.024 to 400 MHz</td>
<td>100 MHz</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Divider</td>
<td>1-128</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Actual frequency (read only)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Static phase shift (degrees)</td>
<td>0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°</td>
<td>00</td>
</tr>
<tr>
<td>User Parameter</td>
<td>Description</td>
<td>Range</td>
<td>Default</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>CLKOS3</td>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Clock divider (in Bypass mode only)</td>
<td>ON/OFF</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Desired frequency</td>
<td>0.024 – 400 MHz</td>
<td>100 MHz</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Divider</td>
<td>1-128</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Actual frequency (read only)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Static phase shift (degrees)</td>
<td>0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°</td>
<td>0°</td>
</tr>
</tbody>
</table>

17.3. 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 user interface 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.

17.4. Use of the Spreadsheet View

Clock preferences can be set in the Pre-MAP Spreadsheet View. Figure 17.3 shows an example screen shot. The Quadrant and DCS/Pure columns are not applicable to the MachXO3L/LF device.

![Figure 17.3. Spreadsheet View Example](image)
18. PLL Reference Clock Switch (PLLREFCS)

The MachXO3L/LF 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 $t_{\text{LOCK}}$ time specified in MachXO3 Family Data Sheet (FPGA-DS02032) 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 18.1 and Table 18.1 show the PLLREFCS definitions.

![PLLREFCS Symbol](image)

**Figure 18.1. PLLREFCS Primitive Symbol**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK0</td>
<td>NO</td>
<td>CLK0</td>
</tr>
<tr>
<td>CLK1</td>
<td>NO</td>
<td>CLK1</td>
</tr>
</tbody>
</table>
| SEL       | NO  | SEL = 0 CLK0 input is selected  
|           |     | SEL = 1 CLK1 input is selected |
| PLLCSOUT  | NO  | PLLCSOUT    |

**Table 18.1. PLLREFCS Primitive Port Definition**

18.1. PLLREFCS Declaration in VHDL Source Code

**Library Instantiation**

```vhdl
library machXO3;
use machXO3.all;
```

**Component Declaration**

```vhdl
component PLLREFCS
  port (CLK0: in std_logic;  
        CLK1: in std_logic;  
        SEL: in std_logic;  
        PLLCSOUT: out std_logic);
end component;
```

**PLLREFCS Instantiation**

```vhdl
I1: PLLREFCS
  port map (CLK0=> CLK0,  
            CLK1=> CLK1,  
            SEL => SEL,  
            PLLCSOUT=>PLLCSOUT);
```

---

© 2014-2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.
18.2. PLLREFCS Usage with Verilog Source Code

Component Declaration

```verilog
module PLLREFCS (CLK0, CLK1, SEL, PLLCSOUT);
    input CLK0;
    input CLK1;
    input SEL;
    output PLLCSOUT;
endmodule
```

PLLREFCS Instantiation

```verilog
PLLREFCS I1(.CLK0 (CLK0),
    .CLK1 (CLK1),
    .SEL (SEL),
    .PLLCSOUT (PLLCSOUT));
```
19. Internal Oscillator (OSCH)

The MachXO3L/LF device has an internal oscillator that can be used as a clock source in a design. On the MachXO3L/LF device, 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 MachXO3L/LF 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 MachXO3L/LF 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 19.3.

19.1. OSCH Primitive Definition

The OSCH primitive can be instantiated in the source code of a design as defined in this section. Figure 19.1 and Table 19.1 through Table 19.3 show the OSCH definitions.

![OSCH Primitive Symbol](image)

**Figure 19.1. OSCH Primitive Symbol**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STDBY</td>
<td>I</td>
<td>Standby – power down the oscillator in standby mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>− STDBY = 0 OSC output is active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>− STDBY = 1 OSC output is OFF</td>
</tr>
<tr>
<td>OSC</td>
<td>O</td>
<td>Clock output port</td>
</tr>
<tr>
<td>SEDSTDBY</td>
<td>O</td>
<td>Standby – power down SED clock*</td>
</tr>
</tbody>
</table>

*Note: 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>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal Frequency (MHz)</td>
<td>NOM_FREQ</td>
<td>2.08, 2.15, 2.22, ... 66.5, 88.67, 133.0 (See Table 19.3 for a complete listing)</td>
<td>2.08 MHz</td>
</tr>
</tbody>
</table>
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 I²C configuration, SPI or I²C user mode operations, background Flash/NVCM updates or SED.

### 19.2. OSCH Declaration in VHDL Source Code

#### Library Instantiation

```vhdl
library machXO3;
use machXO3.all;
```

#### Component and Attribute Declaration

```vhdl
COMPONENT OSCH
  -- synthesis translate_off
  GENERIC (NOM_FREQ: string := "2.56");
  -- synthesis translate_on
  PORT (STDBY:INstd_logic;
        OSC:OUTstd_logic;
        SEDSTDBY:OUTstd_logic);
END COMPONENT;
```

```vhdl
attribute NOM_FREQ : string;
attribute NOM_FREQ of OSCinst0 : label is "2.56";
```

---

Table 19.3. OSCH Supported Frequency Settings

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.08</td>
<td>4.16</td>
<td>8.31</td>
<td>15.65</td>
<td></td>
</tr>
<tr>
<td>2.15</td>
<td>4.29</td>
<td>8.58</td>
<td>16.63</td>
<td></td>
</tr>
<tr>
<td>2.22</td>
<td>4.43</td>
<td>8.87</td>
<td>17.73</td>
<td></td>
</tr>
<tr>
<td>2.29</td>
<td>4.59</td>
<td>9.17</td>
<td>19.00</td>
<td></td>
</tr>
<tr>
<td>2.38</td>
<td>4.75</td>
<td>9.50</td>
<td>20.46</td>
<td></td>
</tr>
<tr>
<td>2.46</td>
<td>4.93</td>
<td>9.85</td>
<td>22.17</td>
<td></td>
</tr>
<tr>
<td>2.56</td>
<td>5.12</td>
<td>10.23</td>
<td>24.18</td>
<td></td>
</tr>
<tr>
<td>2.66</td>
<td>5.32</td>
<td>10.64</td>
<td>26.60</td>
<td></td>
</tr>
<tr>
<td>2.77</td>
<td>5.54</td>
<td>11.08</td>
<td>29.56</td>
<td></td>
</tr>
<tr>
<td>2.89</td>
<td>5.78</td>
<td>11.57</td>
<td>33.25</td>
<td></td>
</tr>
<tr>
<td>3.02</td>
<td>6.05</td>
<td>12.09</td>
<td>38.00</td>
<td></td>
</tr>
<tr>
<td>3.17</td>
<td>6.33</td>
<td>12.67</td>
<td>44.33</td>
<td></td>
</tr>
<tr>
<td>3.33</td>
<td>6.65</td>
<td>13.30</td>
<td>53.20</td>
<td></td>
</tr>
<tr>
<td>3.50</td>
<td>7.00</td>
<td>14.00</td>
<td>66.50</td>
<td></td>
</tr>
<tr>
<td>3.69</td>
<td>7.39</td>
<td>14.78</td>
<td>88.67</td>
<td></td>
</tr>
<tr>
<td>3.91</td>
<td>7.82</td>
<td>15.65</td>
<td>133.00</td>
<td></td>
</tr>
</tbody>
</table>
**OSCH Instantiation**

```vhdl
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
);
```

**19.3. 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
```
Appendix A. Primary Clock Sources and Distribution

Figure A.1. MachXO3L/LF Primary Clock Sources and Distribution

1 For differential clock inputs, there is a corresponding PCLKCX_x pin for each PCLKTx_x that should be used. The software tools will detect and assign this automatically for differential IO_TYPES. Note that PCLKCX_x pins cannot drive the primary clock trees directly and should not be used as independent, single ended clock inputs.

Note: MachXO3L/LF devices have 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 A.2. MachXO3L/LF Primary Clock Muxes
Figure A.3. MachXO3L/LF Primary Clock Muxes – MachXO3L/LF-256 and MachXO3L/LF-640
Appendix B. Edge Clock Sources and Connectivity

Figure B.1. MachXO3L/LF Edge Clock Sources and Connectivity

Note: 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, require that the ECLKBRIDGECS element be instantiated in the design.
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. 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:
NameFanoutDelay (ns)Site Resource
ROUTE 49 2.892ULPPLL.MCLK toR3C14.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:
NameFanoutDelay (ns)Site Resource
IN_DEL--- 1.431 D5.PAD toD5.INCK RxClk
ROUTE 1 0.843 D5.INCK toULPPLL.CLKIN RxClk_c
MCLK_DEL--- 3.605 ULPPLL.CLKIN to ULPPLL.MCLK
pll_inst/pll_utp_0_0
ROUTE 49 2.892 ULPPLL.MCLK toR3C14.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**
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 Spreadsheet View**
The Pre-Map Preference Editor can be used to set the PLL_PHASE_BACK attribute.

1. Open the Spreadsheet View tool.
2. In the Spreadsheet View window, select Timing Preferences Tab.
The INPUT_SETUP/CLOCK_TO_OUT preference in Timing Preferences window with the PLL phase back feature is shown in Figure C.
Figure C.1. Timing Preferences/INPUT_SETUP, CLOCK_TO_OUT Preference Window
Appendix D. PLL WISHBONE Bus Operation

The MachXO3L/LF 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 Hardened Control Functions in MachXO3 Devices (FPGA-TN-02063) 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 D.1.

Table D.1. PLL Data Bus Port Definitions

<table>
<thead>
<tr>
<th>PLL Port Name</th>
<th>I/O</th>
<th>Description</th>
<th>EFB Port Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLLCLK</td>
<td>I</td>
<td>PLL data bus clock input signal</td>
<td>pll_bus_o[16]</td>
</tr>
<tr>
<td>PLLRST</td>
<td>I</td>
<td>PLL data bus reset. This resets only the data bus, not any register values.</td>
<td>pll_bus_o[15]</td>
</tr>
<tr>
<td>PLLSTB</td>
<td>I</td>
<td>PLL data bus strobe signal.</td>
<td>pll_bus_o[14]</td>
</tr>
<tr>
<td>PLLWE</td>
<td>I</td>
<td>PLL data bus write enable signal</td>
<td>pll_bus_o[13]</td>
</tr>
<tr>
<td>PLLADDR [4:0]</td>
<td>I</td>
<td>PLL data bus address</td>
<td>pll_bus_o[12:8]</td>
</tr>
<tr>
<td>PLLDATI [7:0]</td>
<td>I</td>
<td>PLL data bus data input</td>
<td>pll_bus_o[7:0]</td>
</tr>
<tr>
<td>PLLDATO [7:0]</td>
<td>O</td>
<td>PLL data bus data output</td>
<td>pll_bus_i[8:1]</td>
</tr>
<tr>
<td>PLLACK</td>
<td>O</td>
<td>PLL data bus acknowledge signal</td>
<td>pll_bus_i[0]</td>
</tr>
</tbody>
</table>

D.1. PLL Architecture

The MachXO3L/LF 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 D.1. 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.

Note: A1 Mux does not exist in the CLKOP path, but is shown to illustrate the corresponding B1, C1 and D1 Mux location. See Figure 11.1.
MachXO3 sysCLOCK PLL Design and User Guide
Technical Note

The EFB WISHBONE register map for the PLL registers is shown in Table D.2 (add 0x20 for the corresponding locations to access an optional second MachXO3L/LF PLL).

Table D.2. EFB WISHBONE Locations for PLL Registers

<table>
<thead>
<tr>
<th>Reg.</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>MC1_DIVFBK_FRAC[7:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MC1_DIVFBK_FRAC[15:8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MC1_LOADREG</td>
<td>MC1_DELA[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MC1_PLLPDN</td>
<td>MC1_DELB[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MC1_WBRESET</td>
<td>MC1_DELC[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MC1_USE_DESI</td>
<td>MC1_DELD[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MC1REFIN_RESET</td>
<td>MC1_DIVA[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MC1_PLLRST_ENA</td>
<td>MC1_DIVB[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MC1_MRST_ENA</td>
<td>MC1_DIVC[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>MC1_STDBY</td>
<td>MC1_DIVD[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>MC1_ENABLE_SYNC</td>
<td>MC1_PHIB[2:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MC1_PHIA[2:0]</td>
</tr>
<tr>
<td>B</td>
<td>MC1_DCRST_ENA</td>
<td>MC1_PHID[2:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MC1_RESERVED</td>
<td>MC1_PHIC[2:0]</td>
</tr>
<tr>
<td>C</td>
<td>MC1_DDRST_ENA</td>
<td>MC1_SEL_OUTB[2:0]</td>
<td></td>
<td></td>
<td></td>
<td>MC1_INTFB</td>
<td>MC1_SEL_OUTA[2:0]</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>MC1_LOCK[1:0]</td>
<td></td>
<td>MC1_SEL_OUTC[2:0]</td>
<td></td>
<td></td>
<td>MC1_SEL_OUTD[2:0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>MC1_SEL_DIVA[1:0]</td>
<td></td>
<td>MC1_SEL_DIVB[1:0]</td>
<td></td>
<td></td>
<td>MC1_SEL_DIVC[1:0]</td>
<td>MC1_SEL_DIVD[1:0]</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>MC1_CLKOP_TRIM[3:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>MC1_DYN_SOURCE</td>
<td>MC1_LOCK_SEL[2:0]</td>
<td></td>
<td></td>
<td></td>
<td>MC1_ENABLE_CLK[3:0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MC1_TRIMOS3_BYPASS_N</td>
<td>MC1_TRIMOS2_BYPASS_N</td>
<td>MC1_TRIMOS1_BYPASS_N</td>
<td>MC1_TRIMOS0_BYPASS_N</td>
<td>MC1_DYN_SEL[1:0]</td>
<td>MC1_DIRECTION</td>
<td>MC1_ROTATE</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MC1_LF_RESGRND</td>
<td>MC1_SEL_REF1[2:0]</td>
<td></td>
<td></td>
<td></td>
<td>MC1_EN_UP</td>
<td>MC1_SEL_REF2[2:0]</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>MC1_DIVFBK_ORDER[1:0]</td>
<td>MC1_CLKMUX_FB[1:0]</td>
<td></td>
<td></td>
<td></td>
<td>MC1_SEL_FBK[3:0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>MC1_GMC_RESET</td>
<td>MC1_DIVREF[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MC1_FORCE_VFILT</td>
<td>MC1_DIVFBK[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>MC1_LF_PRESET</td>
<td>MC1_LF_RESET</td>
<td>MC1_TEST_ICP</td>
<td></td>
<td>MC1_EN_FILTER_OPAMP</td>
<td>MC1_FLOAT_ICP</td>
<td>MC1_GPROG[2:0]</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>MC1_KPROG[2:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MC1_IPROG[4:0]</td>
</tr>
</tbody>
</table>
### MachXO3 sysCLOCK PLL Design and User Guide

#### Technical Note

© 2014-2022 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at [www.latticesemi.com/legal](http://www.latticesemi.com/legal).

All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

### Registers

<table>
<thead>
<tr>
<th>Reg.</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>MC1_GMC_PRESET</td>
<td>MC1_RPROG[6:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>MC1_GMCREF_SEL[1:0]</td>
<td>MC1_MFGOUT2_SEL[2:0]</td>
<td>MC1_MFGOUT1_SEL[2:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A</td>
<td>MC1_GMCSEL[3:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1B</td>
<td>MC1_RESERVED[4:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LOCK_STS</td>
</tr>
<tr>
<td>1C</td>
<td>RESERVED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Registers 0 through 11 are user accessible registers. The remaining registers are reserved for Lattice use or read-only access.
### Table D.3. PLL Register Descriptions

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Register Addr (Hex)</th>
<th>Size (Bits)</th>
<th>Description</th>
<th>Default Value</th>
<th>User Access</th>
<th>User Interface Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC1_DIVFBK_FRAC[15:0]</td>
<td>0[7:0] 1[7:0]</td>
<td>16</td>
<td>Fractional-N divider value. Fractional-N divider is equal to this value / 65535.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_LOADREG</td>
<td>2[7]</td>
<td>1</td>
<td>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_DECL values are not the same. A CLKOS2 divider output phase shift will occur if the MC1_DIVC and MC1_DECL values are not the same. A CLKOS3 divider output phase shift will occur if the MC1_DIVD and MC1_DECL values are not the same.</td>
<td>0</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>MC1_PLLPDN</td>
<td>3[7]</td>
<td>1</td>
<td>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.</td>
<td>1</td>
<td>Yes</td>
<td>Yes automatic</td>
</tr>
<tr>
<td>MC1_WBRESET</td>
<td>4[7]</td>
<td>1</td>
<td>PLL reset from Wishbone – Equivalent to the RESETM port operation. 0 = PLL normal operation. 1 = PLL reset active.</td>
<td>0</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_USE_DESI</td>
<td>5[7]</td>
<td>1</td>
<td>Controls whether the Fractional-N divider is used. 0 = PLL normal operation. 1 = Use Fractional-N divider.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_REFIN_RESET</td>
<td>6[7]</td>
<td>1</td>
<td>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.</td>
<td>0</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_PLLRST_ENA</td>
<td>7[7]</td>
<td>1</td>
<td>Enable the PLLRESET port. 0 = PLLRESET port not active. 1 = PLLRESET port is enabled.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_MRST_ENA</td>
<td>8[7]</td>
<td>1</td>
<td>Enable the RESETM port. 0 = RESETM port not active. 1 = RESETM port is enabled.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_STDBY</td>
<td>9[7]</td>
<td>1</td>
<td>Enable the STDBY port on PLL 0 = STDBY port not active. 1 = STDBY port is enabled.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Register Name</td>
<td>Register Addr (Hex)</td>
<td>Size (Bits)</td>
<td>Description</td>
<td>Default Value</td>
<td>User Access</td>
<td>User Interface Access</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>--------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>MC1_ENABLE_SYNC</td>
<td>A[7]</td>
<td>1</td>
<td>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.</td>
<td>0</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_DCRST_ENA</td>
<td>B[7]</td>
<td>1</td>
<td>Enable the RESETDC port – CLKOS2 reset. 0 = RESETDC port not active. 1 = RESETDC port is enabled.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DDRST_ENA</td>
<td>C[7]</td>
<td>1</td>
<td>Enable the RESETDD port – CLKOS3 reset. 0 = RESETDD port not active. 1 = RESETDD port is enabled.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DELA[6:0]</td>
<td>2[6:0]</td>
<td>7</td>
<td>CLKOP section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVA[6:0].</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DELB[6:0]</td>
<td>3[6:0]</td>
<td>7</td>
<td>CLKOS section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVB[6:0].</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DELC[6:0]</td>
<td>4[6:0]</td>
<td>7</td>
<td>CLKOS2 section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVC[6:0].</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DELD[6:0]</td>
<td>5[6:0]</td>
<td>7</td>
<td>CLKOS3 section Delay value for coarse phase adjustments. For zero delay this value should be equal to the value of MC1_DIVD[6:0].</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DIVA[6:0]</td>
<td>6[6:0]</td>
<td>7</td>
<td>CLKOP section output divider setting equal to the Divide value -1.</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DIVB[6:0]</td>
<td>7[6:0]</td>
<td>7</td>
<td>CLKOS section output divider setting equal to the Divide value -1.</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DIVC[6:0]</td>
<td>8[6:0]</td>
<td>7</td>
<td>CLKOS2 section output divider setting equal to the Divide value -1.</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DIVD[6:0]</td>
<td>9[6:0]</td>
<td>7</td>
<td>CLKOS3 section output divider setting equal to the Divide value -1.</td>
<td>7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_PHIA[2:0]</td>
<td>A[2:0]</td>
<td>3</td>
<td>Select the VCO phase shift (0-7) for CLKOP. Each tap represents 45 degree shift of the VCO.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_PHIB[2:0]</td>
<td>A[6:4]</td>
<td>3</td>
<td>Select the VCO phase shift (0-7) for CLKOS. Each tap represents 45 degree shift of the VCO.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_PHIC[2:0]</td>
<td>B[2:0]</td>
<td>3</td>
<td>Select the VCO phase shift (0-7) for CLKOS2. Each tap represents 45 degree shift of the VCO.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Register Name</td>
<td>Register Addr (Hex)</td>
<td>Size (Bits)</td>
<td>Description</td>
<td>Default Value</td>
<td>User Access</td>
<td>User Interface Access</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>MC1_PHID[2:0]</td>
<td>B[6:4]</td>
<td>3</td>
<td>Select the VCO phase shift (0-7) for CLKO5. Each tap represents 45 degree shift of the VCO.</td>
<td>0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_INT_LOCK_STICKY</td>
<td>A[3]</td>
<td>1</td>
<td>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.</td>
<td>1</td>
<td>Yes</td>
<td>Not used</td>
</tr>
<tr>
<td>MC1_RESERVED2</td>
<td>B[3]</td>
<td>1</td>
<td>Not used.</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MC1_SEL_OUTA[2:0]</td>
<td>C[2:0]</td>
<td>3</td>
<td>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.</td>
<td>000</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_SEL_OUTB[2:0]</td>
<td>C[6:4]</td>
<td>3</td>
<td>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.</td>
<td>000</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_SEL_OUTC[2:0]</td>
<td>D[5:3]</td>
<td>3</td>
<td>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.</td>
<td>000</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Register Name</td>
<td>Register Addr (Hex)</td>
<td>Size (Bits)</td>
<td>Description</td>
<td>Default Value</td>
<td>User Access</td>
<td>User Interface Access</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>MC1_SEL_OUTD[2:0]</td>
<td>D[2:0]</td>
<td>3</td>
<td>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.</td>
<td>000</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_INTFB</td>
<td>C[3]</td>
<td>1</td>
<td>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.</td>
<td>0</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_LOCK[1:0]</td>
<td>D[7:6]</td>
<td>2</td>
<td>Frequency lock-detector resolution or sensitivity. 00 = +/- 250 ppm 01 = +/- 1000 ppm 10 = +/- 4000 ppm 11 = +/- 16000 ppm</td>
<td>00</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_SEL_DIVA[1:0]</td>
<td>E[7:6]</td>
<td>2</td>
<td>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.</td>
<td>00</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_SEL_DIVB[1:0]</td>
<td>E[5:4]</td>
<td>2</td>
<td>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.</td>
<td>00</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_SEL_DIVC[1:0]</td>
<td>E[3:2]</td>
<td>2</td>
<td>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.</td>
<td>00</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Register Name</td>
<td>Register Addr (Hex)</td>
<td>Size (Bits)</td>
<td>Description</td>
<td>Default Value</td>
<td>User Access</td>
<td>User Interface Access</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>MC1_SEL_DIVD[1:0]</td>
<td>E[1:0]</td>
<td>2</td>
<td>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.</td>
<td>00</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MC1_ENABLE_CLK[3:0]</td>
<td>10[3:0]</td>
<td>4</td>
<td>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.</td>
<td>0001</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_LOCK_SEL[2:0]</td>
<td>10[6:4]</td>
<td>3</td>
<td>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.</td>
<td>000</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MC1_DYN_SOURCE</td>
<td>10[7]</td>
<td>1</td>
<td>Specify whether the Wishbone or external ports control the dynamic phase settings. 0 = Wishbone registers are in control. 1 = External Ports are in control.</td>
<td>1</td>
<td>Yes</td>
<td>Indirect</td>
</tr>
<tr>
<td>Register Name</td>
<td>Register Addr (Hex)</td>
<td>Size (Bits)</td>
<td>Description</td>
<td>Default Value</td>
<td>User Access</td>
<td>User Interface Access</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>MC1_DIRECTION</td>
<td>11[1]</td>
<td>1</td>
<td>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.</td>
<td>0</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>MC1_ROTATE</td>
<td>11[0]</td>
<td>1</td>
<td>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.)</td>
<td>0</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>MC1_DYN_SEL[1:0]</td>
<td>11[3:2]</td>
<td>2</td>
<td>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</td>
<td>00</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>MC1_TRIMOP_BYPASS_N</td>
<td>11[4]</td>
<td>1</td>
<td>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.</td>
<td>0</td>
<td>Yes</td>
<td>Indirect</td>
</tr>
<tr>
<td>MC1_TRIMOS_BYPASS_N</td>
<td>11[5]</td>
<td>1</td>
<td>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.</td>
<td>0</td>
<td>Yes</td>
<td>Indirect</td>
</tr>
<tr>
<td>MC1_TRIMOS2_BYPASS_N</td>
<td>11[6]</td>
<td>1</td>
<td>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, &amp; 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.</td>
<td>0</td>
<td>Yes</td>
<td>Indirect</td>
</tr>
<tr>
<td>MC1_TRIMOS3_BYPASS_N</td>
<td>11[7]</td>
<td>1</td>
<td>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, &amp; 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.</td>
<td>0</td>
<td>Yes</td>
<td>Indirect</td>
</tr>
</tbody>
</table>
Appendix E. MachXO3L/LF Device Usage with Lattice Diamond Design Software

When using the Lattice Diamond software with the MachXO3L/LF device, there are a few minor differences from the screen shots shown in Figure 17.1 and Figure 17.3 in this technical note.

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 E.1 shows a Diamond example screen for this usage.

![Figure E.1. IPexpress Main Window for PLL Module Using Diamond](image)

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 17.2.

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 E.2 shows a Diamond example screen for this usage.
Figure E.2. Spreadsheet View for Clock Selection Using Diamond
Technical Support Assistance

Submit a technical support case via www.latticesemi.com/techsupport.
# Revision History

## Revision 1.5, January 2022

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A. Primary Clock Sources and Distribution</td>
<td>• Updated Figure A.1. MachXO3L/LF Primary Clock Sources and Distribution to remove PCLKCx_x.&lt;br/&gt;• Added footnote to explain PCLKC pins.</td>
</tr>
</tbody>
</table>

## Revision 1.4, December 2021

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>• Changed document number from TN1282 to FPGA-TN-02058.&lt;br/&gt;• Changed document title to MachXO3 sysCLOCK PLL Design and User Guide.&lt;br/&gt;• Updated document template.</td>
</tr>
<tr>
<td>Disclaimers</td>
<td>Added this section.</td>
</tr>
<tr>
<td>Dynamic Phase Adjustment</td>
<td>Changed min setup to 10 ns in Figure 13.1. GPLL VCO Phase Rotation Timing Diagram.</td>
</tr>
<tr>
<td>Appendix A. Primary Clock Sources and Distribution</td>
<td>Updated Figure A.1. MachXO3L/LF Primary Clock Sources and Distribution to include PCLK.</td>
</tr>
</tbody>
</table>

## Revision 1.3, June 2016

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Added XO3L/LF-9400 information to Table 1.1, Number of PLLs, Edge Clocks, and Clock Dividers.</td>
</tr>
<tr>
<td>Dynamic Phase Adjustment</td>
<td>Added Figure 13.1, GPLL VCO Phase Rotation Timing Diagram including description.</td>
</tr>
</tbody>
</table>

## Revision 1.2, January 2016

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Description</td>
<td>Added Phase Alignment information.&lt;br/&gt;Updated Technical Support Assistance section.</td>
</tr>
<tr>
<td>Configuration Modes</td>
<td>Changed CLKI range in Table 17.1. User Parameters in the IPexpress User Interface.</td>
</tr>
<tr>
<td>PLL Architecture</td>
<td>• Added note.&lt;br/&gt;• Revised Figure D.1. PLL CLKOP Output section.</td>
</tr>
<tr>
<td>Technical Support Assistance</td>
<td>Updated contact information.</td>
</tr>
</tbody>
</table>

## Revision 1.1, March 2015

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Updated PLL Reference Clock Switch (PLLREFCS) section. Added VHDL and Verilog example instantiations.</td>
</tr>
<tr>
<td>Dynamic Phase Adjustment</td>
<td>Product name/trademark adjustment. Included MachXO3LF device.</td>
</tr>
</tbody>
</table>

## Revision 1.0, February 2014

<table>
<thead>
<tr>
<th>Section</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>Initial release</td>
</tr>
</tbody>
</table>