CrossLink-NX sysCLOCK PLL Design and Usage Guide

Technical Note

FPGA-TN-02095-1.0

November 2019
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 and with all faults, and all risk associated with such information is entirely with 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.
## Edge Clock Synchronization (ECLKSSYNC)

### 14.1 ECLKSYNC Component Definition

### 14.2 ECLKSYNC Usage in VHDL

## General Routing for Clocks

## sysCLOCK PLL

### 16.1 Functional Description

- 16.1.1 Refclk (CLKI) Divider
- 16.1.2 Feedback Loop (CLKFB) Divider
- 16.1.3 Output Clock Dividers (CLKO, CLKOS, CLKOS2, CLKOS3)
- 16.1.4 Phase Adjustment (Static Mode)
- 16.1.5 Phase Adjustment (Dynamic Mode)

### 16.2 PLL Features

- 16.2.1 Dedicated PLL Inputs
- 16.2.2 PLL Input Clock Mux (PLLREFCS)
- 16.2.3 Standby Mode (Legacy Mode)

### 16.3 PLL Inputs and Outputs

- 16.3.1 CLKI Input
- 16.3.2 CLKFB Input
- 16.3.3 RST Input
- 16.3.4 Dynamic Clock Enables
- 16.3.5 PLLPD_EN_N Input
- 16.3.6 Dynamic Phase Shift Inputs
- 16.3.7 PHASESEL Input
- 16.3.8 PHASEDIR Input
- 16.3.9 PHASESTEP Input
- 16.3.10 PHASELOADREG Input
- 16.3.11 PLL Clock Outputs
- 16.3.12 LOCK Output

### 16.4 Dynamic Phase Adjustment

- 16.4.1 VCO Phase Shift
- 16.4.2 Divider Phase Shift

### 16.5 Low Power Features

- 16.5.1 Dynamic Clock Enable
- 16.5.2 Standby Mode

### 16.6 PLL Usage in IP Catalog

#### 16.6.1 Configuration Tab

#### 16.6.2 PLL Frequency and Phase Configuration

---

Appendix A: Primary Clock Sources and Distribution

Appendix B: Pinout Rules for Clocking in CrossLink-NX Devices

Technical Support Assistance

Revision History
Figures

Figure 3.1. CrossLink-NX Clocking Structure (LIFCL-40/LIFCL-17) ................................................................. 9
Figure 6.1. CrossLink-NX PLL Block Diagram ........................................................................................................ 11
Figure 7.1. PLL Input Pins for LIFCL-40 ...................................................................................................................... 12
Figure 7.2. PLL Input Structure for LIFCL-17 ............................................................................................................... 12
Figure 8.1. Primary Clock Routing Architecture ........................................................................................................... 14
Figure 9.1. PCLKDIV Component Symbol ..................................................................................................................... 16
Figure 10.1. DCS_COMUX Structure for LIFCL-40 and LIFCL-17 ................................................................................ 18
Figure 10.2. DCS Logic Structure for LIFCL-40 and LIFCL-17 .................................................................................. 18
Figure 10.3. Posedge DCS Switch from sel: 0 => 1 .......................................................................................................... 19
Figure 10.4. Posedge DCS Switch from sel: 1 => 0 .......................................................................................................... 19
Figure 10.5. Negedge DCS Switch from sel: 0 => 1 ......................................................................................................... 20
Figure 10.6. sel_force = 1 DCS Clock Switch Glitches ................................................................................................. 20
Figure 10.7. DCS Component Symbol .......................................................................................................................... 21
Figure 11.1. Glitchless DCC Functional Waveform ...................................................................................................... 23
Figure 11.2. DCCA Component Symbol ........................................................................................................................ 24
Figure 12.1. OSC Component Symbol .......................................................................................................................... 26
Figure 13.1. Edge Clock Sources Per Bank .................................................................................................................... 28
Figure 13.2. ECLKDIV Component Symbol .................................................................................................................. 29
Figure 14.1. ECLKSYNC Component Symbol .............................................................................................................. 31
Figure 14.2. ECLKSYNC Functional Waveform .......................................................................................................... 31
Figure 15.1. Gated Clock to the Primary Clock Routing ............................................................................................. 33
Figure 15.2. Gated Clock to Small Logic Domain ......................................................................................................... 33
Figure 16.1. CrossLink-NX PLL Block Diagram ........................................................................................................... 34
Figure 16.2. PLL Component Instance ........................................................................................................................ 35
Figure 16.3. PLL Dedicated Inputs to the PLLREFCS Component for Top Left and Right PLL ........................................ 37
Figure 16.4. RST Input Timing Diagram ........................................................................................................................ 38
Figure 16.5. PLL Phase Shifting Using the PHASESTEP Signal .................................................................................... 41
Figure 16.6. Divider Phase Shift Timing Diagram ........................................................................................................ 42
Figure 16.7. Dynamic Clock Enable for PLL Outputs .................................................................................................. 43
Figure 16.8. IP Catalog Main Window for PLL Module .................................................................................................. 43
Figure 16.9. CrossLink-NX PLL Frequency Configuration in General Tab ................................................................ 44
Figure 16.10. CrossLink-NX PLL Phase Configuration Tab ........................................................................................... 47
Figure 16.11. CrossLink-NX PLL Optional Ports Configuration Tab .............................................................................. 48
Figure 16.12. CrossLink-NX PLL Analog Settings Tab ................................................................................................. 49
Figure A.1. CrossLink-NX Primary Clock Sources and Distribution, LIFCL-40 Devices ................................................ 51
Figure A.2. CrossLink-NX Primary Clock Sources and Distribution, LIFCL-17 Devices ................................................ 52
Tables

Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers ................................................................. 8
Table 9.1. PCLKDIV Component Port Definition .......................................................................................... 16
Table 9.2. ECLKDIV Component Attribute Definition ............................................................................... 16
Table 10.1. DCS Component Port Definition .............................................................................................. 21
Table 10.2. DCS – DCSMODE Attribute ..................................................................................................... 21
Table 11.1. DCCA Component Port Definition ............................................................................................ 24
Table 12.1. OSC Component Port Definition ............................................................................................... 26
Table 12.2. OSC Component Attribute Definition ......................................................................................... 26
Table 13.1. ECLKDIV Component Port Definition ......................................................................................... 29
Table 13.2. ECLKDIV Component Attribute Definition ............................................................................... 29
Table 14.1. ECLKSYNC Component Port Definition ................................................................................... 31
Table 16.1. PLL Component Port Definition ................................................................................................. 36
Table 16.2. PLL Clock Output Enable Signal List ......................................................................................... 39
Table 16.3. PHASESEL Signal Settings Definition ....................................................................................... 39
Table 16.4. PHASEDIR Signal Settings Definition ........................................................................................ 40
Table 16.5. PLL Clock Outputs and ECLK Connectivity ............................................................................... 40
Table 16.6. Tab 1, General Settings, IP Catalog User Interface ................................................................. 45
Table 16.7. Tab 2, PLL Phase Settings, IP Catalog User Interface ............................................................... 47
Table 16.8. Tab 3, PLL Optional Ports, IP Catalog User Interface ............................................................. 49
Table 16.9. Tab 4, PLL Analog Settings, IP Catalog User Interface ............................................................. 50
Table B.1. Clock Input Selection Table ........................................................................................................ 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>DCC</td>
<td>Dynamic Clock Control</td>
</tr>
<tr>
<td>DCS</td>
<td>Dynamic Clock Select</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase Locked Loop</td>
</tr>
<tr>
<td>SED</td>
<td>Soft Error Detect</td>
</tr>
</tbody>
</table>
1. Introduction

This usage guide describes the clock resources available in the CrossLink-NX™ device architecture. Details are provided for Primary Clocks, Edge Clocks, PLLs, the Internal Oscillator, and clocking elements such as Clock Dividers, Clock Multiplexers, and Clock Stop Blocks available in the CrossLink-NX device. The number of PLLs, Edge Clocks, and Clock Dividers for each device is 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>LIFCL-40</th>
<th>LIFCL-17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of PLLs</td>
<td>General purpose PLLs.</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Number of Edge Clocks</td>
<td>Edge Clocks for high speed applications.</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Number of Edge Clock Dividers</td>
<td>Edge Clock Dividers for DDR applications.</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Number of Primary Clock Dividers</td>
<td>Clock dividers for domain crossing applications.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Number of DDRDLLs</td>
<td>DDRDLL used to DDR memory and High Speed I/O interfaces</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

It is very important to validate device pinout so that correct pin placement is used. The Lattice Radiant® tool should be used to validate the pinout while designing the printed circuit board.

2. Clock/Control Distribution Network

CrossLink-NX devices provide global clock distribution in the form of 32 global primary clocks. These Primary clocks can be divided into 16 clocks per clock domain. However, there is a maximum of 64 unique clock input sources. The CrossLink-NX primary clocking structure is Edge Clock rich and contains generous low-skew Primary clock resources.
3. CrossLink-NX Top-Level View

A top level view of the major clocking resources for the CrossLink-NX devices is shown in Figure 3.1. The shaded blocks (PCIe, upper left PLL, and I/O Bank 2/Bank 6/Bank 7) are not available in the LIFCL-17 device.

![CrossLink-NX Clocking Structure (LIFCL-40/LIFCL-17)](image)

Figure 3.1. CrossLink-NX Clocking Structure (LIFCL-40/LIFCL-17)

4. Clocking Architecture Overview

Below is a brief overview of the clocking structure, elements, and PLL. Greater detail is provided starting with the Appendix A. Primary Clock Sources and Distribution and Appendix B Pinout Rules for Clocking in CrossLink-NX Devices section.

4.1. Primary Clock Network

Up to 32 primary clocks can be selected from up to 64 Primary Clock Sources (PLLs, External Inputs, SERDES, and others) and routed to the Primary Clock Network.

The Primary Clock Network provides low-skew, high fan-out clock distribution to all synchronous elements in the FPGA fabric. The Primary Clock Network is divided into two clocking domains, each domain associated with a DCS_CMUX. Each of these domains has 16 clocks that can be distributed to the fabric in the domain. Initially, the Lattice Radiant software automatically routes each clock domain; up to a maximum of 16 clocks. You can change how the clocks are routed by specifying a preference in the Lattice Radiant software to locate the clock to specific domains.

4.2. Edge Clock Network

Edge Clocks are low skew, high speed clock resources used to clock data into/out of the I/O logic of CrossLink-NX. There are three Edge Clocks per bank located on the bottom side of the device.
5. Overview of Other Clocking Elements

5.1. Edge Clock Dividers (ECLKDIV)
Edge Clock dividers are provided to create the divided down clocks used with the I/O Mux/DeMux gearing logic (SCLK inputs to the DDR) and drives to the Primary Clock routing to the fabric. There are four Clock Dividers on the CrossLink-NX device.

5.2. Primary Clock Divider (PCIKDIV)
Primary Clock Divider are provided to create the divided down clocks.

5.3. Dynamic Clock Select (DCS)
The CrossLink-NX dynamic clock select provides run-time selectable glitchless or non-glitchless operation between two independent clock sources to the primary clock network. This clock select allows the selection of clock sources without leaving the dedicated clock resources in the device. There is one dynamic clock select block on the CrossLink-NX device.

5.4. Dynamic Clock Control (DCC)
Dynamic Clock Control allows dynamic clock enable and disables the MIDMUX Feed Line and the four special CIB clock from the core. When a Feed Line is disabled, all the logic and clock signals that are fed by this Feed Line do not toggle. Hence, it reduces the overall dynamic power consumption of the device.

5.5. Edge Clock Sync (ECLKSYNC)
Each ECLK has a block to allow dynamic synchronization of the Edge Clock. This allows you to start and stop the clock synchronous to an event or external signal. These are important for applications requiring exact clock timing relationships on the inputs, such as DDR memories and video applications.

5.6. Oscillator (OSC)
An internal programmable rate oscillator is provided. The oscillator can be used for FPGA configuration, Soft Error Detect (SED), and as a user logic clock source that is available after FPGA configuration. There is one OSCG on the CrossLink-NX device. The oscillator clock output is routed directly to primary clocking.
The oscillator output is not a high-accuracy clock, having a +/- 15% variation in its output frequency. It is mainly used for circuits that do not require a high degree of clock accuracy. Examples of usage are asynchronous logic blocks such as a timer or reset generator, or other logic that require a constantly running clock.
6. sysCLOCK™ PLL Overview

The sysCLOCK PLLs can be used in a variety of clock management applications such as clock injection removal, clock phase adjustment, clock timing adjustment, and frequency synthesis (multiplication and division of a clock). The PLL supports Fractional-N synthesis. The CrossLink-NX IP Catalog PLL user interface shows important timing parameters such as the VCO rate and the PLL loop bandwidth.

The PLL Input sources are:
- Dedicated PLL Input Pins
- Primary Clock Routing
- Edge Clock Routing
- FPGA Fabric

![CrossLink-NX PLL Block Diagram](image)

Figure 6.1. CrossLink-NX PLL Block Diagram

There are three PLLs on the larger density device LIFCL-40 and two PLLs on the smaller density LIFCL-17 device. There is a PLL on three corners (Upper Left, Lower Left and Lower Right) of the larger density devices and the smaller density device has two PLLs, one each in the Lower Left and Lower Right corners. Each PLL has six outputs. All six PLL outputs can feed the Primary Clock and Edge Clock networks.
7. PLL Features

7.1. Dedicated PLL Inputs
The top PLL, which is available only for LIFCL-40, has the dedicated input pin from the upper left bank. The bottom two PLLs have one pair of dedicated input pin on the bottom banks.

![Figure 7.1. PLL Input Pins for LIFCL-40](image1)

![Figure 7.2. PLL Input Structure for LIFCL-17](image2)

7.2. Input PLL Clock Selection (PLLREFCS)
The PLLREFCS component is a non-glitchless multiplexer that allows you to dynamically select between two PLL input reference clocks. The PLLREFCS has the same input clock sources as the PLL. Since the dedicated PLL inputs are routed to the input of the PLLREFCS components, you can dynamically select between two external reference clock inputs for the top corners of the FPGA. This mux can also be used stand-alone with the PLL in bypass mode for more clock muxing capabilities.

7.3. Clock Injection Delay Removal
The clock injection delay removal feature of the PLL removes the delay associated with the PLL and clock tree. This feature is typically used to reduce clock to out timing and remove the delay differences between the PLL output clock and the data input. This feature is performed by aligning the input clock with a feedback clock from the clock tree. Optional delay may also be added to the feedback path to further reduce the clock injection time.

7.4. Clock Phase Adjustment
The clock phase adjustment feature of the PLL provides the ability to set a specific phase offset between the outputs of the PLL. New to the CrossLink-NX device, phase adjustments can be calculated in much finer increments since the frequency is used to calculate the available phase increments. This feature is detailed further in the Dynamic Phase Adjustment section.

7.5. Frequency Synthesis
The PLL can be used to multiply up or divide down an input clock.
7.6. Additional Features

In addition to the major features, the PLL has several other options that can be used in conjunction with the major modes.

- A Legacy mode to reduce power. The Legacy Mode is called a PLL standby mode. But due to the new proposed schemed for CrossLink-NX PLL, it is called a different name and in order to differentiate the new STDBY mode.
8. Primary Clocks

8.1. Primary Clock Sources

The primary clock network has multiple inputs, called primary clock sources, which can be routed directly to the primary clock routing to clock the FPGA fabric.

The primary clock sources that can get to the primary clock routing are:

- Dedicated Clock Input Pins
- PLL Outputs
- CLKDIV Outputs
- Internal FPGA Fabric Entries (with minimum general routing)
- SERDES/PCS/PCSDIV clocks
- OSC Clock

All potential primary clock sources are multiplexed prior to going to the primary clock routing by a mid-mux. There are 56 mid-mux connections and four FPGA fabric connections, 60 total, routed to a multiplexor in the center of the chip called the centermux. From the centermux, primary clocks are selected and distributed to the FPGA fabric. The maximum number of unique clock sources is 16 bottom mid-mux sources + 12 top mid-mux sources + 14 left midmux sources + 14 right mid-mux sources + 4 direct FPGA fabric entry points (from general routing) = 60. The basic clocking structure is shown in Figure 3.1 and elaborated in Appendix A. Primary Clock Sources and Distribution.

8.2. Primary Clock Routing

The primary clock routing network is made up of low skew clock routing resources with connectivity to every synchronous element of the device. Primary clock sources are selected at the mid-mux, then selected in the centermux and distributed on the primary clock routing to clock the synchronous elements in the FPGA fabric. The primary clock routing network is divided into two sections, left and right, called domains. Figure 8.1 is the simplified view of Figure 3.1.

![Figure 8.1. Primary Clock Routing Architecture](image-url)
The centermux can source up to 16 independent primary clocks per domain which can clock the logic located in that domain. The centermux can also route each clock source to all domains. The Lattice Radiant software automatically routes a primary clock to all four domains in the FPGA.

8.3. Dedicated Clock Inputs

The CrossLink-NX device has dedicated pins, called PCLK pins, to bring an external clock source into the FPGA and allow them to be used as FPGA primary clocks. These inputs route directly to the Primary clock network, or to Edge Clock routing resources. A dedicated PCLK clock pin must always be used to route an external clock source to FPGA logic and I/O.

If an external input clock is being sourced to a PLL, then in most cases, the input clock should use a dedicated PLL input pin. SERDES reference clocks also have dedicated SERDES reference clock pins. The CrossLink-NX device allows a PLL reference clock or a SERDES reference clock to come from an external Primary Clock (PCLK) pin and route through the Primary clock network to drive the reference clock to the SERDES or the input of a PLL.
9. Primary Clock Dividers (PCLKDIV)

There is one Primary Clock Divider available in the CrossLink-NX device, located inside the centermux. The Primary Clock Divider provides the following functionalities:

- PCLK Divider supports x2, x4, x8, x16, x32, x64, and x128. When PCLK divider is bypassed, it is x1 mode. The clock divide ratio is statically controlled by configuration bit mc1_pdivxx.
- PCLK Divider can be reset by global Reset signals and sleep mode control signals. The global reset can be disable by a configuration MC1 bit.
- PCLK Divider supports user LSR through CIB port.
- The reset is Asynchronous assert and synchronous de-assert. The divider output starts at the next cycle after the reset is synchronously released.
- Allow gsrn activity be ignored during device power up by gating this signal with done_cfg.
- When exiting from sleep mode, the retention registers are released from the asynchronous reset control.

9.1. PCLKDIV Component Definition

The PCLKDIV component can be instantiated in the source code of a design as defined in this section. Figure 9.1, Table 9.1, and Table 9.2 define the PCLKDIV component. Verilog and VHDL instantiations are included.

![Figure 9.1. PCLKDIV Component Symbol](image)

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKIN</td>
<td>I</td>
<td>Primary Clock Input</td>
</tr>
<tr>
<td>LSRPDIV</td>
<td>I</td>
<td>Local Reset — Active High, asynchronously forces all outputs low.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSRPDIV = 0 Clock outputs are active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSRPDIV = 1 Clock outputs are OFF</td>
</tr>
<tr>
<td>CLKOUT</td>
<td>O</td>
<td>Divide by 1, 2, 4, 8, 16, 32, 64 or 128 Output Port</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIV_PCLKDIV</td>
<td>X1, X2, X4, X8, X16, X32, X64, X128</td>
<td>X1</td>
<td>Primary Clock Divide Ration Selection</td>
</tr>
<tr>
<td>GSR</td>
<td>ENABLE, DISABLE</td>
<td>ENABLED</td>
<td>GSR ENABLE/DISABLE Selection</td>
</tr>
</tbody>
</table>
9.2. PCLKDIV Usage in VHDL

Component Instantiation

Library lattice;
use lattice.components.all;

Component and Attribute Declaration

component PCLKDIV
Generic (DIV_PCLK : string;
    GSR : string);
Port (CLKIN : in STD_LOGIC;
    LSRPDIV : in STD_LOGIC;
    CLKOUT : out STD_LOGIC);
end component;

PCLKDIV Instantiation

attribute DIV_PCLK : string;
attribute DIV_PCLKDIV of I1 : label is “X1”;
attribute GSR : string;
attribute GSR of I1 : label is “DISABLED”;

I1: PCLKDIV
generic map (DIV_PCLKDIV => "2.0",
    GSR => "DISABLED")
port map (CLKIN => CLKIN,
    LSRPDIV => LSRPDIV,
    CLKOUT => CLKOUT);

9.3. PLKDIVF Usage in Verilog

Component and Attribute Declaration

module PCLKDIV (CLKIN, LSRPDIV, CLKOUT);

parameter DIV_PCLKDIV = “X2”; // “X1”, “X2”, “X4”, “X8”, “X16”, “X32”, “X64”, “X128”
parameter GSR = “DISABLED”; // “ENABLED”, “DISABLED”

input CLKIN, LSRPDIV;
output CLKOUT;
endmodule

PCLKDIV Instantiation

defparam I1.DIV_PCLKDIV = "X2'';
defparam I1.GSR = "DISABLED'';
PCLKDIV I1 {
    .CLKin (CLKIN),
    .LSRPDIV (LSRPDIV),
    .CLKOUT (CLKOUT));
10. Dynamic Clock Select (DCS)

The CrossLink-NX device has one dynamic clock select (DCS) block located at the center of the PLC array core, which can drive to any or all the domains. The DCS_CMUX Structure for LIFCL-40 and LIFCL-17 is shown in Figure 10.1.

![DCS_CMUX Structure for LIFCL-40 and LIFCL-17](image)

The DCS block allows dynamic and glitchless selection between two PCLK clock sources. The DCS block share the same clock resource as any PCLK CMUX. This way any two clocks can perform the DCS function. The inputs to the DCS block come from all the outputs of MIDMUXs and local routing that is located at the center of the PLC array core. The output of the DCS is connected to the inputs of Primary Clock Center MUXs. The DCS logic structure is shown in Figure 10.2.

![DCS Logic Structure for LIFCL-40 and LIFCL-17](image)
For glitchless operation, the `DCSMODE` attribute sets the behavior of the DCS output. The additional attribute values and their functions are shown in Table 10.2.

### 10.1. DCS Timing Diagrams

The DCS block allows dynamic and glitchless selection between two PCLK clock sources. The DCS block share the same clock resource as any PCLK CMUX. This way any two clocks can perform the DCS function. Figure 10.3, Figure 10.4, and Figure 10.5 timing diagrams show the DCS in Glitchless operation in conjunction with the DCSMODE attribute. Figure 10.6 timing diagram shows the Non-Glitchless bypass operation scenario.

#### 10.1.1. Functionality – posedge sel switch

The selection switches from current clock to target clock. For posedge configuration, the latch state is low. Below is the sequence of events once sel toggles:

1. Current clock must see posedge then negedge, then is deactivated.
2. Target clock must see posedge then negedge, then output is successfully switched over.

**Figure 10.3. Posedge DCS Switch from sel: 0 => 1**

**Figure 10.4. Posedge DCS Switch from sel: 1 => 0**
10.1.2. Functionality – negedge sel switch

The selection switches from current clock to target clock. For negedge configuration, the latch state is high. Below is the sequence of events once sel toggles:
1. Current clock must see negedge then posedge, then is deactivated.
2. Target clock must see negedge then posedge, then output is successfully switched over.

![Figure 10.5. Negedge DCS Switch from sel: 0 => 1](image)

10.1.3. Functionality – bypass

When sel_force is high, the switch is in bypass mode. Output clock switching is directly transitions between current and target clock, and may have glitches.

![Figure 10.6. sel_force = 1 DCS Clock Switch Glitches](image)
10.2. DCS Component Definition

The DCS component can be instantiated in the source code of a design as defined in this section.

![DCS Component Symbol](image)

**Figure 10.7. DCS Component Symbol**

<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 0 — Default</td>
</tr>
<tr>
<td>CLK1</td>
<td>I</td>
<td>Clock Input port 1</td>
</tr>
<tr>
<td>SEL</td>
<td>I</td>
<td>Input Clock Select</td>
</tr>
<tr>
<td>SELFORCE</td>
<td>I</td>
<td>Selects Glitchless (0) or Non-Glitchless (1) behavior</td>
</tr>
<tr>
<td>DCSOUT</td>
<td>O</td>
<td>Clock Output Port</td>
</tr>
</tbody>
</table>

**Table 10.1. DCS Component Port Definition**

10.3. DCSMODE Attribute

**Table 10.2** provides the behavior of the DCS output based on the setting of the DCSMODE attribute when the pin MODESEL =0. The MODESEL pin is dynamic and can toggle during operation. **Table 10.2** is only valid when MODESEL=0.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Value</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCSMODE</td>
<td>SEL = 0</td>
<td>SEL = 1</td>
<td></td>
</tr>
<tr>
<td>VCC</td>
<td>Clk0</td>
<td>Clk1</td>
<td>Rising edge triggered. Latched state is high.</td>
</tr>
<tr>
<td>GND</td>
<td>Clk0</td>
<td>Clk1</td>
<td>Falling edge triggered. Latched state is low.</td>
</tr>
<tr>
<td>BUFGCECLK1_0</td>
<td>0</td>
<td>CLK1</td>
<td>SEL is active high. Disabled output is low</td>
</tr>
<tr>
<td>BUFGCECLK1</td>
<td>1</td>
<td>CLK1</td>
<td>SEL is active high. Disabled output is high.</td>
</tr>
<tr>
<td>BUFGCECLK0</td>
<td>CLK0</td>
<td>0</td>
<td>SEL is active low. Disabled output is low.</td>
</tr>
<tr>
<td>BUFGCECLK0_1</td>
<td>CLK0</td>
<td>1</td>
<td>SEL is active low. Disabled output is high.</td>
</tr>
<tr>
<td>BUF0</td>
<td>Clk0</td>
<td>Clk0</td>
<td>Buffer for CLK0</td>
</tr>
<tr>
<td>BUF1</td>
<td>Clk1</td>
<td>Clk1</td>
<td>Buffer for CLK1</td>
</tr>
<tr>
<td>SELFORCE= 1</td>
<td>Non-Glitchless</td>
<td>Clk0</td>
<td>Clk1</td>
</tr>
</tbody>
</table>

**Table 10.2. DCS – DCSMODE Attribute**
10.4. DCS Usage in VHDL

Component Instantiation

Library lattice;
use lattice.components.all;

Component and Attribute Declaration

COMPONENT DCS
    GENERIC(DCSMODE : string := "VCC");
    PORT (CLK0 :IN STD_LOGIC;
           CLK1 :IN STD_LOGIC;
           SEL :IN STD_LOGIC;
           SELFORCE :IN STD_LOGIC;
           DCSOUT :OUT STD_LOGIC);
    END COMPONENT;

DCS Instantiation

attribute DCSMODE : string;
attribute DCSMODE of DCSinst0 : label is "VCC";
I1: DCS
genmap(DCSMODE => "VCC")
portmap(
    CLK0 => CLK0,
    CLK1 => CLK1,
    SEL => SEL,
    SELFORCE => SELFORCE,
    DCSOUT => DCSOUT);

10.5. DCS Usage in Verilog

Component and Attribute Declaration

module DCS(CLK0,CLK1,SEL,SELFORCE,DCSOUT);
    input   CLK0;
    input   CLK1;
    input   SEL;
    input   SELFORCE;
    output  DCSOUT;
endmodule

DCS Instantiation

defparam DCSInst0.DCSMODE = "VCC";
DCS DCSInst0 (.
   CLK0   (CLK0),
   CLK1   (CLK1),
   SEL    (SEL),
   SELFORCE (SELFORCE),
   DCSOUT (DCSOUT));
11. Dynamic Clock Control (DCC)

The CrossLink-NX device has a Dynamic Clock Control feature which allows internal logic dynamically enable or disable domain primary clock network. The disable function does not create glitch and increases the clock latency to the primary clock network. Also, this dynamic clock control function can be disabled by a configuration memory fuse to always enable the primary clock network.

This DCC controls the clock sources from the Primary CLOCK MIDMUX before they are fed to the Primary Center MUXs that drive the domain clock network. When a clock network is disabled, all the logic fed by that clock does not toggle, hence, reducing the overall power consumption of the device.

The CrossLink-NX device clock architecture allows both DCC and DCS to function at the same. It should be noted that the clock source used as feedback into the GPLL should always keep enable signal high in the DCC. Otherwise, it leads to loss of lock for the GPLL when toggling the enable signal.

![Glitchless DCC Functional Waveform](image)

**Figure 11.1. Glitchless DCC Functional Waveform**

11.1. DCC

Dynamic Clock Control allows the four clock from the FPGA fabric feeding to the MIDMUX be dynamically enabled and disabled. When a Feed Line is disabled, all the logic and clock signals that are fed by this Feed Line do not toggle. Hence, it reduces the overall dynamic power.
11.2. Component Definition

The DCCA component can be instantiated in the source code of a design as defined in this section. Figure 11.2 and Table 11.1 show the DCCA definitions.

![DCCA Component Symbol](image)

**Figure 11.2. DCCA Component 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 port.</td>
</tr>
<tr>
<td>CE</td>
<td>I</td>
<td>Clock Enable port — CE = 0 CLKO is disabled (CLKO = ‘0’) — CE = 1 CLKO is enabled (CLKO = CLKI)</td>
</tr>
<tr>
<td>CLKO</td>
<td>O</td>
<td>Clock Output Port</td>
</tr>
</tbody>
</table>

11.3. DCC Usage in VHDL

**Component Instantiation**

```vhdl
library lattice;
use lattice.components.all;
Component and Attribute Declaration
COMPONENT DCC
PORT  (CLKI :IN STD_LOGIC;
         CE  :IN STD_LOGIC;
         CLKO :OUT STD_LOGIC);
END COMPONENT;

DCCA Instantiation
I1: DCC
port map (CLKI => CLKI,
         CE  => CE,
         CLKO => CLKO);
```

**DCC Usage in Verilog**

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

DCC DCSInst0 (
.CLKI (CLKI),
.CE (CE),
.CLKO (CLKO));
12. Internal Oscillator (OSC)

The OSC element performs multiple functions on the CrossLink-NX device. It is used for configuration, SED, as well as optionally in user mode. In user mode, the OSC element has the following features:

- It permits a design to be fully self-clocked, as long as the quality of the OSC element’s silicon-based oscillator is adequate.
- If it is unused, it can be turned off for power savings.
- It has an input to dynamically control standby/normal operation.
- It has a direct connection to primary clock routing through the left mid-mux.
- It can be configured for operation at a wide range of frequencies via configuration bits.

12.1. OSC Component Definition

The OSC component can be instantiated in the source code of a design as defined in this section. Figure 12.1 and Table 12.1 below show the OSC definitions.

 OSC

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFSDSCEN</td>
<td>I</td>
<td>HF user clock output enable</td>
</tr>
<tr>
<td>HFCLKOUT</td>
<td>O</td>
<td>450 MHz with programmable divider (2~256) to user</td>
</tr>
<tr>
<td>HFSDCOUT</td>
<td>O</td>
<td>450 MHz with programmable divider (2~256) to user for SED/SEC application</td>
</tr>
<tr>
<td>I2CCKOUT</td>
<td>O</td>
<td>128 kHz Clock output for I²C from LF oscillator</td>
</tr>
<tr>
<td>LFCLKOUT</td>
<td>O</td>
<td>Low frequency clock output; 32 kHz</td>
</tr>
</tbody>
</table>

**Figure 12.1. OSC Component Symbol**

**Table 12.1. OSC Component Port Definition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTR_EN</td>
<td>DISABLED</td>
<td>DISABLED</td>
<td>DTR block enable from MIB</td>
</tr>
<tr>
<td>HF_CLK_DIV</td>
<td>00000001</td>
<td>00000000</td>
<td>User assignable HF oscillator output divider configuration (div2~div256)</td>
</tr>
<tr>
<td>HF_SED_SEC_DIV</td>
<td>00000001</td>
<td>00000000</td>
<td>User assignable HF oscillator output divider configuration (div2~div256)</td>
</tr>
<tr>
<td>HF_FABRIC_EN</td>
<td>DISABLED</td>
<td>DISABLED</td>
<td>High frequency oscillator trim source mux select</td>
</tr>
<tr>
<td>HF_OSC_EN</td>
<td>DISABLED</td>
<td>DISABLED</td>
<td>HF oscillator enable, controlled by the user</td>
</tr>
<tr>
<td>HF_TRIM_NV</td>
<td>00000000</td>
<td>00000000</td>
<td>High frequency oscillator non-volatile trim interface port</td>
</tr>
</tbody>
</table>

**Table 12.2. OSC Component Attribute Definition**
### Name | Value | Default | Description
--- | --- | --- | ---
HFDIV_FABRIC_EN | DISABLED | DISABLED | High frequency oscillator divider configuration mux select, fabric divider enable
LF_FABRIC_EN | DISABLED | DISABLED | Low frequency oscillator trim source mux select, fabric driven trim enable
LF_OUTPUT_EN | DISABLED | DISABLED | Low frequency clock output enable
LF_TRIM_NV | 000000000 | 000000000 | Low frequency oscillator non-volatile trim interface port

### 12.2. OSC Usage in VHDL

#### Component Instantiation

Library lattice;
use lattice.components.all;

#### Component and Attribute Declaration

```vhdl
component OSC
Port
(HFSDSCEN :IN STD_LOGIC;
 HFCLKOUT :out STD_LOGIC;
 HFSDCOUT :out STD LOGIC;
 I2CCKOUT :out STD LOGIC;
 LFCLKOUT :out STD LOGIC);
end component;
```

#### OSCG Instantiation

I1: OSC
port map (HFSDSCSCEN => HFSDSCSCEN,
 HFCLKOUT => HFCLKOUT,
 HFSDCOUT => HFSDCOUT,
 I2CCKOUT => I2CCKOUT,
 LFCLKOUT => LFCLKOUT);

### 12.3. OSC Usage in Verilog

#### Component and Attribute Declaration

```verilog
module OSC (HFSDSCSCEN,HFCLKOUT,HFSDCOUT,I2CCKOUT,LFCLKOUT);
input HFSDSCSCEN;
output HFCLKOUT;
output HFSDCOUT;
output I2CCKOUT;
output LFCLKOUT;
endmodule
```

#### OSC Instantiation

OSC I1 (.HFSDSCSCEN (HFSDSCSCEN),
 .HFCLKOUT (HFCLKOUT),
 .HFSDCOUT (HFSDCOUT),
 .I2CCKOUT (I2CCKOUT),
 .LFCLKOUT (LFCLKOUT));
13. Edge Clocks

Each CrossLink-NX device bottom I/O bank has four ECLK resources. There are three I/O banks at the bottom 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. See Figure 3.1 for ECLK locations and connectivity.

The sources of Edge Clocks are:
- Dedicated Clock (PCLK) pins
- DLLDEL output
- PLL outputs (CLKOP and CLKOS)
- ECLK Bridge
- Internal nodes

The CrossLink-NX device has Edge Clock (ECLK) at the bottom of the device. There are four ECLK network per bank I/O. ECLK Input MUX collects all clock sources available as shown in Figure 13.1. There are three ECLK Input MUXs, one for each I/O bank on the bottom side of the device. Each of these MUX generates total of four ECLK Clock sources for each I/O bank. Each ECLK network from one I/O bank can be bridged to another I/O bank from a wider bus if it is needed.

13.1. Edge Clock Dividers (ECLKDIV)

There are twelve Edge Clock dividers available in the CrossLink-NX device, four for each I/O bank at the bottom of the device. The Clock Divider provides a single divided output with available divide values of 2, 3.5, 4, or 5. The inputs to the Clock Dividers are the Edge Clocks, PLL outputs and Primary Clock Input pins. The outputs of the Clock Divider drive the primary clock network and are mainly used for DDR I/O domain crossing.

13.2. ECLKDIV Component Definition

The ECLKDIV component can be instantiated in the source code of a design as defined in this section. Figure 13.2, Table 13.1, and Table 13.2 define the ECLKDIV component. Verilog and VHDL instantiations are included.
Figure 13.2. ECLKDIV Component Symbol

Table 13.1. ECLKDIV Component Port Definition

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECLKIN</td>
<td>I</td>
<td>Edge Clock Input</td>
</tr>
<tr>
<td>DIVRST</td>
<td>I</td>
<td>Reset input — Active High, asynchronously forces all outputs low. DIVRST = 0 Clock outputs are active, DIVRST = 1 Clock outputs are OFF</td>
</tr>
<tr>
<td>SLIP</td>
<td>I</td>
<td>Signal is used for word alignment. When enabled it slips the output one cycle relative to the input clock.</td>
</tr>
<tr>
<td>DIVOUT</td>
<td>O</td>
<td>Divide by 1, 2, 3.5, 4, or 5 Output Port</td>
</tr>
</tbody>
</table>

Table 13.2. ECLKDIV Component Attribute Definition

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSRN_ECLK</td>
<td>ENABLE</td>
<td>ENABLED</td>
<td>GSR ENABLE/DISABLE Selection</td>
</tr>
<tr>
<td></td>
<td>DISABLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECLK_DIV</td>
<td>1</td>
<td>1</td>
<td>ECLK DIVIDE Ratio selection</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3P5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The SLIP input is intended for use with high-speed data interfaces such as DDR or 7:1 LVDS Video.

13.3. ECLKDIV Usage in VHDL

Component Instantiation

```
Library lattice;
use lattice.components.all;
```

Component and Attribute Declaration

```
component ECLKDIV
Generic (ECLK_DIV : string;
         GSRN_ECLK : string);
Port (DIVRST : in STD_LOGIC;
      ECLKIN : in STD_LOGIC;
      SLIP : in STD_LOGIC;
      DIVOUT : out STD_LOGIC);
end component;
```
ECLKDIV Instantiation

attribute ECLK_DIV : string;
attribute ECLK_DIV of I1 : label is "2.0";
attribute GSRN_ECLK : string;
attribute GSRN_ECLK of I1 : label is "DISABLED";

I1: ECLKDIV
generic map (ECLK_DIV => "2.0",
            GSRN_ECLK => "DISABLED")
port map (DIVRST => DIVRST,
          ECLKIN => ECLKIN,
          SLIP => SLIP,
          DIVOUT => DIVOUT);

13.4. CLKDIVF Usage in Verilog

Component and Attribute Declaration

module ECLKDIV (DIVRST, ECLKIN, SLIP, DIVOUT);

parameter ECLK_DIV = "2.0";          // "2.0", "3.5"
parameter GSRN_ECLK = "DISABLED";   // "ENABLED", "DISABLED"

input DIVRST, ECLKIN, SLIP;
output DIVOUT;
endmodule

CLKDIVF Instantiation

defparam I1.ECLK_DIV = "2.0";
defparam I1.GSRN_ECLK = "DISABLED";
ECLKDIV I1 (.
          .DIVRST (DIVRST),
          .ECLKIN (ECLKIN),
          .SLIP (SLIP),
          .DIVOUT (DIVOUT));
14. Edge Clock Synchronization (ECLKSYNC)

CrossLink-NX devices have a dynamic Edge Clock synchronization control (ECLKSYNC) which allows each Edge Clock to be disabled or enabled glitchlessly from core logic if desired. This allows you to synchronize the Edge Clock to an event or external signal if desired. It also allows the design to dynamically disable a clock and its associated logic in the design when it is not needed and thus save power. Applications such as DDR2, DDR3, and 7:1 LVDS for display use this component for clock synchronization.

14.1. ECLKSYNC Component Definition

The ECLKSYNC component can be instantiated in the source code of a design as defined in this section. Asserting the STOP control signal has the ability to stop the Edge Clock to synchronize the signals derived from ECLK and used in high-speed DDR mode applications such as DDR memory, generic DDR, and 7:1 LVDS.

Control signal STOP is synchronized with ECLK when asserted. When control signal STOP is asserted, the clock output is forced to low after the fourth falling edge of the input ECLKI. When the STOP signal is released, the clock output starts to toggle at the fourth rising edge of the input ECLKI clock.

Figure 14.1 and Table 14.1 show the ECLKSYNC component definition.

![ECLKSYNC Component Symbol](image)

Table 14.1. ECLKSYNC Component Port Definition

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECLKIN</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>ECLKOUT</td>
<td>O</td>
<td>Clock Output Port</td>
</tr>
</tbody>
</table>

![ECLKSYNC Functional Waveform](image)
14.2. ECLKSYNC Usage in VHDL

Component Instantiation

Library lattice;
use lattice.components.all;

Component and Attribute Declaration

COMPONENT ECLKSYNC
PORT
  ECLKIN :IN STD_LOGIC;
  STOP  :IN STD_LOGIC;
  ECLKOUT :OUT STD_LOGIC);
END COMPONENT;

ECLKSYNC Instantiation

I1: ECLKSYNC
port map (
  ECLKIN => ECLKIN,
  STOP => STOP,
  ECLKOUT => ECLKOUT);

ECLKSYNC Usage in Verilog

Component and Attribute Declaration

module ECLKSYNC (ECLKIN,STOP,ECLKOUT);
input  ECLKIN;
input  STOP;
output  ECLKOUT;
endmodule

ECLKSYNCB Instantiation

ECLKSYNC ECLKSYNCInst0 (.
  .ECLKIN (ECLKIN),
  .STOP (STOP),
  .ECLKOUT (ECLKOUT));
15. General Routing for Clocks

The CrossLink-NX device architecture supports the ability to use data routing or general routing for a clock. This capability is intended to be used for small areas of the design to allow additional flexibility in linking dedicated clocking resources and building very small clock trees. General routing cannot be used for Edge Clocks for applications that use the DDR registers in the I/O components of the FPGA.

Software limits the distance of a general routing based (gated) clock to one PLC in distance to a primary clock entry point. If the software cannot place the clock gating logic close enough to a primary clock entry point, the error below occurs:

- **ERROR-par** – Unable to reach a primary clock entry point for general route clock <net> in the minimum required distance of one PLC.

There are multiple entry points to the Primary clock routing throughout the CrossLink-NX device fabric. In this case, it is recommended to add a preference for this gated clock to use primary routing.

![Figure 15.1. Gated Clock to the Primary Clock Routing](image)

For a very small clock domain, you can limit the distance of a general routing based (gated) clock to one PLC in distance to the logic it clocks. You must group this logic (UGROUP) with a **BBOX = 1, 1** (see Lattice Radiant Help > Constraints Reference Guide > Preferences > UGROUP) as well as specify a **PROHIBIT PRIMARY** on the generated clock. If the software cannot place the logic tree within the BBOX, an error occurs.

![Figure 15.2. Gated Clock to Small Logic Domain](image)
16. sysCLOCK PLL

The CrossLink-NX PLL provides features such as clock injection delay removal, frequency synthesis, and phase adjustment. Figure 16.1 shows a block diagram of the CrossLink-NX PLL.

![CrossLink-NX PLL Block Diagram](image)

**Figure 16.1. CrossLink-NX PLL Block Diagram**
Figure 16.2. PLL Component Instance
### Table 16.1. PLL Component Port Definition

<table>
<thead>
<tr>
<th>Signal</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[2:0]</td>
<td>I</td>
<td>Select the output affected by Dynamic Phase adjustment.</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 adjustment step.</td>
</tr>
<tr>
<td>PHASELOADREG</td>
<td>I</td>
<td>Load dynamic phase adjustment values into PLL.</td>
</tr>
<tr>
<td>PLLPD_EN_N</td>
<td>I</td>
<td>Standby signal to power down the PLL.</td>
</tr>
<tr>
<td>LEGACY</td>
<td>I</td>
<td>Power mode setting to enable legacy mode</td>
</tr>
<tr>
<td>RST</td>
<td>I</td>
<td>Resets the whole PLL.</td>
</tr>
<tr>
<td>ENCLKOP</td>
<td>I</td>
<td>Enable PLL output CLKOP.</td>
</tr>
<tr>
<td>ENCLKOS</td>
<td>I</td>
<td>Enable PLL output CLKOS.</td>
</tr>
<tr>
<td>ENCLKOS2</td>
<td>I</td>
<td>Enable PLL output CLKOS2.</td>
</tr>
<tr>
<td>ENCLKOS3</td>
<td>I</td>
<td>Enable PLL output CLKOS3.</td>
</tr>
<tr>
<td>ENCLKOS4</td>
<td>I</td>
<td>Enable PLL output CLKOS4.</td>
</tr>
<tr>
<td>ENCLKOS5</td>
<td>I</td>
<td>Enable PLL output CLKOS5.</td>
</tr>
<tr>
<td>CLKOP</td>
<td>O</td>
<td>PLL main output clock.</td>
</tr>
<tr>
<td>CLKOS</td>
<td>O</td>
<td>PLL output clock.</td>
</tr>
<tr>
<td>CLKOS2</td>
<td>O</td>
<td>PLL output clock2.</td>
</tr>
<tr>
<td>CLKOS3</td>
<td>O</td>
<td>PLL output clock3.</td>
</tr>
<tr>
<td>CLKOS4</td>
<td>O</td>
<td>PLL output clock4.</td>
</tr>
<tr>
<td>CLKOS5</td>
<td>O</td>
<td>PLL output clock5.</td>
</tr>
<tr>
<td>LOCK</td>
<td>O</td>
<td>PLL LOCK to CLKI, Asynchronous signal. Active high indicates PLL lock.</td>
</tr>
<tr>
<td>Refclk</td>
<td>O</td>
<td>Output of Reference clock.</td>
</tr>
<tr>
<td>LMMI_CLK</td>
<td>I</td>
<td>CIB LMMI interface clock</td>
</tr>
<tr>
<td>LMMI_OFFSET[6:0]</td>
<td>I</td>
<td>CIB LMMI interface address offset (LSB of address bus)</td>
</tr>
<tr>
<td>LMMI_REQUEST</td>
<td>I</td>
<td>CIB LMMI interface request signal</td>
</tr>
<tr>
<td>LMMI_RESETN</td>
<td>I</td>
<td>CIB LMMI interface reset, active low</td>
</tr>
<tr>
<td>LMMI_WDATA[7:0]</td>
<td>I</td>
<td>CIB LMMI interface write data</td>
</tr>
<tr>
<td>LMMI_WR_RDN</td>
<td>I</td>
<td>CIB LMMI interface Write/Read control; 1=write, 0=read.</td>
</tr>
<tr>
<td>LMMI_RDATA[7:0]</td>
<td>O</td>
<td>CIB LMMI interface read data</td>
</tr>
<tr>
<td>LMMI_RDATA_VALID</td>
<td>O</td>
<td>CIB LMMI interface read data valid signal</td>
</tr>
<tr>
<td>LMMI_READY</td>
<td>O</td>
<td>CIB LMMI interface ready signal</td>
</tr>
</tbody>
</table>

### 16.1. Functional Description

#### 16.1.1. Refclk (CLKI) Divider

The CLKI divider is used to control the input clock frequency into the PLL block. The valid input frequency range is specified in the device data sheet.

#### 16.1.2. Feedback Loop (CLKFB) Divider

The CLKFB divider is used to divide the feedback signal, effectively multiplying the output clock. The VCO block increases the output frequency until the divided feedback frequency equals the input frequency. The output of the feedback divider must be within the phase detector frequency range specified in the device data sheet. This port is only available to user interface when user clock option is selected for feedback clock. Otherwise, this port is connected by the tool to the appropriate signal you selected in the software.
16.1.3. Output Clock Dividers (CLKOP, CLKOS, CLKOS2, CLKOS3)
The output Clock Dividers allow the VCO frequency to be scaled up to the maximum range to minimize 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.

16.1.4. Phase Adjustment (Static Mode)
The CLKOP, CLKOS, CLKOS2, CLKOS3, CLKOS4, and CLKOS5 outputs can be phase adjusted relative to the enabled unshifted output clock. New to the CrossLink-NX device, phase adjustments are now calculated values in the software tools based on VCO clock frequency. This provides a finer phase shift depending on the required frequency. The clock output selected as the feedback cannot use the static phase adjustment feature since it causes the PLL to unlock.

16.1.5. Phase Adjustment (Dynamic Mode)
The phase adjustments can also be controlled in a dynamic mode using the PHASESEL, PHASEDIR, PHASESTEP, and PHASELOADREG ports. The clock output selected as the feedback should not use the dynamic phase adjustment feature. See the Dynamic Phase Adjustment section for usage details. The clock output selected as the feedback cannot use the dynamic phase adjustment feature since it causes the PLL to unlock.

Similar restrictions apply to other clocks.

16.2. PLL Features

16.2.1. Dedicated PLL Inputs
Every PLL has a dedicated low skew input that routes directly to its reference clock input. These are the recommended inputs for a PLL. It is possible to route a PLL input from the Primary clock routing, but it incurs more clock input injection delays, which are not natively compensated for using feedback, compared to a dedicated PLL input. There is one PLLs in each corner of the FPGA on bigger densities. Each PLL on the CrossLink-NX has one pair dedicated PLL input pin.

16.2.2. PLL Input Clock Mux (PLLREFCS)
Each CrossLink-NX PLL contains an input mux to dynamically switch between two input reference clocks. The output of the PLLREFCS is routed directly into the PLL. There is one PLLREFCS component in each top left and right corner of the FPGA. In order to enhance the clock muxing capability of the CrossLink-NX device, the dedicated clock inputs for the PLLs in each corner are routed to the PLLREFCS component in a corner along with the other potential PLL sources such as Edge Clocks and primary clocks. This structure is seen in Figure 16.3.

![Figure 16.3. PLL Dedicated Inputs to the PLLREFCS Component for Top Left and Right PLL](image)

This adds a lot of flexibility for designs that need to switch between two external clocks.
16.2.3. Standby Mode (Legacy Mode)

The CrossLink-NX device 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. Standby mode is very similar to holding the PLL in reset since the VCO is turned off and needs to regain lock when exiting standby. In both cases, reset and standby mode, the PLL retains its programming.

You MUST hold the PLL in standby for a minimum of 1 ms in order to be sure the PLL analog circuits are fully reset and analog startup is stable.

16.3. PLL Inputs and Outputs

16.3.1. CLKI Input

The CLKI signal is the reference clock for the PLL. It must conform to the specifications in the data sheet for the PLL to operate correctly. The CLKI signal can come from a dedicated PLL input pin or from internal 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 can be divided by the input (M) divider to create one input to the phase detector of the PLL.

16.3.2. CLKFB Input

The CLKFB signal is the feedback signal to the PLL. The feedback signal is used by the Phase Frequency Detector inside the PLL to determine if the output clock needs adjustment to maintain the correct frequency and phase. The CLKFB signal can come from a primary clock net (feedback mode = CLKO[P/S2/S3]) to remove the primary clock routing injection delay, from a dedicated external dual-purpose I/O pin (feedback mode = UserClock) to account for board level clock alignment, or from an internal PLL connection (feedback mode = INT_O[P/S2/S3]) for simple feedback. The feedback clock signal is divided by the feedback (N) divider to create an input to the VCO of the PLL. A bypassed PLL output cannot be used as the feedback signal.

16.3.3. RST Input

At power-up, an internal power-up reset signal from the configuration block resets the PLL. At runtime, an active high, asynchronous, user-controlled PLL reset signal 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 cause the outputs to be logic 0. In bypass mode, the output does not reset.

After the RST signal is deasserted, the PLL starts the lock-in process and takes tLOCK time, about 16 ms, to complete PLL lock. Figure 16.4 shows the timing diagram of the RST input. The RST signal is active high. The RST signal is optional. Trst = 1 ms reset pulse width, Trstrec = 1 ns time after a reset before the divider output starts counting again.

---

**Figure 16.4. RST Input Timing Diagram**
16.3.4. Dynamic Clock Enables

Each PLL output has a user input signal to dynamically enable/disable its output clock glitchlessly. When the clock enable signal is set to logic 0, the corresponding output clock is held to logic 0.

Table 16.2. PLL Clock Output Enable Signal List

<table>
<thead>
<tr>
<th>Clock Enable Signal Name</th>
<th>Corresponding PLL Output</th>
<th>IP Catalog Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCLKOP</td>
<td>CLKOP</td>
<td>“Clock Enable OP”</td>
</tr>
<tr>
<td>ENCLKOS</td>
<td>CLKOS</td>
<td>“Clock Enable OS”</td>
</tr>
<tr>
<td>ENCLKOS2</td>
<td>CLKOS2</td>
<td>“Clock Enable OS2”</td>
</tr>
<tr>
<td>ENCLKOS3</td>
<td>CLKOS3</td>
<td>“Clock Enable OS3”</td>
</tr>
<tr>
<td>ENCLKOS4</td>
<td>CLKOS4</td>
<td>“Clock Enable OS4”</td>
</tr>
<tr>
<td>ENCLKOS5</td>
<td>CLKOS5</td>
<td>“Clock Enable OS5”</td>
</tr>
</tbody>
</table>

The Dynamic Clock Enable function allows you to save power by stopping the corresponding output clock when not in use. The clock enable signals are optional and are only available if you select the corresponding option in IP Catalog Wizard. If a clock enable signal is not requested, its corresponding output is active at all times when the PLL is instantiated unless the PLL is placed into standby mode. You cannot access a clock enable signal in IP Catalog Wizard when using it for external feedback to avoid shutting off the feedback clock input.

16.3.5. PLLPD_EN_N Input

The PLLPD_EN_N signal is used to put the PLL into a low power standby mode when it is not required. The PLLPD_EN_N signal is optional and is only available if you select the Enable Powerdown Mode in the IP Catalog wizard. The PLLPD_EN_N signal is active low. When asserted, the PLL outputs are pulled to 0 and the PLL is reset. You need to stay in the Power Down mode for at least 1 ms to make sure the PLL analog circuits are fully reset and to have a stable analog startup.

16.3.6. Dynamic Phase Shift Inputs

The CrossLink-NX PLL has five ports to allow for dynamic phase adjustment from FPGA logic. The Dynamic Phase Adjustment section elaborates on how you should drive these ports.

16.3.7. PHASESEL Input

The PHASESEL[2:0] inputs are used to specify which PLL output port is affected by the dynamic phase adjustment ports. The settings available are shown in the Dynamic Phase Adjustment section. The PHASESEL signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed. The PHASESEL signal is optional and is available if you select the Dynamic Phase Ports option in IP Catalog Wizard.

Table 16.3. PHASESEL Signal Settings Definition

<table>
<thead>
<tr>
<th>PHASESEL[1:0]</th>
<th>PLL Output Shifted</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>CLKOS</td>
</tr>
<tr>
<td>001</td>
<td>CLKOS2</td>
</tr>
<tr>
<td>010</td>
<td>CLKOS3</td>
</tr>
<tr>
<td>011</td>
<td>CLKOS4</td>
</tr>
<tr>
<td>100</td>
<td>CLKOS5</td>
</tr>
<tr>
<td>101</td>
<td>CLKOP</td>
</tr>
</tbody>
</table>
16.3.8. PHASEDIR Input
The PHASEDIR input is used to specify which direction the dynamic phase shift occurs, advanced (leading) or delayed (lagging). When PHASEDIR = 0, then the phase shift is delayed. When PHASEDIR = 1, then the phase shift is advanced. The PHASEDIR signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed. The PHASEDIR signal is optional and is available if you select the Dynamic Phase ports option in IP Catalog Wizard.

<table>
<thead>
<tr>
<th>PHASEDIR</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Delayed (lagging)</td>
</tr>
<tr>
<td>1</td>
<td>Advanced (leading)</td>
</tr>
</tbody>
</table>

16.3.9. PHASESTEP Input
The PHASESTEP signal is used to initiate a VCO dynamic phase shift for the clock output port and in the direction specified by the PHASESEL and PHASEDIR inputs. This phase adjustment is done by changing the phase of the VCO in 45° increments. The VCO phase changes on the negative edge of the PHASESTEP input after four VCO cycles. This is an active low signal and the minimum pulse width (both high and low) of PHASESTEP pulse is four cycles of VCO running period. The PHASESTEP signal is optional and is available if you select the Dynamic Phase ports option in IP Catalog Wizard. The PHASESEL and PHASEDIR are required to have a setup time of 5 ns prior to PHASESTEP falling edge.

16.3.10. PHASELOADREG Input
The PHASELOADREG signal is used to initiate a post-divider dynamic phase shift, relative to the unshifted output, for the clock output port and in the direction specified by the PHASESEL and PHASEDIR inputs. A phase shift is started on the falling edge of the PHASELOADREG signal and there is a minimum pulse width of 10 ns from assertion to desertion. The PHASESEL and PHASEDIR are required to have a setup time of 5 ns prior to PHASELOADREG falling edge. The PHASELOADREG signal is optional and is available if you select the Dynamic Phase ports option in IP Catalog Wizard.

16.3.11. PLL Clock Outputs
The PLL has four outputs, listed in Table 16.5. All four outputs can be routed to the Primary clock routing of the FPGA. All four outputs can be phase shifted statically or dynamically if external feedback on the clock is not used. They can also statically or dynamically adjust their output duty cycle. The outputs can come from their output divider or the reference clock input (PLL bypass). In bypass mode, the output divider can be bypassed or used to divide the reference clock.

<table>
<thead>
<tr>
<th>Clock Output Name</th>
<th>Edge Clock Connectivity</th>
<th>Selectable Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKOP</td>
<td>No ECLK Connection</td>
<td>Always Enabled</td>
</tr>
<tr>
<td>CLKOS</td>
<td>No ECLK Connection</td>
<td>Selectable via IP Catalog</td>
</tr>
<tr>
<td>CLKOS2</td>
<td>No ECLK Connection</td>
<td>Selectable via IP Catalog</td>
</tr>
<tr>
<td>CLKOS3</td>
<td>No ECLK Connection</td>
<td>Selectable via IP Catalog</td>
</tr>
<tr>
<td>CLKOS4</td>
<td>No ECLK Connection</td>
<td>Selectable via IP Catalog</td>
</tr>
<tr>
<td>CLKOS5</td>
<td>No ECLK Connection</td>
<td>Selectable via IP Catalog</td>
</tr>
</tbody>
</table>
16.3.12. 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 achieves lock within 16 ms. Once lock is achieved, the PLL LOCK signal is asserted. The LOCK signal can be set in IP Catalog Wizard in either the default unsticky frequency lock mode by checking the Provide PLL Lock Signal or sticky lock mode by selecting PLL Lock is Sticky. In sticky lock mode, once the LOCK signal is asserted (logic 1), it stays asserted until a PLL reset is asserted. In the default lock mode of unsticky frequency lock, if during operation the input clock or feedback signals to the PLL become invalid, the PLL loses lock and the LOCK output de-asserts (logic 0). 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 is available if you select the Provide PLL Lock signal option in IP Catalog Wizard.

16.4. Dynamic Phase Adjustment

Dynamic phase adjustment of the PLL output clocks can be affected without reconfiguring the FPGA by using the dedicated dynamic phase-shift ports of the PLL.

All six output clocks, CLKOP, CLKOS, CLKOS2, CLKOS3, CLKOS4, and CLKOS5 have the dynamic phase adjustment feature but only one output clock can be adjusted at a time. Table 16.5 shows the output clock selection settings available for the PHASESEL[1:0] signal. The PHASESEL signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed.

The selected output clock phase is either advanced or delayed depending upon the value of the PHASEDIR port or signal. Table 16.4 shows the PHASEDIR settings available. The PHASEDIR signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed.

16.4.1. VCO Phase Shift

Once the PHASESEL and PHASEDIR have been set, a VCO phase adjustment is made by toggling the PHASESTEP signal from the current setting. Each pulse of the PHASESTEP signal generates a phase step based on this equation:

\[(\text{CLKO}<n>_\_\text{FPHASE}/ (8*\text{CLKO}<n>_\_\text{DIV})) * 360\]

Where \(<n>\) is the clock output specified by PHASESEL (CLKOP/OS/OS2/OS3). Values for \(\text{CLKO}<n>_\_\text{FPHASE}\) and \(\text{CLKO}<n>_\_\text{DIV}\) are located in the HDL source file.

The PHASESTEP signal is latched in on the falling edge and is subject to a minimum wait of four VCO cycles prior to pulsing the signal again. One step size is the smallest phase shift that can be generated by the PLL in one pulse. The dynamic phase adjustment results in a glitch free adjustment when delaying the output clock, but glitches may result when advancing the output clock.

![Figure 16.5. PLL Phase Shifting Using the PHASESTEP Signal](image-url)
For Example:
PHASESEL[2:0]=3'b000 to select CLKOS for phase shift
PHASEDIR =1'b0 for selecting delayed (lagging) phase
Assume the output is divided by 2, CLKOS_DIV = 2
The CLKOS_FPHASE is set to 1.
The above signals need to be stable for 5 ns before the falling edge of PHASESTEP and the minimum pulse width of PHASESTEP should be four VCO clock cycles. It should also stay low for four VCO Clock Cycles.
For each toggling of PHASESTEP, you are getting [1/(8*2)]*360 = 22.5 degree phase shift (delayed).

16.4.2. Divider Phase Shift
Once the PHASESEL and PHASEDIR have been set a post-divider phase adjustment is made by toggling the PHASELOADREG signal. Each pulse of the PHASELOADREG signal generates a phase shift. The step size relative to the unshifted output is specified by this equation:

\[
\frac{[\text{CLKO}<n>\_\text{CPHASE} - \text{CLKO}<n>\_\text{DIV}]}{\text{CLKO}<n>\_\text{DIV} + 1} \times 360
\]

Where <n> is the clock output specified by PHASESEL (CLKOP/OS/OS2/OS3). Values for CLKO<n>_CPHASE and CLKO<n>_DIV are located in the HDL source file. Please note that if these values are both 1, no shift is made.

![Figure 16.6. Divider Phase Shift Timing Diagram](image)

*Note:  Minimum Time Before Shifting Again Equation = 2.5*(CLKO<n>_DIV + 1) + (CLKO<n>_CPHASE +1) ] * (Period of Divider Clock).

16.5. Low Power Features
The CrossLink-NX PLL contains several features that allows you to reduce the power usage of a design including Standby mode support and Dynamic clock enable.

16.5.1. Dynamic Clock Enable
The Dynamic Clock Enable feature allows you to glitchlessly enable and disable selected output clocks during periods when not used in the design. A disabled output clock is logic 0. Re-enabled clocks start on the falling edge of CLKOP. 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, ENCLKOS3, ENCLKOS4, and ENCLKOS5. Each clock enable port has an option in the IP Catalog user interface to bring the signal to the top level ports of the PLL. If external feedback is used on a port or if the clock output is not enabled, its dynamic clock enable port is unavailable.
16.5.2. Standby Mode

The PLL can also be put into standby mode. This is similar to reset in that the PLL is still powered, however, the VCO is not running and the clock outputs driven low. The PLL enters Standby mode when the STDBY signal is driven high and the outputs are driven low. You need to stay in the STDBY mode for at least 1 ms to make sure the PLL analog circuits are fully reset and to have a stable analog startup. The PLL can be restarted when it is needed again and the output clocks are reactivated. It takes $T_{lock\_time} = 10$ ms to achieve PLL lock again. To support this mode, the Standby Port option is in the IP Catalog Wizard user interface and causes the STDBY port to be brought out to the top level of the PLL module.

16.6. PLL Usage in IP Catalog

IP Catalog is used to create and configure a PLL. PLL can be found in the IP Catalog under Module - Architecture Modules. The graphical user interface is used to select parameters for the PLL. The result is an HDL block to be used in the simulation and synthesis flow.

The main window when the PLL is selected is shown in Figure 16.8. When opening IP Catalog inside a Lattice Radiant project, the only entry required is the file name as the other entries are set to the project settings. After entering the module name of choice, click Next to open the PLL configuration window as shown in Figure 16.8.
16.6.1. Configuration Tab

The configuration window lists all user accessible attributes with default values set. Upon completion, click Generate to generate the source.

16.6.2. PLL Frequency and Phase Configuration

In the General Tab, enter the input and output clock frequencies and the software calculates the divider settings. If an entered value is out of range, it is displayed in red and an error message is displayed. You can also select a tolerance value from the Tolerance % drop-down box.

If you are new to the CrossLink-NX PLL user interface, enter the desired phase shift and the software calculates the closest achievable shift. After the desired phase is entered, clicking the Calculate button displays the closest achievable phase shift in the Actual Phase text box. If an entered value is out of range, it is displayed in red and an error message is displayed.

General Tab

![Figure 16.9. CrossLink-NX PLL Frequency Configuration in General Tab](image-url)
<table>
<thead>
<tr>
<th>User Parameters</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
<th>Corresponding HDL Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKI</td>
<td>Frequency</td>
<td>10 — 800 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKI</td>
</tr>
<tr>
<td></td>
<td>Divider Desired Value (read only)</td>
<td>1 - 128</td>
<td>1</td>
<td>CLKI_DIV</td>
</tr>
<tr>
<td></td>
<td>Divider Actual Value (read only)</td>
<td>—</td>
<td>1</td>
<td>—</td>
</tr>
<tr>
<td>PLL Reference Clock</td>
<td>PLL Reference Clock from Pin</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>I/O Standard for Reference Clock</td>
<td>LVDS, SBLVDS, SLVS, HSTL15_I, HSTL15D_I, LVTTL33, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS18H, LVCMOS15, LVCMOS15H, LVCMOS12, LVCMOS12H, LVCMOS10, LVCMOS10H, LVCMOS10r</td>
<td>LVDS</td>
<td>—</td>
</tr>
<tr>
<td>CLKFB</td>
<td>Feedback Mode</td>
<td>INTCLOCK, CLKOP, INTCLKOS, CLKOS, INTCLKOS2, CLKOS2, INTCLKOS3, CLKOS3, INTCLKOS4, CLKOS4, INTCLKOS5, CLKOS5</td>
<td>CLKOP</td>
<td>FEEDBK_PATH</td>
</tr>
<tr>
<td></td>
<td>FBK Divider Desired Value (read only)</td>
<td>1 — 128</td>
<td>1</td>
<td>CLKFB_DIV</td>
</tr>
<tr>
<td></td>
<td>FBK Divider Actual Value (read only)</td>
<td>1 — 128</td>
<td>1</td>
<td>—</td>
</tr>
<tr>
<td>CLKFB Modifier</td>
<td>Fractional-N Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>CLKFB Modifier</td>
<td>Spread Spectrum Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>CLKFB Modifier</td>
<td>FBK Divider Desired Value (read only)</td>
<td>0 — 4095</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>CLKFB Modifier</td>
<td>FBK Divider Actual Value (read only)</td>
<td>—</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>VCO</td>
<td>VCO Frequency (read only)</td>
<td>—</td>
<td>800</td>
<td>CLKFB_DIV</td>
</tr>
<tr>
<td>CLKOP</td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>OUTDIVIDER_MUXA</td>
</tr>
<tr>
<td></td>
<td>Desired Frequency*</td>
<td>3.125 — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOP</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>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Divider Desired Value (read only)</td>
<td>1 - 128</td>
<td>—</td>
<td>CLKOP_DIV</td>
</tr>
<tr>
<td></td>
<td>Divider Actual Value (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Frequency Actual Value (MHz) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>ERROR (PPM) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Enable Trim for CLKOP</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Duty Trim Options Mode</td>
<td>Rising/Falling</td>
<td>Falling</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Duty Trim Options Delay Multiplier</td>
<td>0, 1, 2, 4</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>CLKOS</td>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>CLKOS_Enable</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>OUTDIVIDER_MUXB</td>
</tr>
<tr>
<td></td>
<td>Desired Frequency*</td>
<td>3.125 — 400 MHz</td>
<td>100 MHz</td>
<td>—</td>
</tr>
<tr>
<td>User Parameters</td>
<td>Description</td>
<td>Range</td>
<td>Default</td>
<td>Corresponding HDL Attribute</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------</td>
<td>----------------------------</td>
<td>---------</td>
<td>-----------------------------</td>
</tr>
<tr>
<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>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Divider Desired Value (read only)</td>
<td>1 - 128</td>
<td>—</td>
<td>—</td>
<td>CLKOS_DIV</td>
</tr>
<tr>
<td>Divider Actual Value (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Frequency Actual Value (MHz) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ERRPR (PPM) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Desired Frequency*</td>
<td>3.125 — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOS2</td>
<td></td>
</tr>
<tr>
<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>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Divider Desired Value (read only)</td>
<td>1 - 128</td>
<td>—</td>
<td>—</td>
<td>CLKOS2_DIV</td>
</tr>
<tr>
<td>Divider Actual Value (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Frequency Actual Value (MHz) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ERRPR (PPM) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Desired Frequency*</td>
<td>3.125 — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOS3</td>
<td></td>
</tr>
<tr>
<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>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Clock Divider (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CLKOS3_DIV</td>
</tr>
<tr>
<td>Divider Actual Value (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Frequency Actual Value (MHz) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ERROR (PPM) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Desired Frequency*</td>
<td>3.125 — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOS4</td>
<td></td>
</tr>
<tr>
<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>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Clock Divider (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CLKOS4_DIV</td>
</tr>
<tr>
<td>Divider Actual Value (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Frequency Actual Value (MHz) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ERROR (PPM) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Enable</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Bypass</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Desired Frequency*</td>
<td>3.125 — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOS5</td>
<td></td>
</tr>
<tr>
<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>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Clock Divider (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>CLKOS5_DIV</td>
</tr>
<tr>
<td>Divider Actual Value (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Frequency Actual Value (MHz) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ERROR (PPM) (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

**Note:** If this clock is selected as feedback, the minimum output frequency that is achievable is 10 MHz.
Phase Tab

![Diagram of PLL configuration]

**Figure 16.10. CrossLink-NX PLL Phase Configuration Tab**

<table>
<thead>
<tr>
<th>User Parameters</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
<th>Corresponding HDL Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKOP</td>
<td>Static Phase Shift Desired Degrees</td>
<td>0 - 360</td>
<td>0</td>
<td>CLKOP_CPHASE, CLKOP_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Static Phase Shift Actual Degrees (read only)</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>CLKOS</td>
<td>Static Phase Shift Desired Degrees</td>
<td>0 - 360</td>
<td>0</td>
<td>CLKOS_CPHASE, CLKOS_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Static Phase Shift Actual Degrees (read only)</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>CLKOS2</td>
<td>Static Phase Shift Desired Degrees</td>
<td>0 - 360</td>
<td>0</td>
<td>CLKOS2_CPHASE, CLKOS2_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Static Phase Shift Actual Degrees (read only)</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>CLKOS3</td>
<td>Static Phase Shift Desired Degrees</td>
<td>0 - 360</td>
<td>0</td>
<td>CLKOS3_CPHASE, CLKOS3_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Static Phase Shift Actual Degrees (read only)</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>CLKOS4</td>
<td>Static Phase Shift Desired Degrees</td>
<td>0 - 360</td>
<td>0</td>
<td>CLKOS4_CPHASE, CLKOS4_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Static Phase Shift Actual Degrees (read only)</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>CLKOS5</td>
<td>Static Phase Shift Desired Degrees</td>
<td>0 - 360</td>
<td>0</td>
<td>CLKOS5_CPHASE, CLKOS5_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Static Phase Shift Actual Degrees (read only)</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>
Optional Ports Tab

![Optional Ports Tab Diagram](image)

Figure 16.11. CrossLink-NX PLL Optional Ports Configuration Tab
### Table 16.8. Tab 3, PLL Optional Ports, IP Catalog User Interface

<table>
<thead>
<tr>
<th>User Parameters</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
<th>Corresponding HDL Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Phase ports</td>
<td>Provides Dynamic Phase Shift ports.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>DPHASE_SOURCE</td>
</tr>
<tr>
<td>Clock Enable OP</td>
<td>Provides ENCLKOP; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>Clock Enable OS</td>
<td>Provides ENCLKOS; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>Clock Enable OS2</td>
<td>Provides ENCLKOS2; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>Clock Enable OS3</td>
<td>Provides ENCLKOS3; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>Clock Enable OS4</td>
<td>Provides ENCLKOS4; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>Clock Enable OS5</td>
<td>Provides ENCLKOS5; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>PLL Reset Option</td>
<td>Provides PLL Reset signal.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>PLLRST_ENA</td>
</tr>
<tr>
<td>Lock Settings</td>
<td>Provides the LOCK signal.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>LMMI Interface</td>
<td>Provides LMMI Port.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>Power Mode Settings</td>
<td>Enable Legacy Mode.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enable Power down Mode.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>

### Analog Settings Tab

![Module/IP Block Wizard](image)

*Figure 16.12. CrossLink-NX PLL Analog Settings Tab*
Table 16.9. Tab 4, PLL Analog Settings, IP Catalog User Interface

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
<th>Corresponding HDL Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>Operating Voltage</td>
<td>1 V</td>
<td>1 V</td>
<td>1 V</td>
</tr>
<tr>
<td>BW Factor</td>
<td>BW Factor</td>
<td>10 - 30</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

For the PLL, IP Catalog sets attributes in the HDL module 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 design flow, FREQUENCY preferences is included in the preference file to automatically constrain the clocks produced by the PLL. For a step-by-step guide to using IP Catalog, refer to the IP Catalog User Manual.
Appendix A.  Primary Clock Sources and Distribution

Figure A.1 and Figure A.2 show the inputs into the Primary Clock Network through the mid-mux into the centermux for each device. There are DCC components at the input of the centermux to allow you to stop the clock to save power.

Figure A.1. CrossLink-NX Primary Clock Sources and Distribution, LIFCL-40 Devices
Figure A.2. CrossLink-NX Primary Clock Sources and Distribution, LIFCL-17 Devices
Appendix B. Pinout Rules for Clocking in CrossLink-NX Devices

In the CrossLink-NX device, as with all other architectures, there are general rules and guidelines for board designers to follow. These rules give the best possible timing and allow for a successful design.

In the .csv file where pins are listed, under the `Dual Function` section, you can see the PCLK and PLL input pins listed as below:

Primary Clock Input Pin — PCLKT<Bank>_<0/1>
Dedicated PLL Input Pin — <LOC>_GPLL0T_IN

**Table B.1. Clock Input Selection Table**

<table>
<thead>
<tr>
<th>Clock Input</th>
<th>Pin to Use</th>
<th>Clock Routing Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock Input to Logic Directly</td>
<td>PCLK Input Pin</td>
<td>Uses Primary Clock Routing for the Clock.</td>
</tr>
<tr>
<td>Clock Input to PLL Only</td>
<td>PLL Input Pin</td>
<td>Uses a Dedicated PLL Input. No Primary Clock Routing is used.</td>
</tr>
<tr>
<td>Clock Input to Logic and PLL</td>
<td>PCLK Input Pin</td>
<td>Uses Primary Clock Routing for the Clock.</td>
</tr>
<tr>
<td>Clock input to more than 2 PLLs</td>
<td>PCLK Input Pin</td>
<td>Uses Primary Clock Routing for the Clock.</td>
</tr>
</tbody>
</table>
Technical Support Assistance

For technical support or for additional information regarding security, lock policy settings, and authentication commands, submit a technical support case through www.latticesemi.com/techsupport.
## Revision History

**Revision 1.0, November 2019**

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