sysCLOCK PLL Design User Guide for Nexus Platform

Technical Note

FPGA-TN-02095-2.7

October 2025

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
APB	Advanced Peripheral Bus
CDR	Clock and Data Recovery
CIB	Common Interface Block
DCC	Dynamic Clock Control
DCS	Dynamic Clock Select
DDR	Double Data Rate
DLL	Delay Locked Loop
DTR	Digital Temperature Readout
GSR	Global Set Reset
IP	Intellectual Property
LMMI	Lattice Memory Mapped Interface
LVDS	Low-Voltage Differential Signaling
MIB	Memory Interface Block
PCS	Physical Coding Sublayer
PLC	Programmable Logic Cell
PLL	Phase-Locked Loop
RGMII	Reduced Gigabit Media Independent Interface
SED	Soft Error Detect
SERDES	Serializer/Deserializer
SGMII	Serial Gigabit Media Independent Interface
VCO	Voltage Controlled Oscillator
VHDL	VHSIC Hardware Description Language

1. Introduction

This user guide describes the clock resources available in the Lattice Nexus™ Platform architecture, which includes CrossLink™-NX, Certus™-NX, CertusPro™-NX, and MachXO5™-NX product families.

The 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 Nexus 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

Parameter	Description	LIFCL-17 LFD2NX-17 LFD2NX-9	LIFCL-33 LIFCL-33U	LIFCL-40 LFD2NX-40 LFD2NX-28	LFD2NX-15 LFD2NX-25 LFD2NX-35 LFD2NX-65	LFCPNX-50	LFCPNX-100 LFMXO5-55TDQ	LFMXO5-15 LFMXO5-20TD LFMXO5-20TDQ LFMXO5-25 LFMXO5-30TD LFMXO5-30TDQ LFMXO5-35/T LFMXO5-65/T
Number of PLLs	General purpose Phase Locked Loops.	2	1	3	2	3	4	2
Number of Edge Clocks	Edge Clocks for high-speed interfaces.	12	12	12	8	12	12	8
Number of Edge Clock Dividers	Edge Clock Dividers for high-speed interfaces.	12	12	12	8	12	12	8
Number of Primary Clock Dividers	Programmable Primary Clock dividers for domain crossing applications.	1	1	1	1	2	2	1
Number of DDRDLLs	DDRDLL used for DDR memory and High Speed I/O interfaces	2	2	2	2	2	2	2

It is very important to validate the device pinout using the Lattice Radiant™ tool to avoid implementation issues.

2. Clock/Control Distribution Network

Nexus devices provide global clock distribution in the form of global primary clocks. The device is organized into clock regions; each clock region can accommodate 16 primary clocks. For CrossLink-NX, Certus-NX, and MachXO5-NX, there are two clock regions and for CertusPro-NX there are four clock regions. There is a maximum of 64 unique clock input sources. The Nexus primary clocking structure is Edge Clock rich and contains generous low-skew Primary clock resources.

3. Nexus Top-Level View

A top-level view of the major clocking resources for CrossLink-NX and Certus-NX (LFD2NX-9, LFD2NX-17, LFD2NX-28, and LFD2NX-40) devices are 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, LFD2NX-17, and LFD2NX-9 devices. The MIPI_DPHY0 and MIPI_DPHY1 on the top are only available for the CrossLink-NX family.

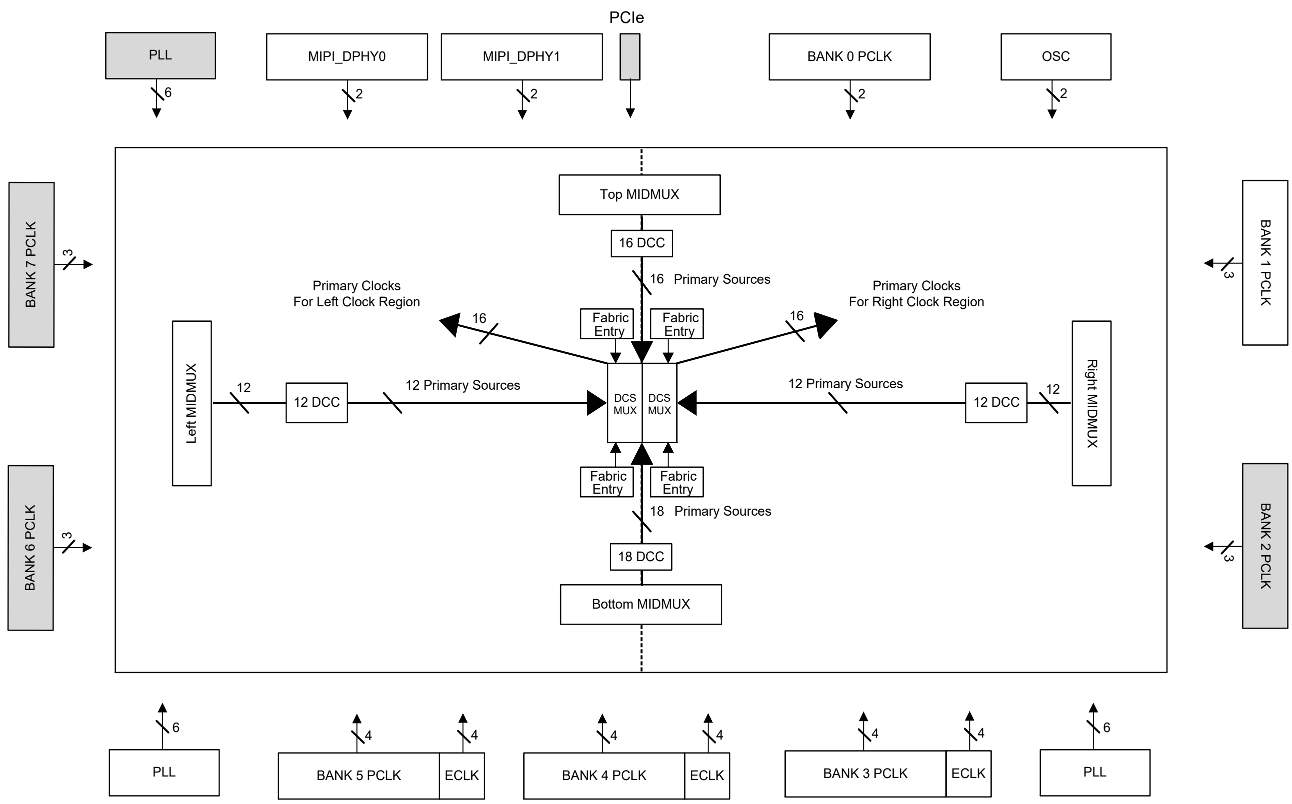

Figure 3.1. CrossLink-NX and Certus-NX (LFD2NX-9, LFD2NX-17, LFD2NX-28, and LFD2NX-40) Clocking Structure

A top-level view of the major clocking resources for the CertusPro-NX (LFD2NX-100) and MachXO5-NX (LFMXO5-55TDQ) devices are shown in Figure 3.2. The Upper Right PLL is only for LFCPNX-100.

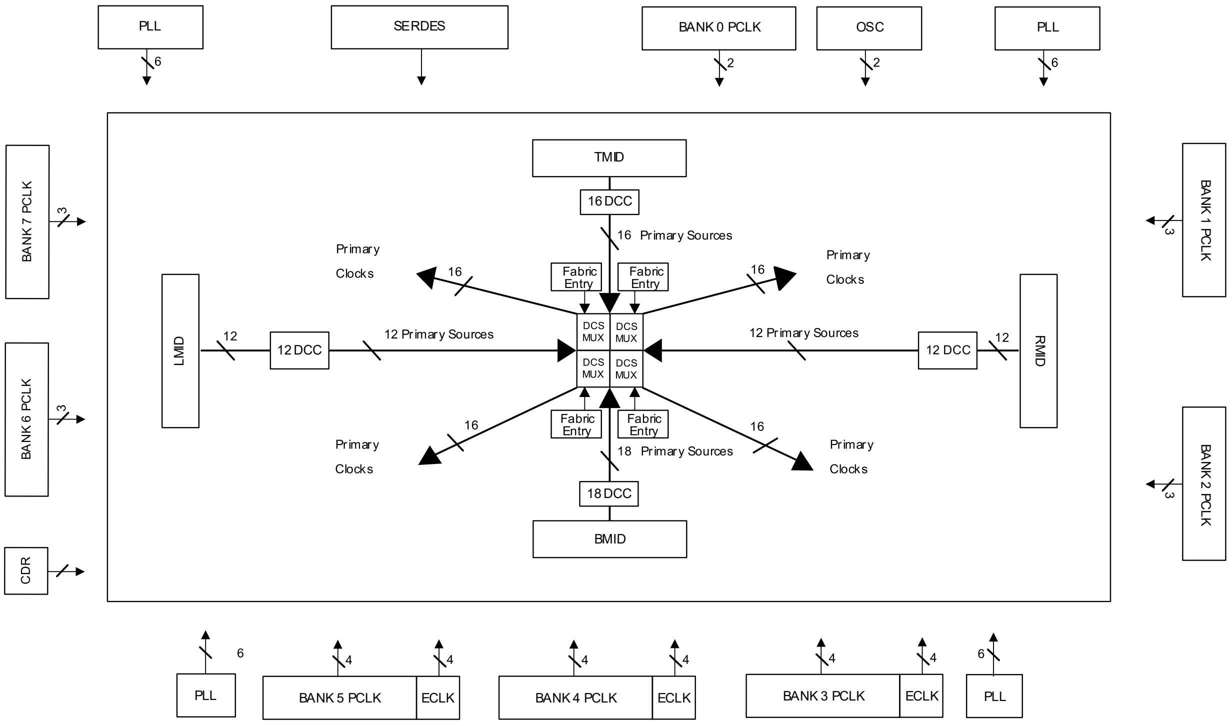

Figure 3.2. CertusPro-NX (LFCPNX-100) and MachXO5-NX (LFMXO5-55TDQ) Clocking Structure

A top-level view of the major clocking resources for the MachXO5-NX (LFMXO5-15, LFMXO5-20TD, LFMXO5-20TDQ, LFMXO5-25, LFMXO5-30TD, LFMXO5-30TDQ, LFMXO5-35/T, and LFMXO5-65/T) and Certus-NX (LFD2NX-15, LFD2NX-25, LFD2NX-35, and LFD2NX-65) devices is shown in Figure 3.3. Bank 11 on the bottom is not available for LFMXO5-25 and LFD2NX-15 devices.

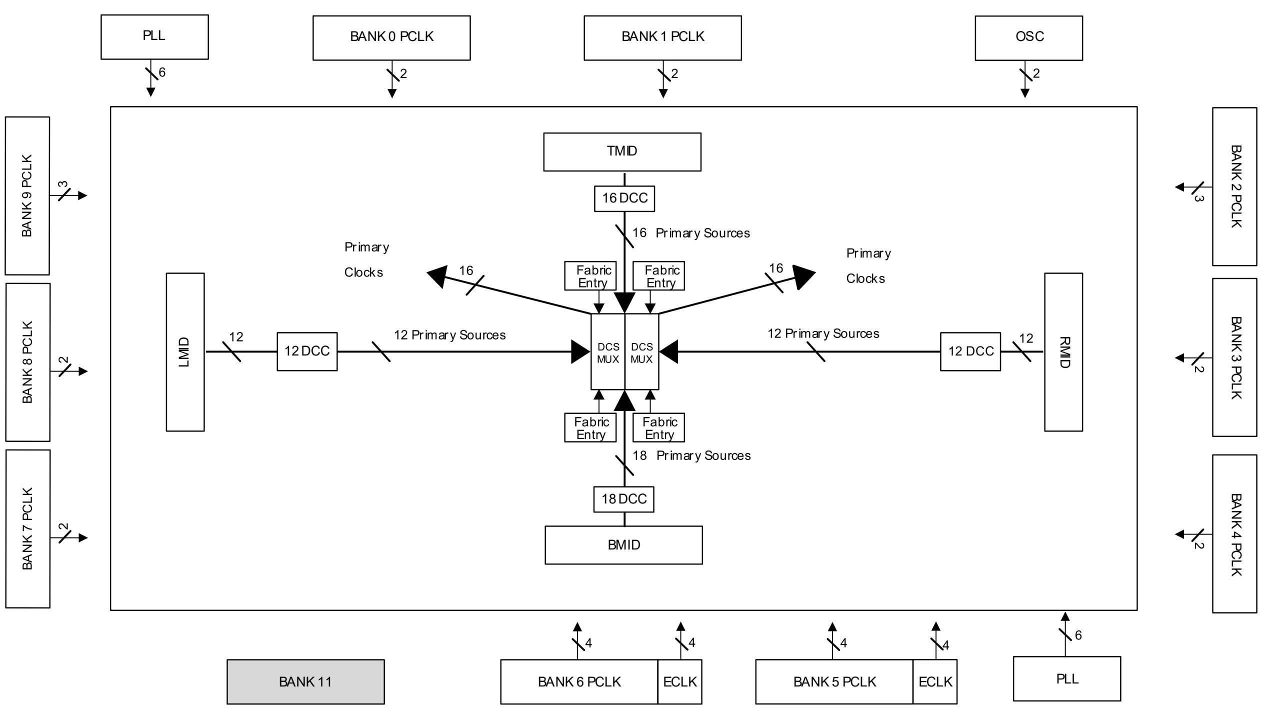

Figure 3.3. MachXO5-NX (LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T) and Certus-NX (LFD2NX-15/25/35/65) Clocking Structure

A top-level view of the major clocking resources for the CrossLink-NX-33 and CrossLink-NX-33U devices are shown in Figure 3.4.

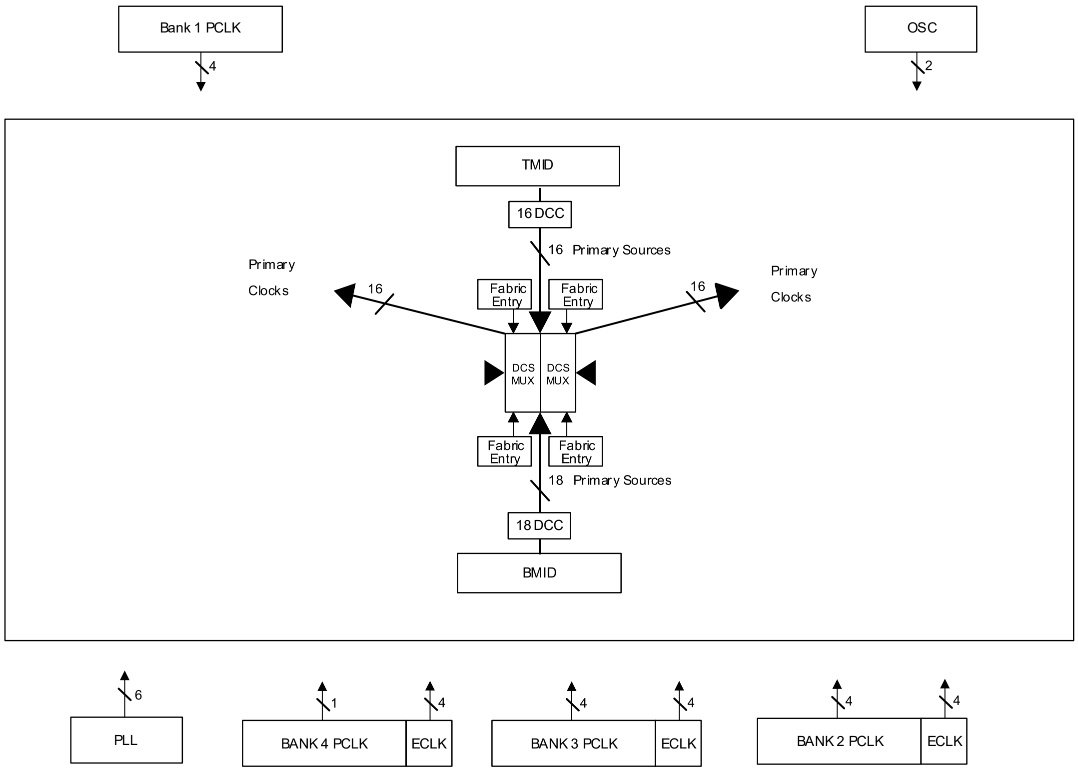

Figure 3.4. CrossLink-NX-33 and CrossLink-NX-33U Clocking Structure

4. Clocking Architecture Overview

This section provides 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 Nexus Devices section.

4.1. Primary Clock Network

Up to 32 primary clocks (for CrossLink-NX, CrossLink-NX-33, CrossLink-NX-33U, Certus-NX, and MachXO5-NX) or 64 primary clocks (for CertusPro-NX) 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 regions (for CrossLink-NX, CrossLink-NX-33, CrossLink-NX-33U, Certus-NX, and MachXO5-NX) or four clocking regions (for CertusPro-NX), each region associated with a DCS_CMUX. Each of these regions has 16 clocks that can be distributed to the fabric in the region. Initially, the Lattice Radiant software automatically routes each clock region; up to a maximum of 16 clocks. The user can change how the clocks are routed by specifying a preference in the Lattice Radiant project constraints file to locate the clock to specific region.

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 Nexus devices. There are four Edge Clocks per bank located on the bottom side of the device.

5. Overview of Clocking Components

5.1. Edge Clock Dividers (ECLKDIV)

Edge Clock dividers are provided to create the divided down clocks used for the I/O Mux/DeMux gearing logic (SCLK inputs of DDR I/O) and they drive the Primary Clock network. There are twelve Edge Clock Dividers on the Nexus device.

5.2. Primary Clock Divider (PCLKDIVSP)

For CrossLink-NX, Certus-NX, and MachXO5-NX, one programmable Primary Clock Divider is provided to create the divided down clocks. For CertusPro-NX, two programmable Primary Clock Dividers are available.

5.3. Dynamic Clock Select (DCS)

The 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, Certus-NX, and MachXO5-NX devices, and there are two dynamic clock select blocks on the CertusPro-NX device.

5.4. Dynamic Clock Control (DCC)

Dynamic Clock Control allows dynamic clock to enable and disable the MIDMUX selection for the four special common interface block (CIB) clocks from the core. When the input is disabled, all the logic and clock signals that are fed by the output do not toggle. Hence, it reduces the overall dynamic power consumption of the device.

5.5. Edge Clock Sync (ECLKSYNC)

The Nexus devices have dynamic edge clock synchronization control (ECLKSYNC) which allows each edge clock to be disabled or enabled glitchlessly from core logic if desired. This allows the user 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.

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 OSCA on the Nexus device. The oscillator clock output is routed directly to primary clocking.

The oscillator output is not a high-accuracy clock. 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. Primary Clocks

6.1. 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 connect to the primary clock routing are:

- Dedicated Clock Input Pins
- PLL Outputs
- PCLKDIVSP/ECLKDIV Outputs
- Internal FPGA Fabric Entries (with minimum general routing)
- SGMII[1]-CDR, SERDES/PCS clocks
- OSC Clock

Note: The SGMII interface using LVDS I/O has limitations when operating across the full specified temperature range. Lattice recommends using alternative interfaces, such as SERDES or RGMII, for designs requiring Gigabit Ethernet. Refer to the Knowledge Base article for details. Contact your local Lattice sales representative for more information.

All potential primary clock sources are multiplexed prior to going to the primary clock routing by a MIDMUX. There are 58 MIDMUX connections and four FPGA fabric connections, 62 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:

18 bottom MIDMUX sources + 16 top MIDMUX sources + 12 left MIDMUX sources + 12 right MIDMUX sources + 4 direct FPGA fabric entry points (from general routing) = 62.

The basic clocking structure is shown in Figure 3.1 and Figure 3.2, elaborated in Appendix A. Primary Clock Sources and Distribution.

6.2. 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 MIDMUX, then selected in the centermux and distributed on the primary clock routing to clock the synchronous elements in the FPGA fabric. For CrossLink-NX, Certus-NX, and MachXO5-NX, the primary clock routing network is divided into left and right regions. Figure 6.1 is the simplified view of Figure 3.1. For CertusPro-NX, the primary clock routing network is divided into four regions, up-left, up-right, low-left, and low-right. Figure 6.2 is the simplified view of Figure 3.2.

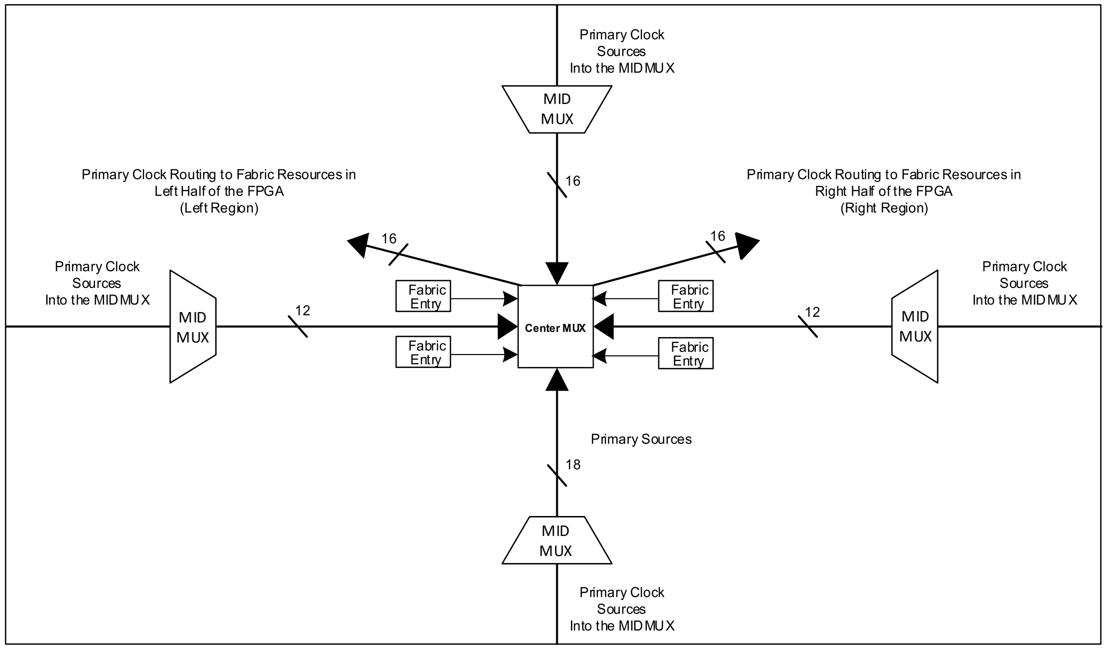

Figure 6.1. Primary Clock Routing Architecture for CrossLink-NX, Certus-NX, and MachXO5-NX (LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T)

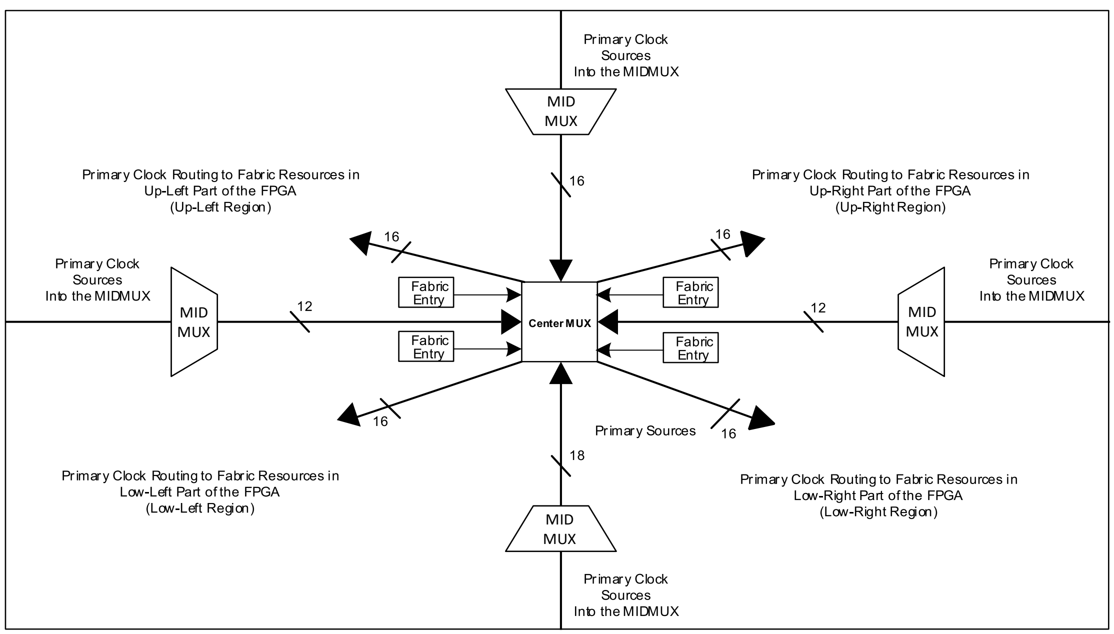

Figure 6.2. Primary Clock Routing Architecture for CertusPro-NX and MachXO5-NX (LFMXO5-55TDQ)

The centermux can source up to 16 independent primary clocks per region, which can clock the logic located in that region. The centermux can also route each clock source to all regions. The Lattice Radiant software automatically routes a primary clock to the regions in the FPGA.

6.3. Dedicated Clock Inputs

The Nexus 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 and to Edge Clock routing resources. A dedicated PCLK clock pin must always be used to route an external clock source to FPGA and I/O logic.

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 as described in Dedicated PLL Inputs section. SERDES reference clocks also have dedicated SERDES reference clock pins. The Nexus 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. See Appendix A for more details.

7. Primary Clock Divider (PCLKDIVSP)

Inside the centermux, one (for CrossLink-NX, Certus-NX, and MachXO5-NX) or two (for CertusPro-NX) Primary Clock Dividers are available. Each Primary Clock Divider provides the following functionalities:

- PCLK Divider supports ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, and ÷128. When PCLK divider is bypassed, it is ÷1 mode.
- PCLK Divider can be reset by global Reset signals and sleep mode control signals. The global reset can be disabled by a configuration bit.
- PCLK Divider supports user Local Reset 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 GSR activity to be ignored during device power up by gating this signal with internal DONE.
- When exiting from sleep mode, the retention registers are released from the asynchronous reset control.

7.1. PCLKDIVSP Component Definition

The PCLKDIVSP component can be instantiated in the source code of a design as defined in this section. Figure 7.1, Table 7.1, and Table 7.2 define the PCLKDIVSP component. Verilog and VHDL instantiations are included.

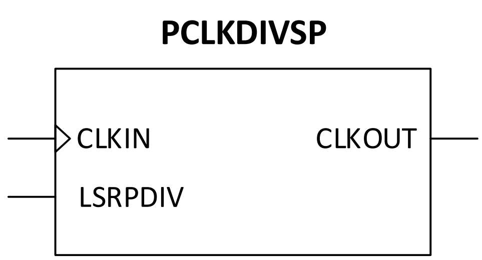

Figure 7.1. PCLKDIVSP Component Symbol

Table 7.1. PCLKDIVSP Component Port Definition

Port Name	I/O	Description
CLKIN	I	Primary Clock Input
LSRPDIV	I	Local Reset — Active High, asynchronously forces all outputs low. LSRPDIV = 0 Clock outputs are active LSRPDIV = 1 Clock outputs are OFF
CLKOUT	O	Output Port

Table 7.2. PCLKDIVSP Component Attribute Definition

Name	Value	Default	Description
DIV_PCLKDIV	X1, X2, X4, X8, X16, X32, X64, X128	X1	Primary Clock Divide Ratio Selection
GSR	ENABLE DISABLE	ENABLED	GSR ENABLE/DISABLE Selection

7.2. PCLKDIVSP Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
component PCLKDIVSP
generic (DIV_PCLKDIV      : string;
         GSR     : string);
port    (CLKIN   : in STD_LOGIC;
         LSRPDIV : in STD_LOGIC;
         CLKOUT  : out STD_LOGIC);
end component;
```

PCLKDIVSP Instantiation

```
attribute DIV_PCLKDIV : string;
attribute DIV_PCLKDIV of I1 : label is "X1";
attribute GSR : string;
attribute GSR of I1 : label is "DISABLED";

I1: PCLKDIVSP
generic map (DIV_PCLKDIV => "X2",
             GSR         => "DISABLED")
port map    (CLKIN       => CLKIN,
             LSRPDIV     => LSRPDIV,
             CLKOUT      => CLKOUT);
```

7.3. PCLKDIVSP Usage in Verilog

Component and Attribute Declaration

```
module PCLKDIVSP (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
```

PCLKDIVSP Instantiation

```
defparam I1.DIV_PCLKDIV = "X2";
defparam I1.GSR = "DISABLED";
PCLKDIVSP I1 (
   .CLKIN     (CLKIN),
   .LSRPDIV   (LSRPDIV),
   .CLKOUT    (CLKOUT));
```

8. Dynamic Clock Select (DCS)

One (for CrossLink-NX, Certus-NX, and MachXO5-NX) or two (for CertusPro-NX) dynamic clock select (DCS) blocks are located at the center of the PLC array, which can drive to any or all the regions. The DCS_CMUX Structures are shown in Figure 8.1 and Figure 8.2.

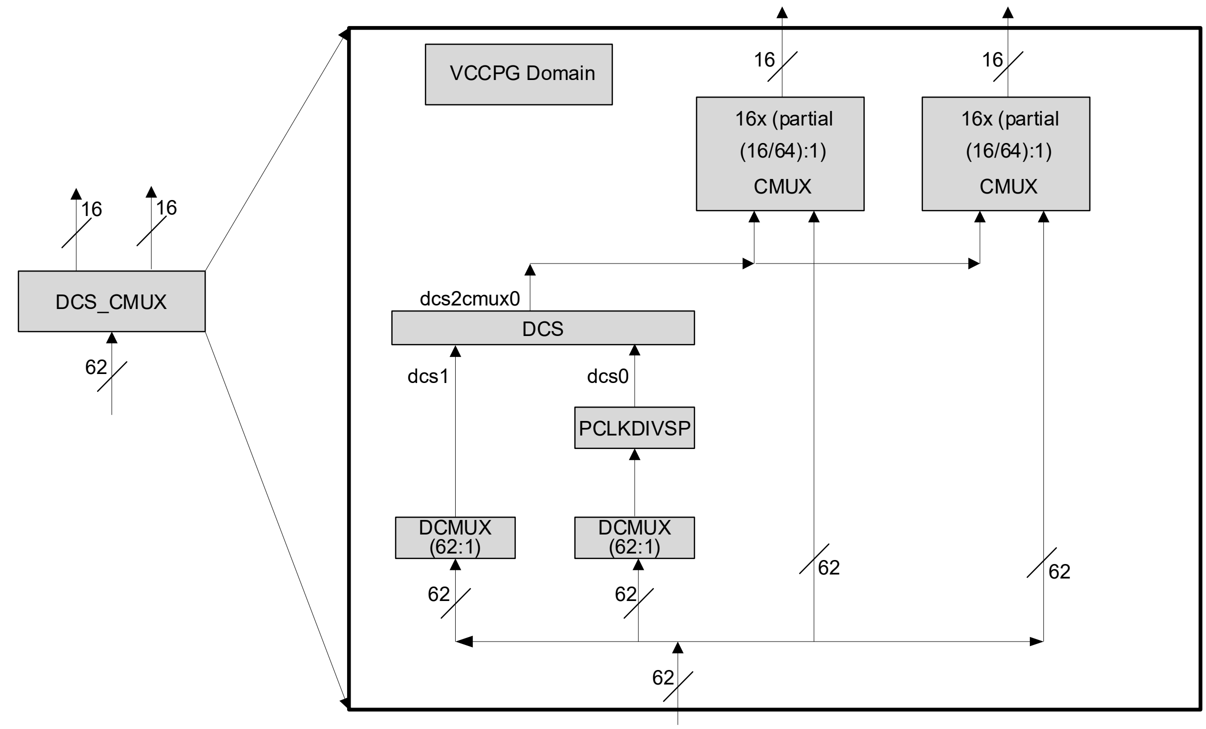

Figure 8.1. DCS_CMUX Structure for CrossLink-NX, Certus-NX, and MachXO5-NX

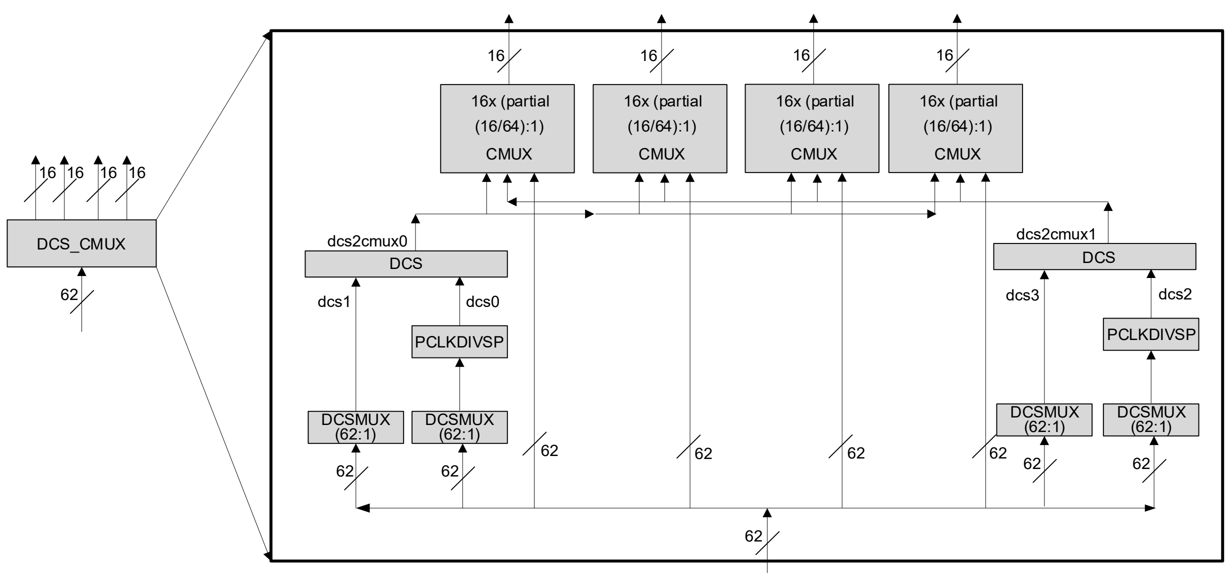

Figure 8.2. DCS_CMUX Structure for CertusPro-NX

The DCS block allows dynamic and glitchless selection between two PCLK clock sources. The DCS block shares the same clock resource as any PCLK CMUX. This way the DCS function can be performed on any two primary clock sources. 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. The output of the DCS is connected to the inputs of Primary Clock Center MUXs. The DCS logic structure is shown in Figure 8.3.

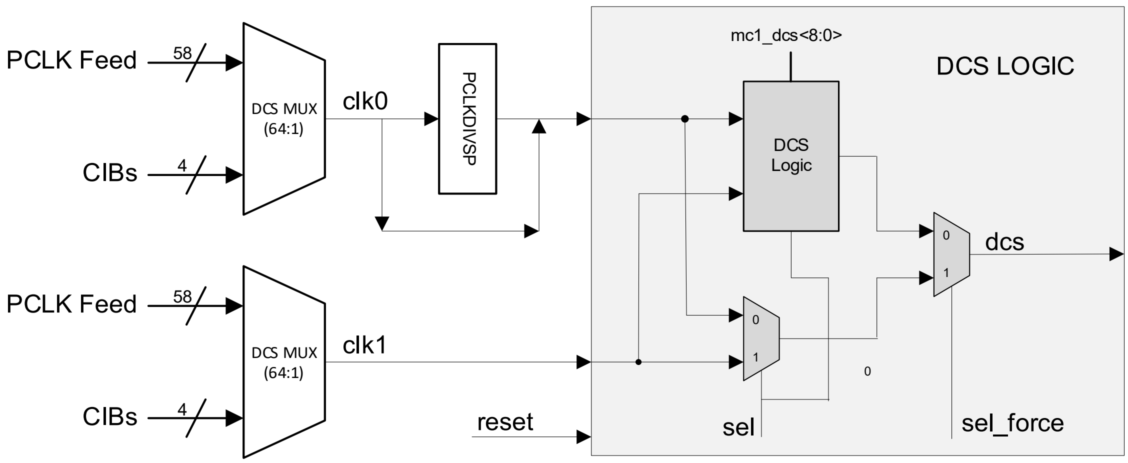

Figure 8.3. DCS Logic Structure

The *DCSMODE* attribute sets the behavior of the DCS output. The DCS attributes are described in Table 8.2.

8.1. DCS Timing Diagrams

The DCS block allows dynamic and glitchless selection between two PCLK clock sources. The DCS block shares the same clock resource as any PCLK CMUX. Therefore, the DCS function can be performed on any two primary clock sources. Figure 8.4, Figure 8.5, and Figure 8.6 show the DCS in glitchless operation in conjunction with the DCSMODE attribute. Figure 8.7 shows the non-glitchless bypass operation scenario.

8.1.1. 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.

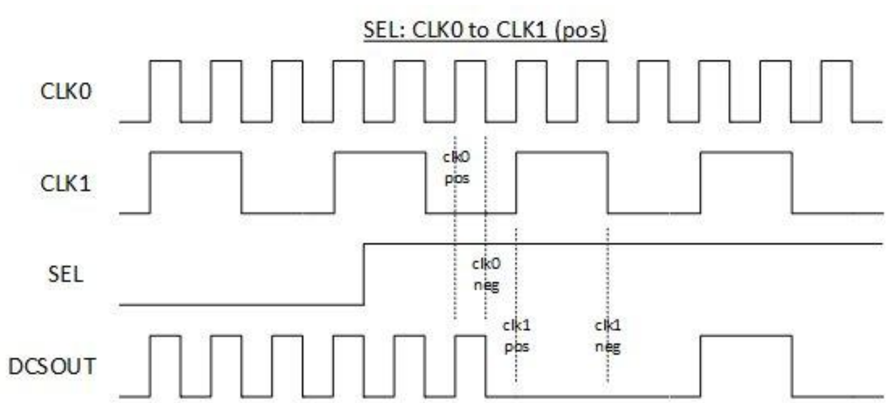

Figure 8.4. Posedge DCS Switch from SEL: 0 => 1

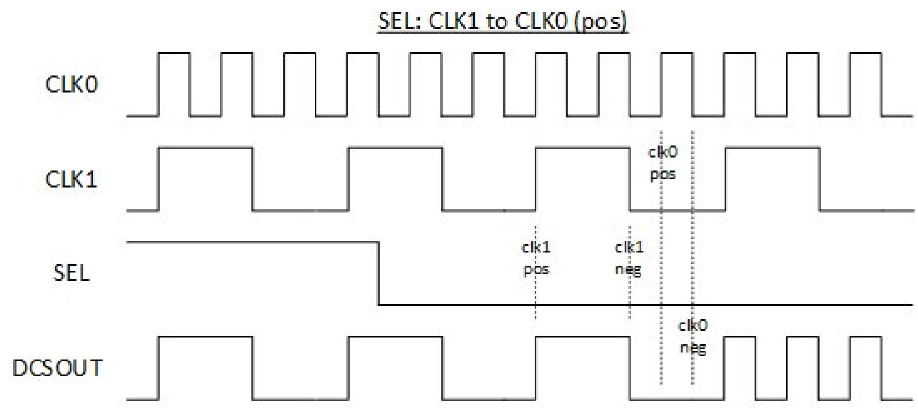

Figure 8.5. Posedge DCS Switch from SEL: 1 => 0

8.1.2. 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.

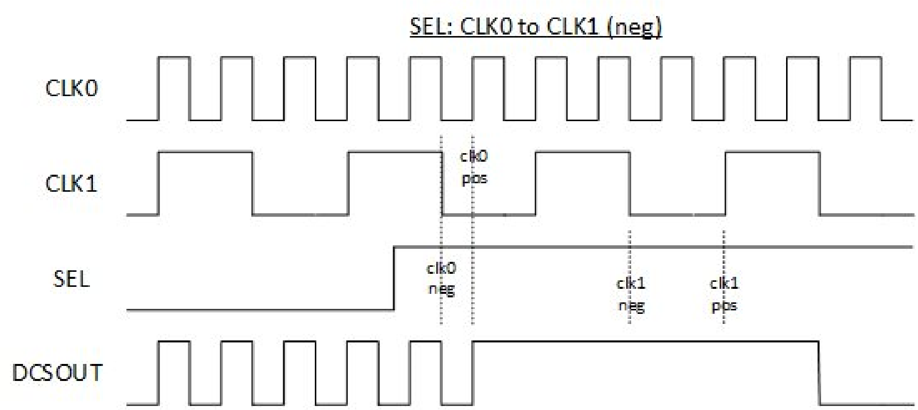

Figure 8.6. Negedge DCS Switch from SEL: 0 => 1

8.1.3. Functionality – bypass

When SELFORCE is high, the switch is in bypass mode. The output clock transitions immediately from the current clock to the target clock and may have glitches.

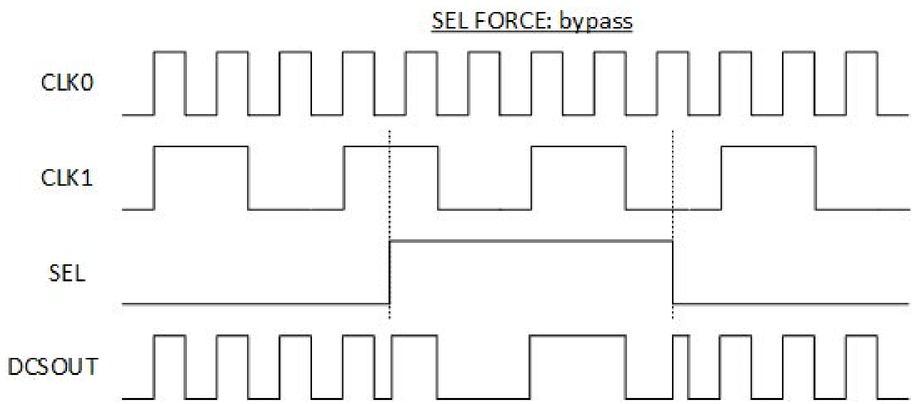

Figure 8.7. SELFORCE = 1 DCS Clock Switch

8.2. DCS Component Definition

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

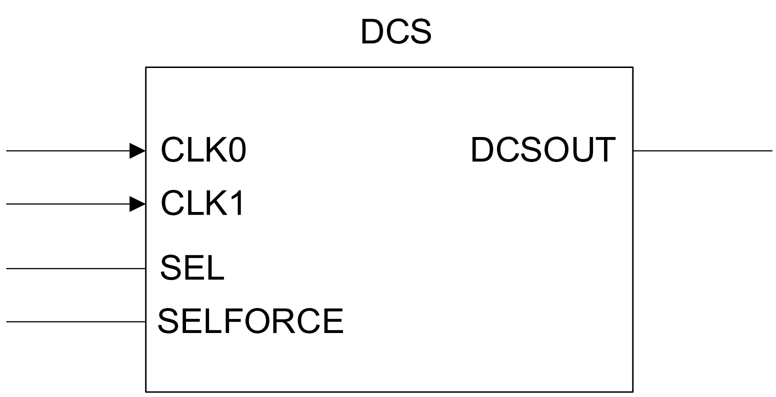

Figure 8.8. DCS Component Symbol

Table 8.1. DCS Component Port Definition

Port Name	I/O	Description
CLK0	I	Clock Input port 0 — Default
CLK1	I	Clock Input port 1
SEL	I	Input Clock Select
SELFORCE	I	Selects Glitchless (*0*) or Non-Glitchless (*1*) behavior Glitchless: No risk of glitch during switching. Non-Glitchless: There is risk of glitch during switching.
DCSOUT	O	Clock Output Port

Table 8.2 provides the behavior of the DCS output based on the setting of the *DCSMODE* attribute and the SELFORCE pin input. The SELFORCE pin is dynamic and can toggle during operation. The glitchless switching is only achievable when SELFORCE = *0*.

Table 8.2. DCS – DCSMODE Attribute

Attribute Name	Attribute Value	Output		Description
		SEL = 0	SEL = 1	
DCSMODE (SELFORCE = 0)	VCC	1	1	VCC
	GND	0	0	VSS
	DCS	CLK0	CLK1	Falling edge triggered, latched state is low
	DCS_1	CLK0	CLK1	Rising edge triggered, latched state is high
	BUFGCECLK1	0	CLK1	SEL is active high, disabled output is low
	BUFGCECLK1_1	1	CLK1	SEL is active high, disabled output is high.
	BUFGCECLK0	CLK0	0	SEL is active low, disabled output is low.
	BUFGCECLK0_1	CLK0	1	SEL is active low, disabled output is high.
	BUF0	CLK0	CLK0	Buffer for CLK0
	BUF1	CLK1	CLK1	Buffer for CLK1
SELFORCE = 1	Non-Glitchless	CLK0	CLK1	—

8.3. DCS Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
COMPONENT DCS
GENERIC(DCSMODE : string := “DCS”);
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 "DCS";
I1: DCS
generic map(
   DCSMODE => “DCS”)
port map (
   CLK0 => CLK0
  ,CLK1 => CLK1
  ,SEL => SEL
  ,SELFORCE => SELFORCE
  ,DCSOUT => DCSOUT);
```

8.4. 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 = "DCS";
DCS DCSInst0 (
.CLK0     (CLK0),
.CLK1     (CLK1),
.SEL      (SEL),
.SELFORCE  (SELFORCE),
.DCSOUT   (DCSOUT));
```

9. Dynamic Clock Control (DCC)

The Nexus device has a Dynamic Clock Control feature which allows internal logic to dynamically enable or disable the region primary clock network. This gating function does not create glitches or increase 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.

The DCC controls the clock sources from the Primary CLOCK MIDMUX before they are fed to the Primary Center MUXs that drive the region clock network. When a clock network is disabled, the power consumption of all the associated logic is greatly reduced.

The Nexus device clock architecture allows both DCC and DCS to function at the same time. Care must be taken when the clock source is used as input to the PLL. The DCC should remain enabled, otherwise if the PLL input clock stops toggling, the PLL loses locked and the PLL output clock also stops toggling.

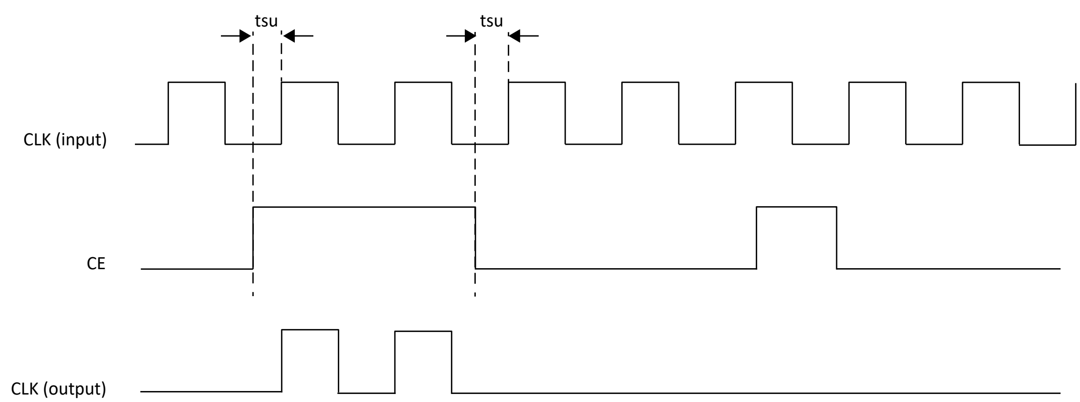

Figure 9.1. Glitchless DCC Functional Waveform

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

9.1. Component Definition

The DCC component can be instantiated in the source code of a design as defined in this section. Figure 9.2, Table 9.1, and Table 9.2 show the DCC definitions.

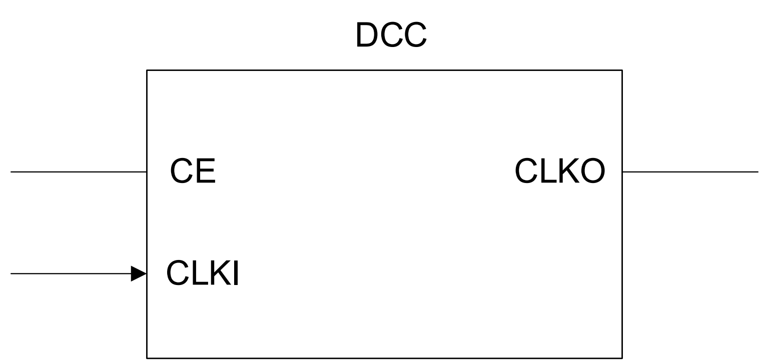

Figure 9.2. DCC Component Symbol

Table 9.1. DCC Component Port Definition

Port Name	I/O	Description
CLKI	I	Clock Input port.
CE	I	Clock Enable port — CE = 0 CLKO is disabled (CLKO = '0') — CE = 1 CLKO is enabled (CLKO = CLKI)
CLKO	O	Clock Output Port

Table 9.2. DCC Component Attribute Definition

Name	Value	Default	Description
DCCEN	0 1	0	Enables dynamic control. "0": CLKO = CLKI regardless of the CE input. "1": CLKO depends on the CE input.

9.2. DCC Usage in VHDL

Component Instantiation

```
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;
```

DCC Instantiation

```
I1: DCC
port map (
   CLKI  => CLKI,
  CE    => CE,
  CLKO  => CLKO);
DCC Usage in Verilog
Component and Attribute Declaration
module DCC(CLKI,CE,CLKO);
input   CLKI;
input   CE;
output  CLKO;
endmodule
```

9.3. DCC Usage in Verilog

DCC Instantiation

```
DCC DCSInst0 (
.CLKI (CLKI),
.CE   (CE),
.CLKO (CLKO));
```

10. Internal Oscillator (OSCA, OSCD)

The OSCA component (for Nexus devices, except the LIFCL-33U device) and OSCD component (for the LIFCL-33U device) perform multiple functions on the Nexus device. The OSCA and OSCD components are used for configuration, SED, as well as optionally in user mode. In user mode, the OSCA and OSCD components have the following features:

- It permits a design to be fully self-clocked, as long as the quality of the OSCA component silicon-based oscillator is adequate.
- If it is unused, it can be turned off for power savings through the Power Controller IP Module.
- It has an input to dynamically control standby/normal operation.
- It has a direct connection to primary clock routing through the top MIDMUX. For CertusPro-NX, the right MIDMUX can also be used for the direct connection to primary clock routing.
- It can be configured for operation at a wide range of frequencies through the configuration bits.

10.1. OSCA Component Definition

The OSCA component can be instantiated in the source code of a design as defined in this section. Figure 10.1 and Table 10.1 show the OSCA definitions.

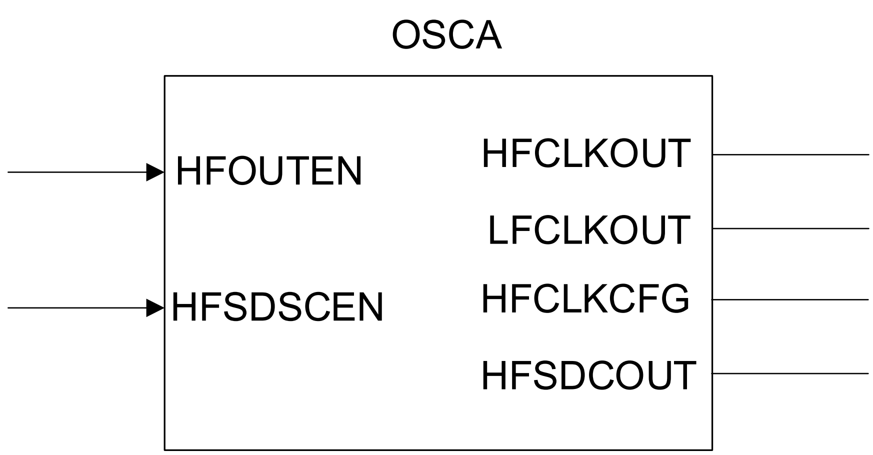

Figure 10.1. OSCA Component Symbol

Table 10.1. OSCA Component Port Definition

Port Name	I/O	Description
HFOUTEN	I	High Frequency User Clock Output Enable
HFSDSCEN	I	High Frequency User Clock Output Enable
HFCLKOUT	O	450 MHz with Programmable Divider (2~256) to User
HFSDCOUT	O	450 MHz with Programmable Divider (2~256) to User for SED/SEC Application
LFCLKOUT	O	Low Frequency Clock Output; 32 kHz
HFCLKCFG	O	High Frequency Reference Clock; 450 MHz

Table 10.2. OSCA Component Attribute Definition

Name	Value	Default	Description
HF_CLK_DIV	00000001 ~ 11111111	00000001	User-assignable HF oscillator output divider configuration (div2~div256)
HF_SED_SEC_DIV	00000001 ~ 11111111	00000001	User-assignable HF oscillator output divider configuration (div2~div256)
HF_OSC_EN	DISABLED ENABLED	DISABLED	HF oscillator enable, controlled by the user
LF_OUTPUT_EN	DISABLED ENABLED	DISABLED	Low frequency clock output enable

10.2. OSCA Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
Component OSCA
generic (
       HF_CLK_DIV : string;
       HF_SED_SEC_DIV : string;
       HF_OSC_EN : string;
       LF_OUTPUT_EN : string
           )
port (
  HFOUTEN : in std_logic;
  HFSDSCEN : in std_logic;
  HFCLKOUT : out std_logic;
  LFCLKOUT : out std_logic;
  HFCLKCFG : out std_logic;
  HFSDCOUT : out std_logic
);
```

OSCA Instantiation

```
I1: OSCA
generic map (
  HF_CLK_DIV : "1", --(DIV = 2)
  HF_SED_SEC_DIV : "1", --(DIV = 2)
  HF_OSC_EN : "ENABLED",
  LF_OUTPUT_EN : "ENABLED"
            )
port map (
  HFOUTEN => HFOUTEN,
  HFSDSCEN => HFSDSCEN,
  HFCLKOUT => HFCLKOUT,
  LFCLKOUT => LFCLKOUT,
  HFCLKCFG => HFCLKCFG,
  HFSDCOUT =>HFSDCOUT
);
```

10.3. OSCA Usage in Verilog

OSCA Instantiation

```
OSCA I1 #(
.HF_CLK_DIV ("1"), //DIV = 2
.HF_SED_SEC_DIV ("1"), //DIV = 2
.HF_OSC_EN ("ENABLED"),
.LF_OUTPUT_EN ("ENABLED"),
)(
.HFOUTEN (HFOUTEN ),
.HFSDSCEN (HFSDSCEN),
.HFCLKOUT (HFCLKOUT),
.LFCLKOUT (LFCLKOUT),
.HFCLKCFG (HFCLKCFG),
.HFSDCOUT (HFSDCOUT)
);
```

10.4. OSCD Component Definition

The OSCD component can be instantiated in the source code of a design as defined in this section. Figure 10.2 and Table 10.3 show the OSCD definitions.

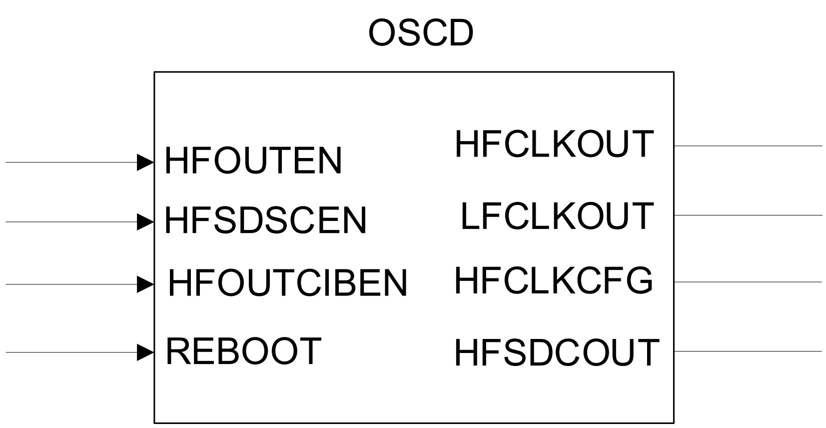

Figure 10.2. OSCD Component Symbol

Table 10.3. OSCD Component Port Definition

Port Name	I/O	Description
HFOUTEN	I	High Frequency User Clock Output Enable
HFSDSCEN	I	High Frequency User Clock Output Enable
HFOUTCIBEN	I	CIB control to turn on and off HFCLKOUT output in user mode
REBOOT	I	CIB control to enable and disable HFCLKCFG output in user mode
HFCLKOUT	O	450 MHz with Programmable Divider (2~256) to User
HFSDCOUT	O	450 MHz with Programmable Divider (2~256) to User for SED/SEC Application
LFCLKOUT	O	Low Frequency Clock Output; 32 kHz
HFCLKCFG	O	High Frequency Reference Clock; 450 MHz

Table 10.4. OSCD Component Attribute Definition

Name	Value	Default	Description
DTR_EN	DISABLED ENABLED	DISABLED	DTR block enable
HF_CLK_DIV	00000001 ~ 11111111	00000001	User-assignable HF oscillator output divider configuration (div2~div256)
HF_SED_SEC_DIV	00000001 ~ 11111111	00000001	User-assignable HF oscillator output divider configuration (div2~div256)
HF_FABRIC_EN	DISABLED ENABLED	DISABLED	HF oscillator fabric trim source enable
HF_OSC_EN	DISABLED ENABLED	DISABLED	HF oscillator enable, controlled by the user
LF_FABRIC_EN	DISABLED ENABLED	DISABLED	Low frequency clock fabric trim source enable
LF_OUTPUT_EN	DISABLED ENABLED	DISABLED	Low frequency clock output enable
DEBUG_N	DISABLED ENABLED	DISABLED	Ignore or enable the SLEEP/STOP function in USER mode. 0: Ignore the SLEEP/STOP mode

10.5. OSCD Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
Component OSCD
generic (
       DTR_EN : string;
       HF_CLK_DIV : string;
       HF_SED_SEC_DIV : string;
       HF_FABRIC_EN : string;
       HF_OSC_EN : string;
       LF_FABRIC_EN : string
       LF_OUTPUT_EN : string
       DEBUG_EN : string
            )
port (
  HFOUTEN : in std_logic;
  HFSDSCEN : in std_logic;
  HFOUTCIBEM : in std_logic;
  REBOOT : in std_logic;
  HFCLKOUT : out std_logic;
  LFCLKOUT : out std_logic;
  HFCLKCFG : out std_logic;
  HFSDCOUT : out std_logic
);
```

OSCD Instantiation

```
I1: OSCD
generic map (
  DTR_EN : “DISABLE”,
  HF_CLK_DIV : "1", --(DIV = 2)
```

```
  HF_SED_SEC_DIV : "1", --(DIV = 2)
  HF_FABRIC_EN : "DISABLE",
  HF_OSC_EN : "ENABLED",
  LF_FABRIC_EN : "DISABLE",
  LF_OUTPUT_EN : "ENABLED",
  DEBUG_N : "1"
            )
port map (
  HFOUTEN => HFOUTEN,
  HFSDSCEN => HFSDSCEN,
  HFOUTCIBEN => HFOUTCIBEN,
  REBOOT => REBOOT,
  HFCLKOUT => HFCLKOUT,
  LFCLKOUT => LFCLKOUT,
  HFCLKCFG => HFCLKCFG,
  HFSDCOUT =>HFSDCOUT
);
```

10.6. OSCD Usage in Verilog

OSCD Instantiation

```
OSCD I1 #(
.DTR_EN ("DISBALE"),
.HF_CLK_DIV ("1"), //DIV = 2
.HF_SED_SEC_DIV ("1"), //DIV = 2
.HF_FABRIC_EN ("DISABLE"),
.HF_OSC_EN ("ENABLED"),
.LF_FABRIC_EN ("DISABLE"),
.LF_OUTPUT_EN ("ENABLED"),
.DEBUG_N ("1")
)(
.HFOUTEN (HFOUTEN ),
.HFSDSCEN (HFSDSCEN),
.HFOUTCIBEN (HFOUTCIBEN),
.REBOOT (REBOOT),
.HFCLKOUT (HFCLKOUT),
.LFCLKOUT (LFCLKOUT),
.HFCLKCFG (HFCLKCFG),
.HFSDCOUT (HFSDCOUT)
);
```

11. Edge Clocks

Each Nexus 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
- Bottom PLL outputs (CLKOP, CLKOS, CLKOS2, CLKOS3, CLKOS4, and CLKOS5)
- ECLK Bridge
- Internal nodes

The Nexus device has Edge Clock (ECLK) at the bottom of the device. There are four ECLK networks per I/O bank. ECLK Input MUX collects all clock sources available as shown in Figure 11.1. 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.

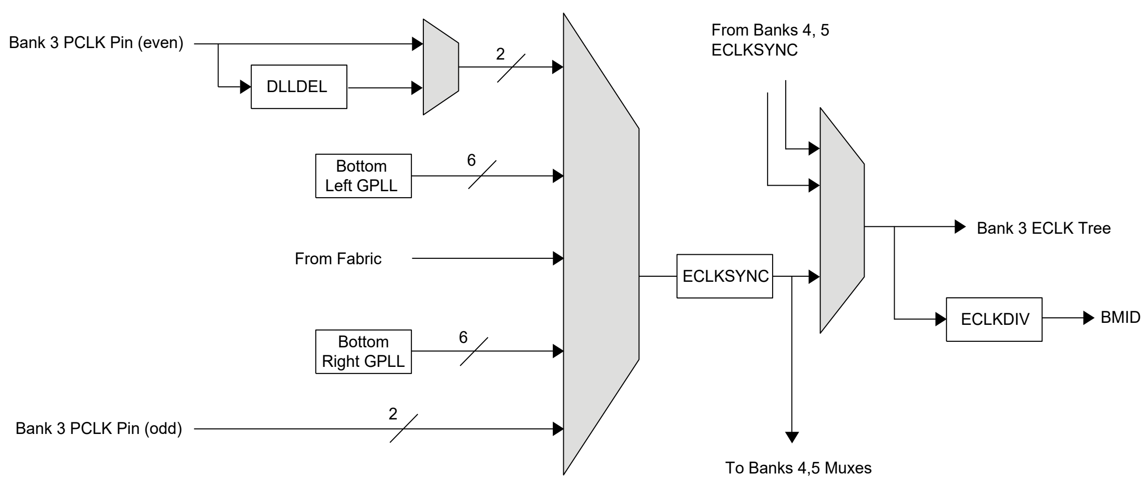

Figure 11.1. Edge Clock Sources Per Bank

11.1. Edge Clock Dividers (ECLKDIV)

There are twelve Edge Clock dividers available in the Nexus 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.

11.2. ECLKDIV Component Definition

The ECLKDIV component can be instantiated in the source code of a design as defined in this section. Figure 11.2, Table 11.1, and Table 11.2 define the ECLKDIV component. Verilog and VHDL instantiations are included.

Figure 11.2. ECLKDIV Component Symbol

Table 11.1. ECLKDIV Component Port Definition

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

Table 11.2. ECLKDIV Component Attribute Definition

Name	Value	Default	Description
GSR	ENABLED DISABLED	ENABLED	GSR ENABLE/DISABLE Selection
ECLK_DIV	DISABLE “2” “3P5” “4” “5”	DISABLE	ECLK DIVIDE Ratio selection (“3P5” = 3.5)

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

11.3. ECLKDIV Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
component ECLKDIV
Generic (ECLK_DIV      : string;
         GSR     : 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 GSR : string;
attribute GSR of I1 : label is "DISABLED";

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

11.4. ECLKDIV Usage in Verilog

Component and Attribute Declaration

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

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

input  DIVRST, ECLKIN, SLIP;
output DIVOUT;
endmodule
```

ECLKDIV Instantiation

```
defparam I1.ECLK_DIV = "2.0";
defparam I1.GSR = "DISABLED";
ECLKDIV I1 (
   .DIVRST      (DIVRST),
   .ECLKIN     (ECLKIN),
   .SLIP  (SLIP),
   .DIVOUT    (DIVOUT));
```

12. Edge Clock Synchronization (ECLKSYNC)

Nexus 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 the user 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.

12.1. 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 12.1 and Table 12.1 show the ECLKSYNC component definition.

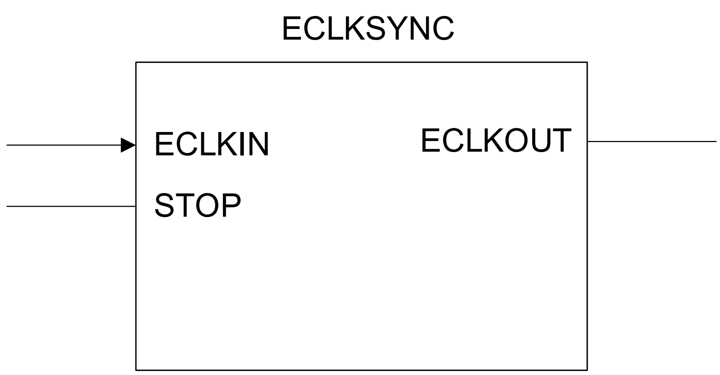

Figure 12.1. ECLKSYNC Component Symbol

Table 12.1. ECLKSYNC Component Port Definition

Port Name	I/O	Description
ECLKIN	I	Clock Input port.
STOP	I	Control signal to stop Edge Clock • STOP = 0 Clock is Active • STOP = 1 Clock is Off
ECLKOUT	O	Clock Output Port

Table 12.2. ECLKSYNC Component Attribute Definition

Name	Value	Default	Description
STOP_EN	DISABLE ENABLE	DISABLE	STOP ENABLE/DISABLE Selection

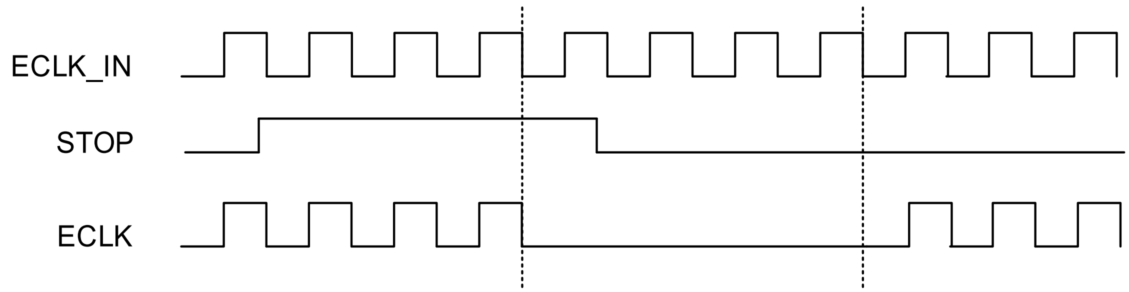

Figure 12.2. ECLKSYNC Functional Waveform

12.2. 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);
```

12.3. ECLKSYNC Usage in Verilog

Component and Attribute Declaration

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

ECLKSYNC Instantiation

```
ECLKSYNC ECLKSYNCInst0 (
 .ECLKIN  (ECLKIN),
 .STOP   (STOP),
 .ECLKOUT (ECLKOUT));
```

13. General Routing for Clocks

The Nexus device architecture supports the ability to use 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 Nexus device fabric. In this case, it is recommended to add a preference for this gated clock to use primary routing.

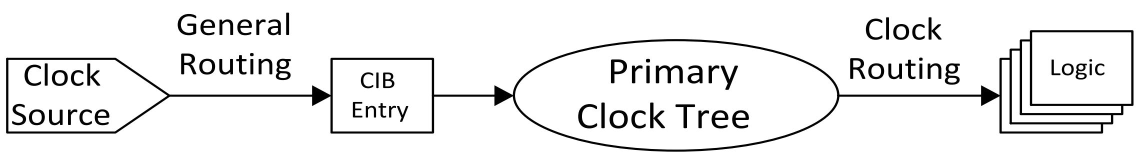

Figure 13.1. Gated Clock to the Primary Clock Routing

For a very small clock domain, user can limit the distance of a general routing based (gated) clock to one PLC in distance to the logic it clocks. The user must group this logic (UGROUP) with a *BBOX* (see Lattice Radiant Help > Constraints Reference Guide > Preferences > UGROUP) and specify a *PROHIBIT PRIMARY* on the generated clock. The *PROHIBIT_PRIMARY* constraint allows the pin to be used as a clock source while the *BBOX* constraint is also included to ensure that timing closure can be obtained even without using a dedicated PCLK pin. If the software cannot place the logic tree within the *BBOX*, an error occurs.

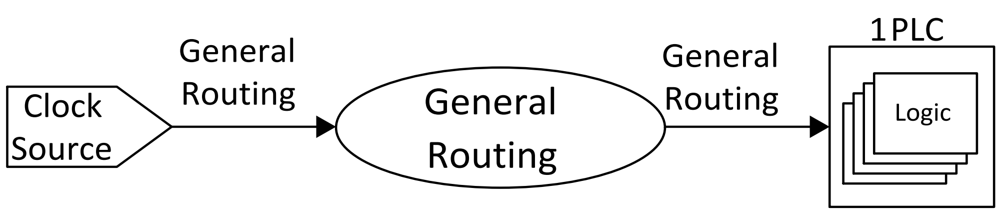

Figure 13.2. Gated Clock to Small Logic Domain

14. sysCLOCK PLL

14.1. sysCLOCK PLL Overview

The sysCLOCK™ PLLs can be used in a variety of clock management applications such as clock injection delay removal, clock phase adjustment, clock timing adjustment, and frequency synthesis (multiplication and division of a clock). The PLL supports Fractional-N synthesis. The Nexus 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. See Appendix A for more details.
- Primary Clock Routing
- Edge Clock Routing
- FPGA Fabric

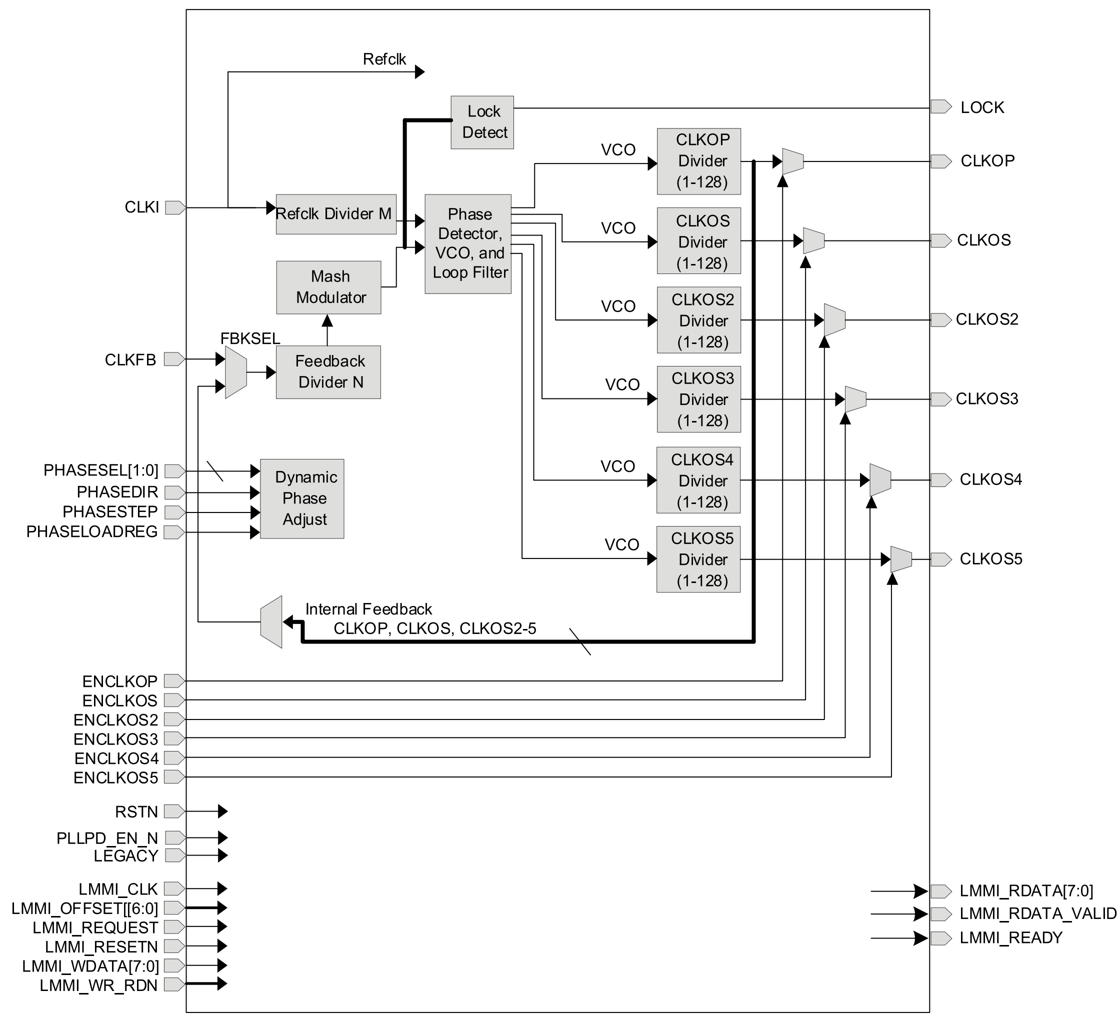

Figure 14.1. Nexus PLL Block Diagram

There are three PLLs for LIFCL-40, LFD2NX-40, and LFD2NX-28 at three corners as Upper Left, Lower Left and Lower Right, two PLLs for LIFCL-17, LFD2NX-17, and LFD2NX-9 at two corners as Lower Left and Lower Right corners. There are three PLLs for LFCPNX-50 at three corners as Upper Left, Lower Left and Lower Right, four PLLs for LFCPNX-100 at four corners as Upper Left, Upper Right, Lower Left and Lower Right. There are two PLLs for LFMXO5 at two corners as Upper Left and Lower Right. There is one PLL for LIFCL-33 at Lower Left corner. Each PLL has six outputs. All six PLL outputs can feed the Primary Clock and Edge Clock networks.

14.2. PLL Features

14.2.1. Dedicated PLL Inputs

Every PLL has a dedicated low skew input (PLLCK) 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. In each corner of one Nexus device, there is one PLL at most. Each PLL on the Nexus device has one pair of dedicated PLL input pins. For details on changing the clock routing preference, refer to *Section 8.3. ldc_set_location Constraint* of the Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059).

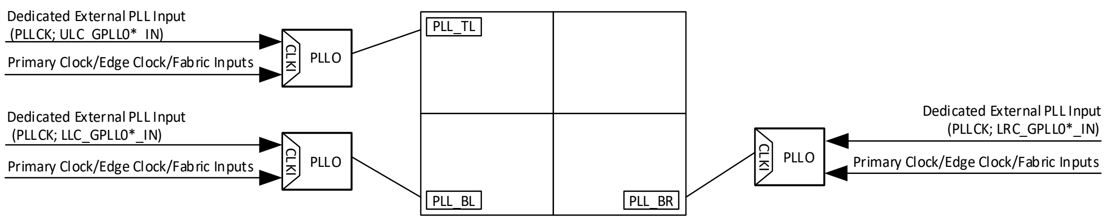

Figure 14.2. PLL Input Pins for LIFCL-40, LFD2NX-40, and LFD2NX-28

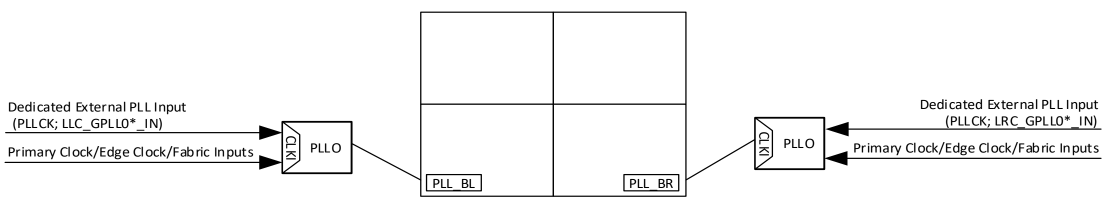

Figure 14.3. PLL Input Pins for LIFCL-17, LFD2NX-17, and LFDD2NX-9

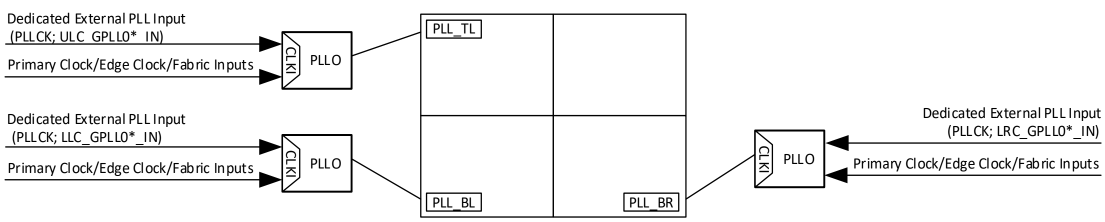

Figure 14.4. PLL Input Pins for LFCPNX-50

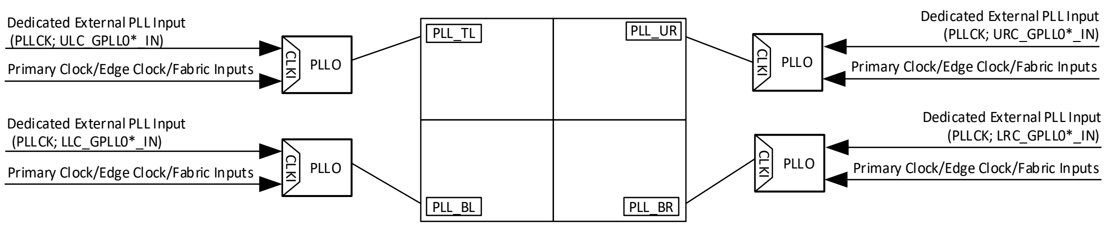

Figure 14.5. PLL Input Pins for LFCPNX-100 and LFMXO5-55TDQ

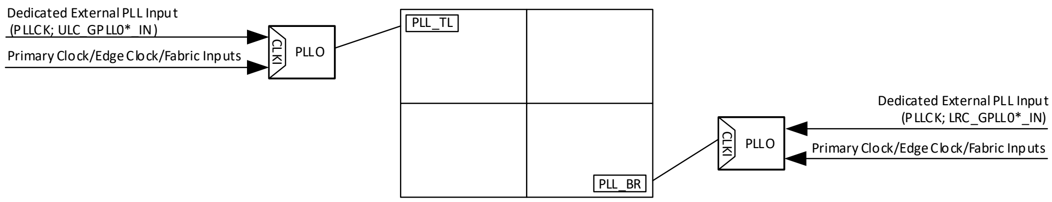

Figure 14.6. PLL Input Pins for LFD2NX-15/25/35/65 and LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T

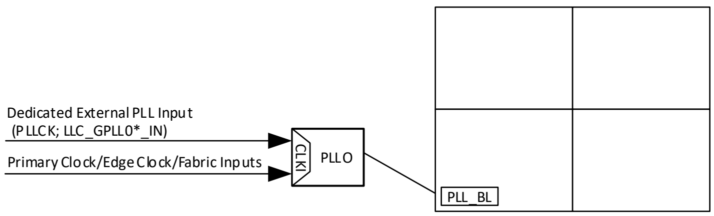

Figure 14.7. PLL Input Pins for LIFCL-33 and LIFCL-33U

14.2.2. 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 the clock path delay which benefits system synchronous input and output timing. This feature is performed by aligning the PLL 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.

14.2.3. 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 Nexus 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.

14.2.4. Frequency Synthesis

The PLL can be used to multiply up or divide down an input clock.

14.2.5. Legacy Mode (Standby)

In addition to the major features, the PLL has a Legacy Mode to reduce power. The Legacy Mode was called PLL standby mode. But due to the new proposed schemed for Nexus PLLs, it is given a different name to differentiate with the new STDBY mode. The Legacy Mode 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.

The user 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.

14.3. sysCLOCK PLL Component Definition

The PLL component can be instantiated in the source code of a design as defined in this section. Figure 14.8 and Table 14.1 show the definitions.

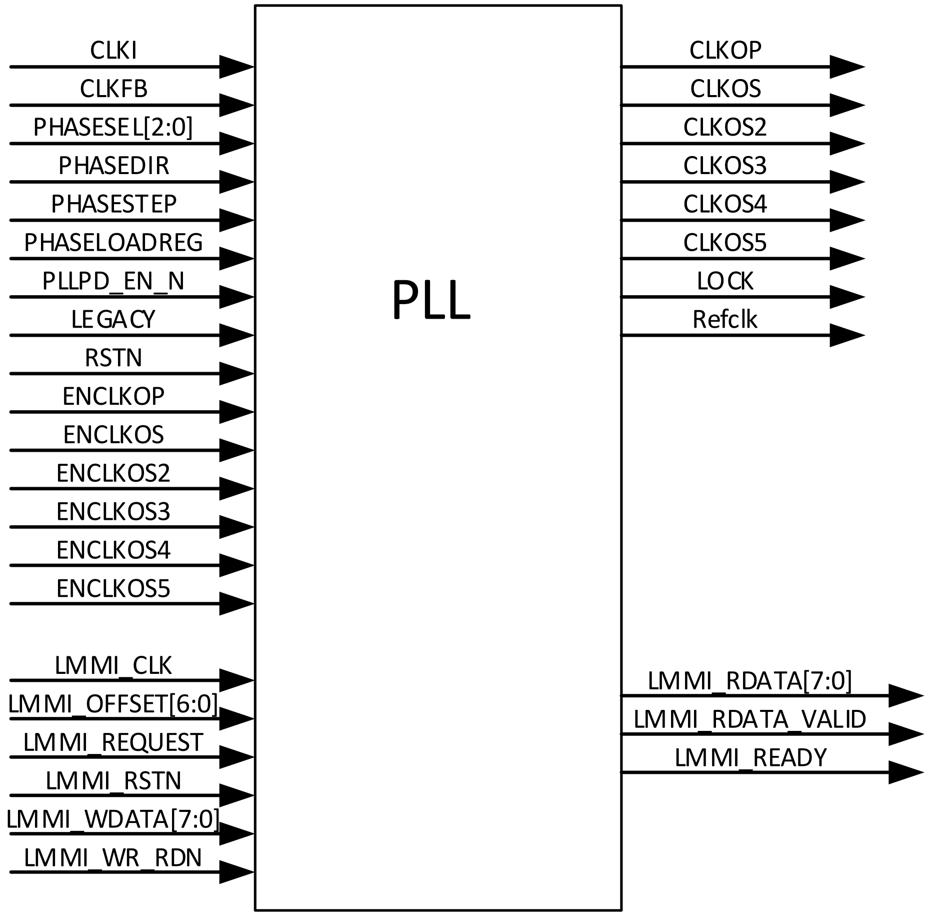

Figure 14.8. PLL Component Instance

Table 14.1. PLL Component Port Definition

Signal	I/O	Description
CLKI	I	Input Clock to PLL.
CLKFB	I	Feedback Clock.
PHASESEL[2:0]	I	Select the output affected by Dynamic Phase adjustment.
PHASEDIR	I	Dynamic Phase adjustment direction.
PHASESTEP	I	Dynamic Phase adjustment step.
PHASELOADREG	I	Not used. Tie low
PLLPD_EN_N	I	Standby signal to power down the PLL.
LEGACY	I	Power mode setting to enable legacy mode
RST	I	Resets the entire PLL.
ENCLKOP	I	Enable PLL output CLKOP.
ENCLKOS	I	Enable PLL output CLKOS.
ENCLKOS2	I	Enable PLL output CLKOS2.
ENCLKOS3	I	Enable PLL output CLKOS3.
ENCLKOS4	I	Enable PLL output CLKOS4.
ENCLKOS5	I	Enable PLL output CLKOS5.
CLKOP	O	PLL main output clock.
CLKOS	O	PLL output clock.
CLKOS2	O	PLL output clock2.
CLKOS3	O	PLL output clock3.
CLKOS4	O	PLL output clock4.
CLKOS5	O	PLL output clock5.
LOCK	O	Indicates PLL is now locked to CLKI, Asynchronous signal. Active high indicates PLL lock.
Refclk	O	Output of Reference clock.
LMMI_CLK	I	CIB LMMI interface clock
LMMI_OFFSET[6:0]	I	CIB LMMI interface address offset (LSB of address bus)
LMMI_REQUEST	I	CIB LMMI interface request signal
LMMI_RESETN	I	CIB LMMI interface reset, active low
LMMI_WDATA[7:0]	I	CIB LMMI interface write data
LMMI_WR_RDN	I	CIB LMMI interface Write/Read control; 1=write, 0=read.
LMMI_RDATA[7:0]	O	CIB LMMI interface read data
LMMI_RDATA_VALID	O	CIB LMMI interface read data valid signal
LMMI_READY	O	CIB LMMI interface ready signal

14.4. Functional Description

14.4.1. Refclk (CLKI) Divider

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

14.4.2. 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 user selected in the software.

14.4.3. Output Clock Dividers (CLKOP, CLKOS, CLKOS2, CLKOS3, CLKOS4, CLKOS5)

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.

14.4.4. 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 Nexus devices, 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.

14.4.5. Phase Adjustment (Dynamic Mode)

The phase adjustments can also be controlled in a dynamic mode using the PHASESEL, PHASEDIR, and PHASESTEP ports. 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.

14.5. PLL Inputs and Outputs

14.5.1. 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. If the optional reset port is added to the PLL, the reference clock must be stable before the RST signal is deasserted.

14.5.2. CLKFB Input

The CLKFB signal is the feedback signal to the PLL. The feedback signal is used by the Phase 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/S/S2/S3/S4/S5]) to remove the primary clock routing injection delay from an internal PLL connection (feedback mode = INT_O[P/S/S2/S3/S4/S5]) for simple feedback. The feedback clock signal is 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.

14.5.3. RST Input

At power-up, an internal power-up reset signal from the configuration block resets the PLL. Additionally, an active high, asynchronous, user-controlled reset port can be optionally added to the PLL. 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 logic *0*. In bypass mode, the output does not reset. If the optional reset pot is added to the PLL, the reference clock must be stable before the RST signal is deasserted.

After the RST signal is deasserted, the PLL starts the lock-in process and takes t_{LOCK} time to complete. Figure 14.9 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.

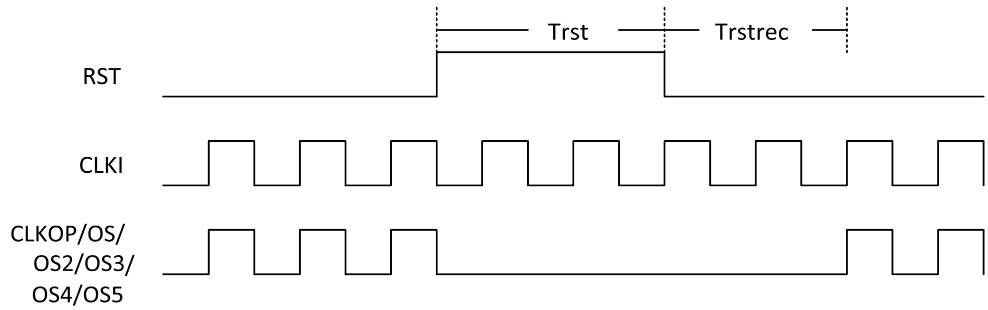

Figure 14.9. RST Input Timing Diagram

14.5.4. 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 14.2. PLL Clock Output Enable Signal List

Clock Enable Signal Name	Corresponding PLL Output	IP Catalog Option Name
ENCLKOP	CLKOP	"Clock Enable OP"
ENCLKOS	CLKOS	"Clock Enable OS"
ENCLKOS2	CLKOS2	"Clock Enable OS2"
ENCLKOS3	CLKOS3	"Clock Enable OS3"
ENCLKOS4	CLKOS4	"Clock Enable OS4"
ENCLKOS5	CLKOS5	"Clock Enable OS5"

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

14.5.5. 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 user 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. The user 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.

14.5.6. Dynamic Phase Shift Inputs

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

14.5.7. 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 is signals are pulsed. The PHASESEL signal is optional and is available if user select the *Enable Dynamic Phase Ports* option in IP Catalog Wizard.

Table 14.3. PHASESEL Signal Settings Definition

PHASESEL[2:0]	PLL Output Shifted
000	CLKOP
001	CLKOS
010	CLKOS2
011	CLKOS3
100	CLKOS4
101	CLKOS5

14.5.8. 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 is pulsed. The PHASEDIR signal is optional and is available if user select the *Enable Dynamic Phase Ports* option in IP Catalog Wizard.

Table 14.4. PHASEDIR Signal Settings Definition

PHASEDIR	Direction
0	Delayed (lagging)
1	Advanced (leading)

14.5.9. 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 VCO cycles. The PHASESTEP signal is optional and is available if user select the *Enable 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.

14.5.10. PLL Clock Outputs

The PLL has six outputs, listed in Table 14.5. All six outputs can be routed to the Primary clock routing of the FPGA. All six 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 14.5. PLL Clock Outputs and ECLK Connectivity

Clock Output Name	Edge Clock Connectivity	Selectable Output
CLKOP	ECLK Connection	Always Enabled
CLKOS	ECLK Connection	Selectable through IP Catalog
CLKOS2	No ECLK Connection	Selectable through IP Catalog
CLKOS3	No ECLK Connection	Selectable through IP Catalog
CLKOS4	No ECLK Connection	Selectable through IP Catalog
CLKOS5	No ECLK Connection	Selectable through IP Catalog

14.5.11. LOCK Output

The LOCK output provides information about the status of the PLL. After the device is powered up and the input clock is stable, the PLL achieves lock within t_{LOCK} time. 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 user select the Provide PLL Lock Signal option in IP Catalog Wizard.

14.6. Dynamic Phase Adjustment

Dynamic phase adjustment of the PLL output clocks can be done 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 14.3 shows the output clock selection settings available for the PHASESEL[2:0] signal. The PHASESEL signal must be stable for 5 ns before the PHASESTEP is pulsed.

The selected output clock phase is either advanced or delayed depending upon the value of the PHASEDIR port. Table 14.4 shows the PHASEDIR settings available. The PHASEDIR signal must be stable for 5 ns before the PHASESTEP is pulsed.

14.6.1. VCO Phase Shift

When the PHASESEL and PHASEDIR have been set, a VCO phase adjustment is made by toggling the PHASESTEP signal. Each pulse of PHASESTEP shifts output clock (as selected by PHASESEL) by 1/8 of the VCO period, forward or backward (as selected by PHASEDIR). The resulting phase shift to the PLL output clock cycle is represented by the following equation:

Output clock phase change per step = [1/(8 × (DIVO<n>_ACTUAL_STR + 1))] × 360°

Where <n> is the clock output specified by PHASESEL (CLKOP/OS/OS2/OS3/OS4/OS5). Values for DIVO<n>_ACTUAL_STR are located in the HDL source file generated by IP Catalog Wizard.

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.

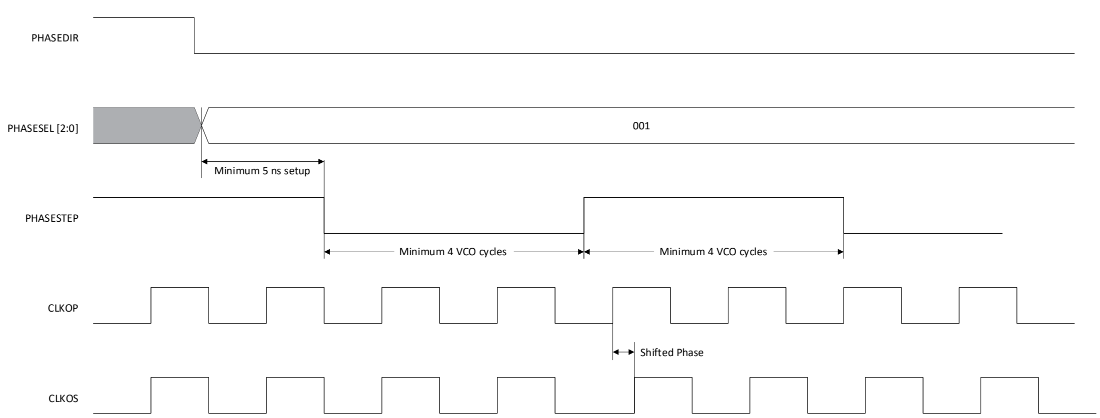

Figure 14.10. PLL Phase Shifting Using the PHASESTEP Signal

For Example:

PHASESEL[2:0]=3'b001 to select CLKOS for phase shift

PHASEDIR =1'b0 for selecting delayed (lagging) phase

Assume the output is divided by 2, DIVOS_ACTUAL_STR = 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).

14.7. Fractional-N Synthesis Operation

The Nexus PLL supports high resolution (12-bit) fractional-N synthesis through Radiant IP Catalog. 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 IP Catalog user interface by checking the *Enable fractional-N Divider* box under the *General* tab with the *Configuration Mode* set in either *Frequency* mode or *Divider* mode. When enabled, Fractional-N synthesis is applied to all active PLL outputs.

In *Frequency* configuration mode, user needs to set the CLCKI Frequency (10 MHz – 800 MHz) and the desired output Frequency Desired Value (6.25 MHz – 800 MHz) with reasonable Tolerance (0.5% - 10%). Then the Feedback Divider Actual Value (integer), Feedback Divider Actual Value (Fractional) and the clock output *Divider Actual Value* is automatically set. The *Frequency Actual Value* and *ERROR* (PPM) are automatically calculated.

In *Divider* configuration mode, user needs to set the CLKI *Frequency*, CLKI *Divider Desired Value*, CLKFB *FBK Divider Desired Value (Integer)*, CLKFB *FBK Divider Desired Value (Fractional)* and the clock output *Divider Desired Value*. Then, the output clock *Frequency Actual Value* is calculated automatically in the user interface.

The output frequency is given by the equation:

$$F_{out} = \frac{F_{CLKI}}{\mathrm{M} \times \mathrm{O}} \times (\mathrm{N} + \frac{\mathrm{F}}{4096})$$

Where:

F_{out} is the output *Frequency Actual Value*.

F_{CLKI} is the CLKI input frequency.

M is the CLKI *Divider Desired Value*.

N is the CLKFB *FBK Divider Desired Value (Integer)*.

F is the CLKFB *FBK Divider Desired Value (Fractional)*.

O is the output *Divider Actual Value.*

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 results 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.

14.8. Spread Spectrum Clock Generation

The Nexus PLL supports Spread spectrum clock generation through Radiant IP Catalog. The spread spectrum function is integrated with the Fractional-N controls and supports *Centered Spread* or *Down Spread*, triangle wave, 0.25% per step from 0.25% to 2.00% with modulation frequency range from 24.42 kHz to 200 kHz. The Spread Spectrum Clock Generation is enabled in the IP Catalog user interface by checking the *Enable Spread Spectrum Clock Generation* box under the *General* tab. In the *Spread Spectrum* section, select *Spread Spectrum Profile* for *Centered Spread* or *Down Spread*, set the *Triangle Modulation Depth* (0.25% – 2.00% with 0.25% step size) and the *Desired Modulation Frequency* (24.42 kHz – 200 kHz). When enabled, spread spectrum characteristics is applied to all active PLL outputs. Figure 14.11 and Figure 14.12 show the spread spectrum profiles.

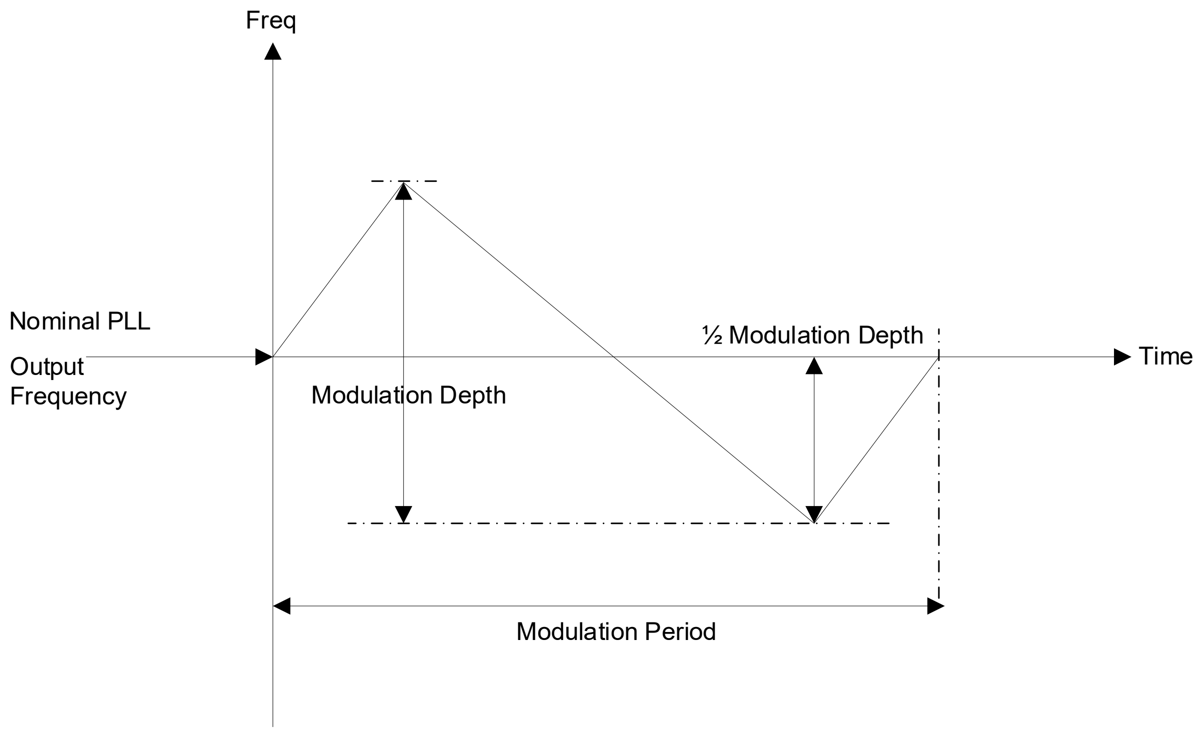

Figure 14.11. Center Spread Profile

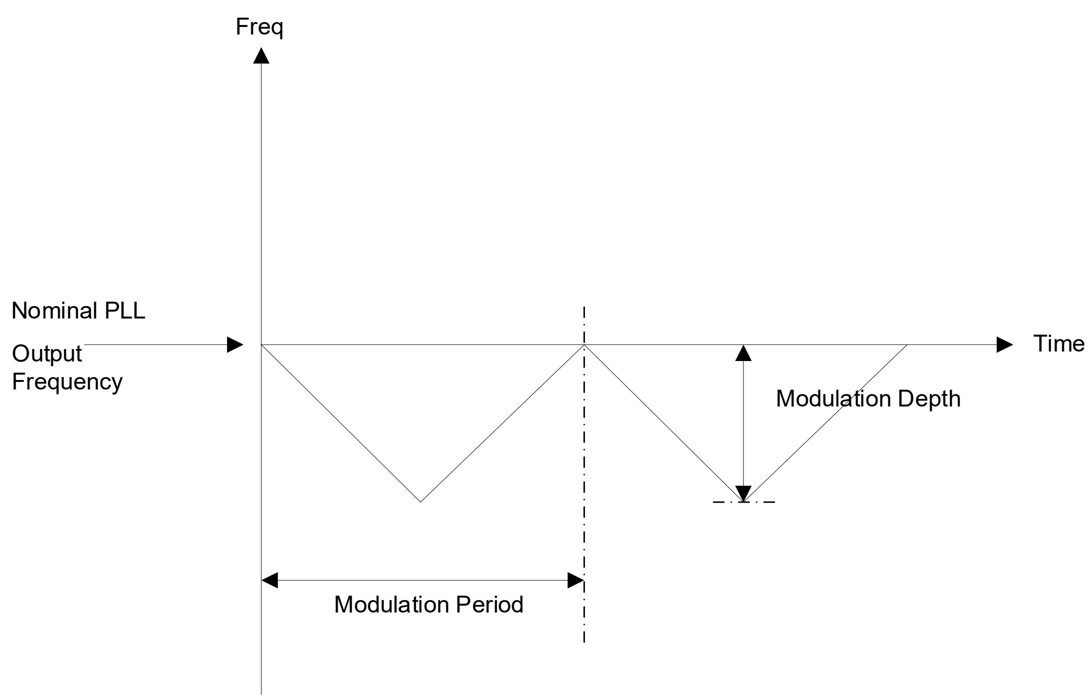

Figure 14.12. Down Spread Profile

14.9. Low Power Features

The Nexus PLL contains several features that allows user to reduce the power usage of a design including Standby mode support and Dynamic clock enable.

14.9.1. Dynamic Clock Enable

The Dynamic Clock Enable feature allows user 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 the associated clocks. 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.

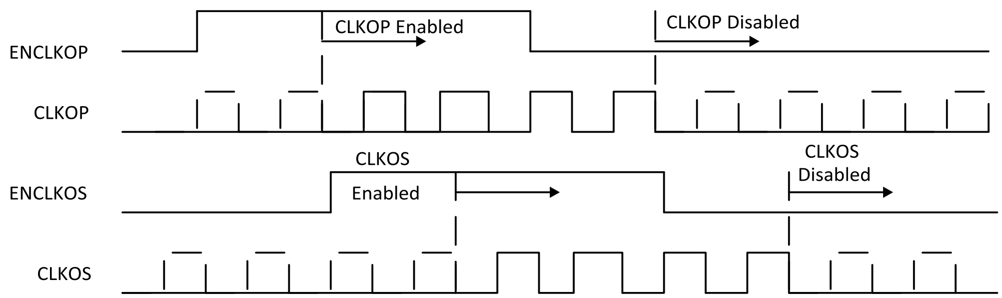

Figure 14.13. Dynamic Clock Enable for PLL Outputs

14.10. 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 14.14. 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 14.14.

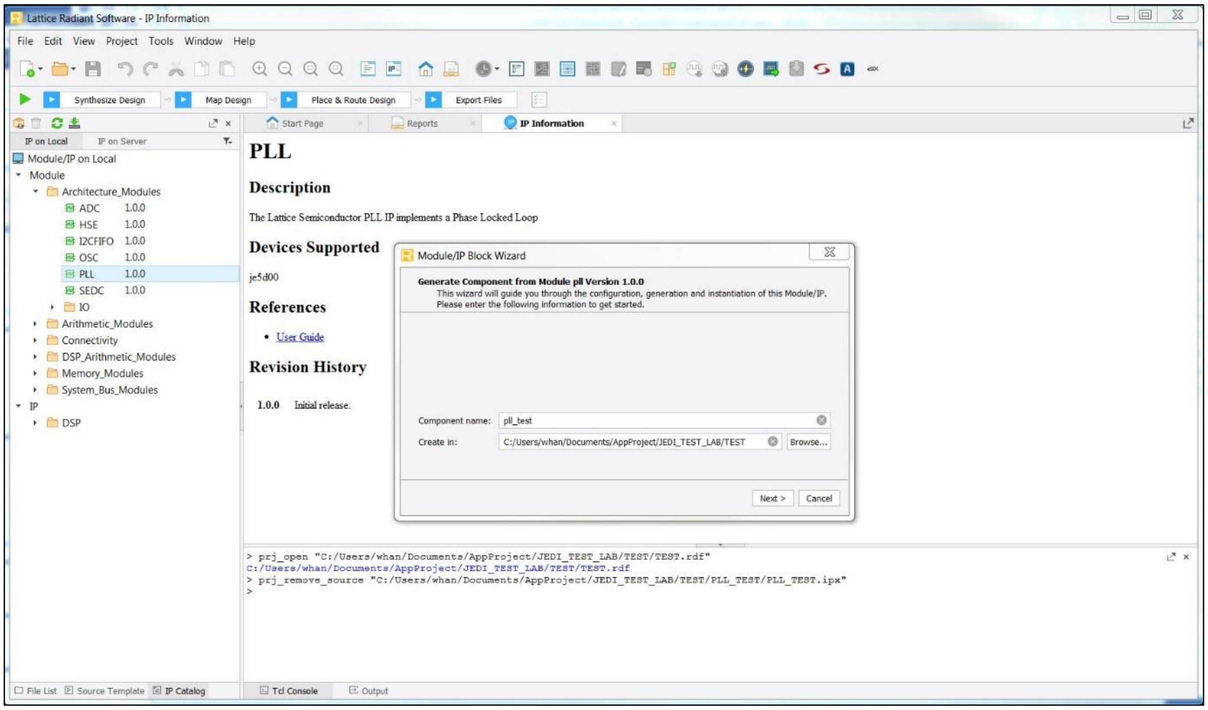

Figure 14.14. IP Catalog Main Window for PLL Module

14.10.1. Configuration Tab

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

14.10.2. 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. The user can also select a tolerance value from the *Tolerance %* drop-down box.

If required, enter the desired phase shift and click the Calculate button. The software calculates the closest achievable phase shift and displays it 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

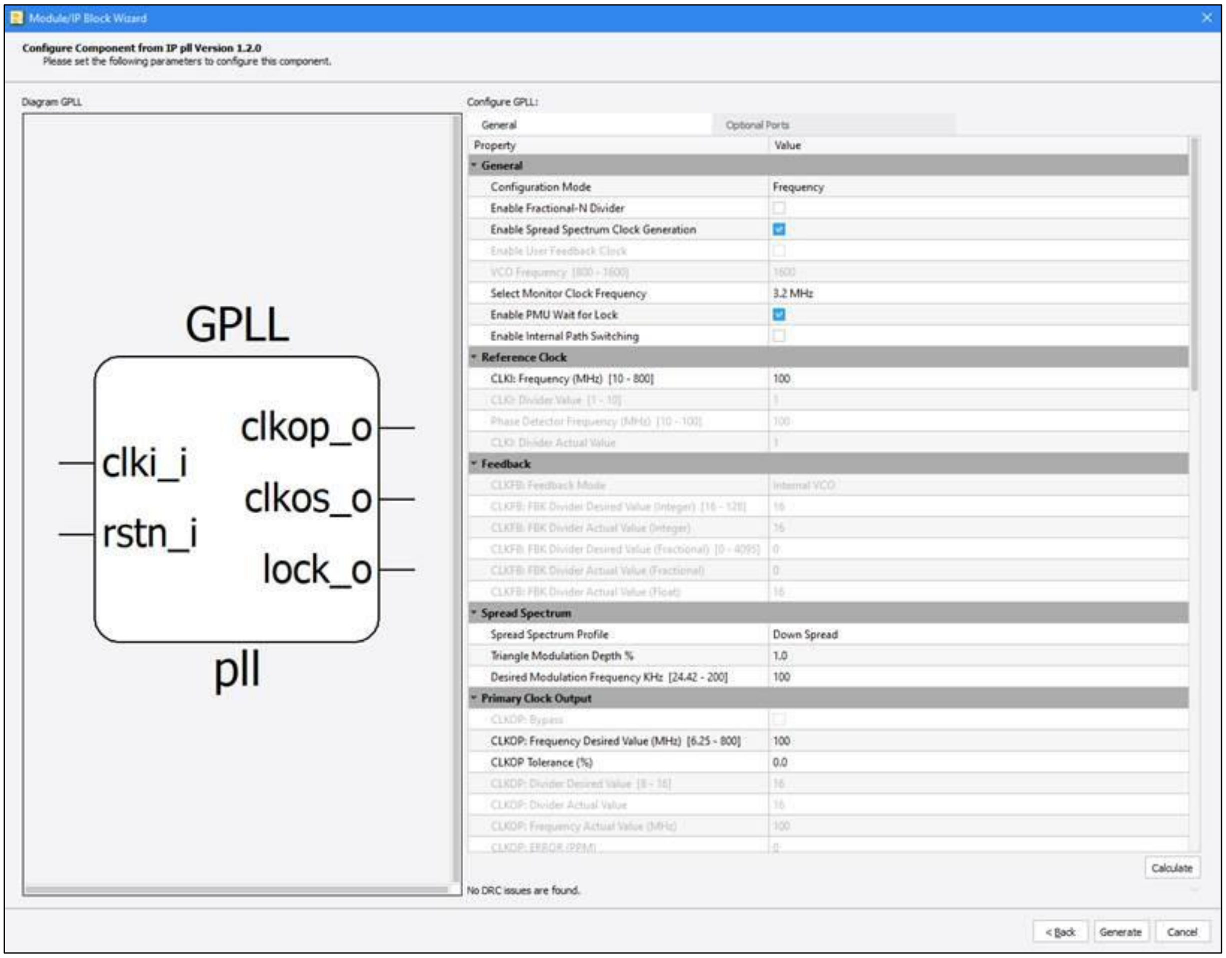

Figure 14.15. Nexus PLL Frequency Configuration in General Tab

Table 14.6. Tab 1, General Settings, IP Catalog User Interface

User Parameters	Range	Default	Description
General			
Configuration Mode	Frequency, Divider	Frequency	Select the configuration mode. Frequency – set the desired input and output frequencies. Divider – set the desired input frequency and desired divider settings.
Enable Fractional-N Divider	Checked, Unchecked	Unchecked	Enable/Disable the Fractional Feedback Clock Divider.
Enable Spread Spectrum Clock Generation	Checked, Unchecked	Unchecked	Enable/Disable the Spread Spectrum Clock Generation.
Enable User Feedback Clock	Checked, Unchecked	Unchecked	When enabled, feedback clock is from user input.
VCO Frequency	Calculated	N/A	Display only.
Select Monitor Clock Frequency	3.2 MHz, 1.0 MHz	3.2 MHz	Select the frequency for reference clock monitoring logic.
Enable Internal Path Switching	Checked, Unchecked	Unchecked	Enable/Disable the internal path switching during POR/Sleep/Standby modes.
Reference Clock			
CLKI: Frequency (MHz)	18–800	100	Set the Reference Clock frequency. (Applicable for Frequency mode only)
CLKI: Divider Value	1–128	1	Set the Reference Clock divider. (Applicable for Divider mode only)
Phase Detector Frequency (MHz)	Calculated	N/A	Display only.
CLKI: Divider Actual Value	Calculated	N/A	Display only.
Feedback			
CLKFB: Feedback Mode	CLKOP, CLKOS, CLKOS2, CLKOS3, CLKOS4, CLKOS5, INTCLKOP, INTCLKOS, INT_CLKOS2, INTCLKOS3, INTCLKOS4, INTCLKOS5	CLKOP	Select the feedback clock from the enabled PLL clock outputs (internal or external).
CLKFB: FBK Divider Desired Value (Integer)	1–128	1	Set the Feedback Clock divider. (Applicable for Divider mode only)
CLKFB: FBK Divider Actual Value (Integer)	Calculated	N/A	Display only.
CLKFB: FBK Divider Desired Value (Fractional)	0 to 4095	0	Set the Feedback Clock fractional divider. (Applicable if Fractional-N Divider is enabled)
CLKFB: FBK Divider Actual Value (Fractional)	Calculated	N/A	Display only.
CLKFB: FBK Divider Actual Value (Float)	Calculated	N/A	Display only (Integer + Fractional).
Spread Spectrum			
Spread Spectrum Profile	Down Spread, Center Spread	Down Spread	Select the Spread Spectrum Profile. (Applicable if Spread Spectrum Clock Generation is enabled)

User Parameters	Range	Default	Description
Triangle Modulation Depth %	0.25, 0.5, 0.75,…,2.0	1.0	Select the modulation depth. (Applicable if Spread Spectrum Clock Generation is enabled)
Desired Modulation Frequency kHz	24.42 kHz–200 kHz	100	Set the desired modulation frequency. (Applicable if Spread Spectrum Clock Generation is enabled)
Clock Output			
CLKO*: Enable	Checked, Unchecked	Unchecked	Enable/Disable PLL Clock Output
CLKO*: Bypass	Checked, Unchecked	Unchecked	Bypass the actual divider output and output the reference clock instead.
CLKO*: Frequency Desired Value (MHz)	6.25–800 MHz	100	Set the Output Clock frequency. (Applicable for Frequency mode only)
CLKO*: Tolerance (%)	0, 0.1, 0.2, 0.5, 1, 2, 5, 10	0.0	Set the acceptable tolerance for actual vs desired output frequency.
CLKO*: Divider Desired Value	1–128	8	Set the Output Clock frequency. (Applicable for Frequency mode only)
CLKO*: Divider Actual Value	Calculated	N/A	Display Only.
CLKO*: Frequency Actual Value (MHz)	Calculated	N/A	Display Only.
CLKO*: Static Phase Shift (Degrees)	0, 45, 90, 135, 180, 225, 270, 315	0	Set the desired clock output phase.
CLKO*: ERROR (PPM)	Calculated	0	Display Only. Difference between desired and actual frequencies.
CLKO*: Enable Trim for CLKO*	Checked, Unchecked	Unchecked	Enable/Disable Trim for clock output.
CLKO*: Duty Trim Options Mode	Rising, Falling	Falling	Select Trim mode.
CLKO*: Duty Trim Options Delay Multiplier	0, 1, 2, 4	0	Select Trim Delay Multiplier.

Optional Ports Tab

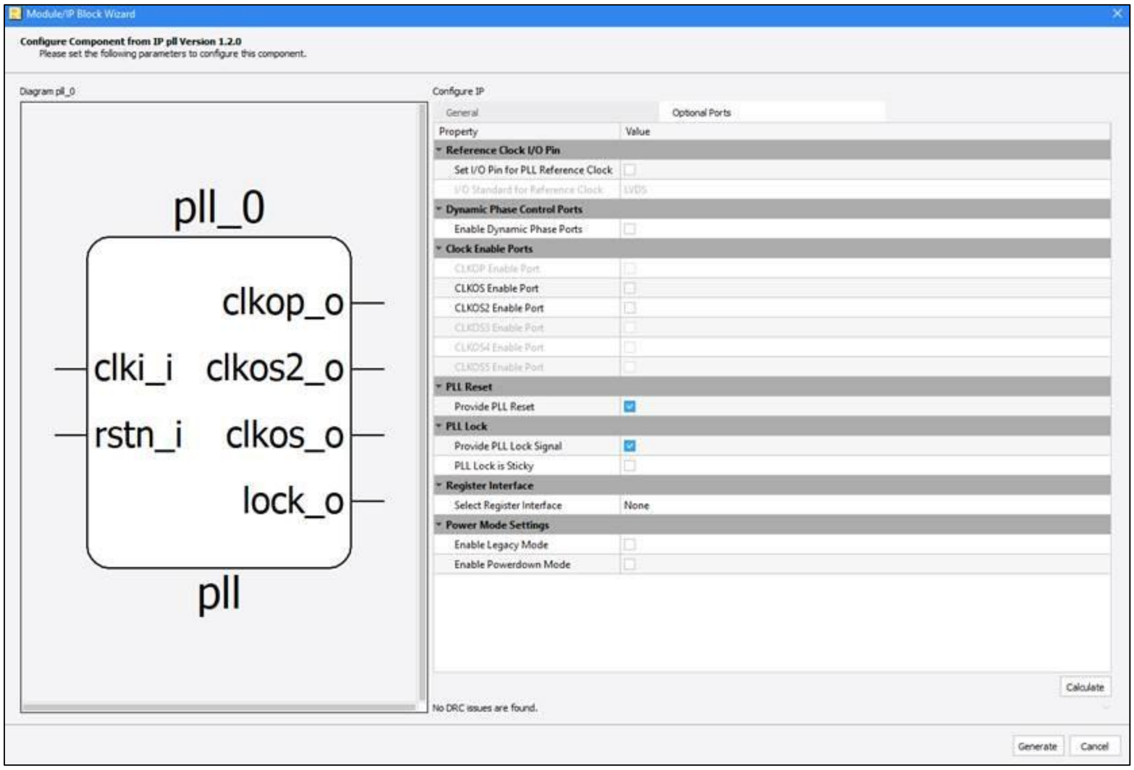

Figure 14.16. Nexus PLL Optional Ports Configuration Tab

Table 14.7. Tab 2, PLL Optional Ports, IP Catalog User Interface

User Parameters	Range	Default	Description
Reference Clock I/O Pin			
Set I/O Pin for PLL Reference Clock	Checked, Unchecked	Unchecked	Enable/Disable I/O Pin option for reference clock.
I/O Standard for Reference Clock	LVDS, SUBLVDS, SLVS, HSTL15_I, HSTL15D_I, LVTTL33, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS18H, LVCMOS15, LVCMOS15H, LVCMOS12, LVCMOS12H, LVCMOS10, LVCMOS10H, LVCMOS10R	LVDS	Select type of I/O pin.
Dynamic Phase Control Ports			
Enable Dynamic Phase Ports	Checked, Unchecked	Unchecked	Enable/Disable dynamic phase control ports.

User Parameters	Range	Default	Description
Clock Enable Ports			
CLKOP/CLKOS[n] Enable Port	Checked, Unchecked	Unchecked	Set to provide clock enable port.
PLL Reset			
Provide PLL Reset	Checked, Unchecked	Checked	Set to provide PLL reset port.
PLL Lock			
Provide PLL Lock Signal	Checked, Unchecked	Checked	Set to provide PLL lock port.
PLL Lock is Sticky	Checked, Unchecked	Unchecked	Set the behaviour of PLL lock signal.
Register Interface			
Select Register Interface	None, APB, LMMI	None	Select type of register interface.
Power Mode Settings			
Enable Legacy Mode	Checked, Unchecked	Unchecked	Set to provide legacy port.
Enable Powerdown Mode	Checked, Unchecked	Unchecked	Set to provide power down port.

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, the FREQUENCY preferences are 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

The following figures show the inputs into the Primary Clock Network through the MIDMUX into the centermux for each device. There are DCC components at the input of the centermux to allow user to stop the clock to save power.

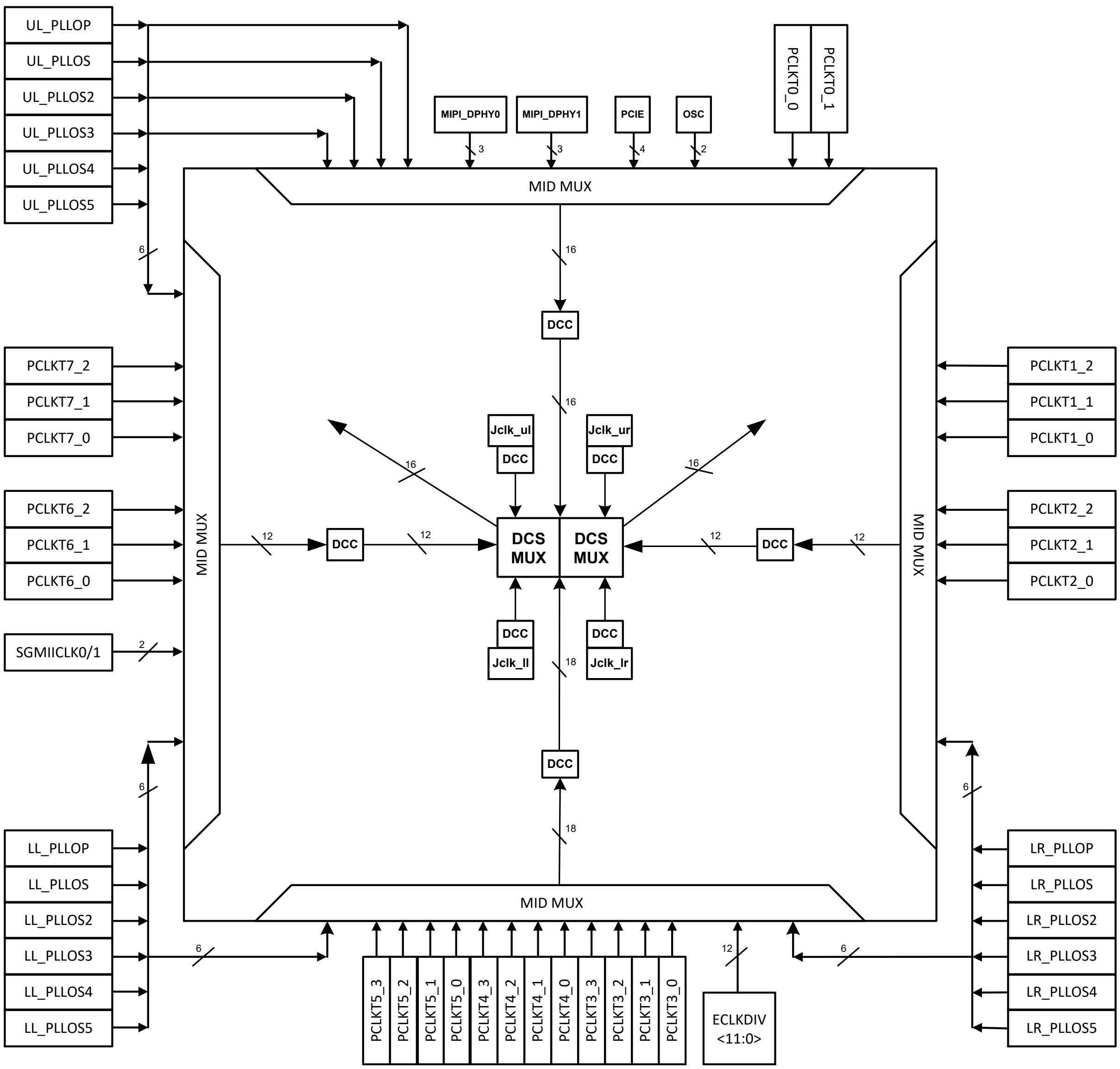

Figure A.1. Nexus Primary Clock Sources and Distribution, LIFCL-40, LFD2NX-40, and LFD2NX-28 Devices

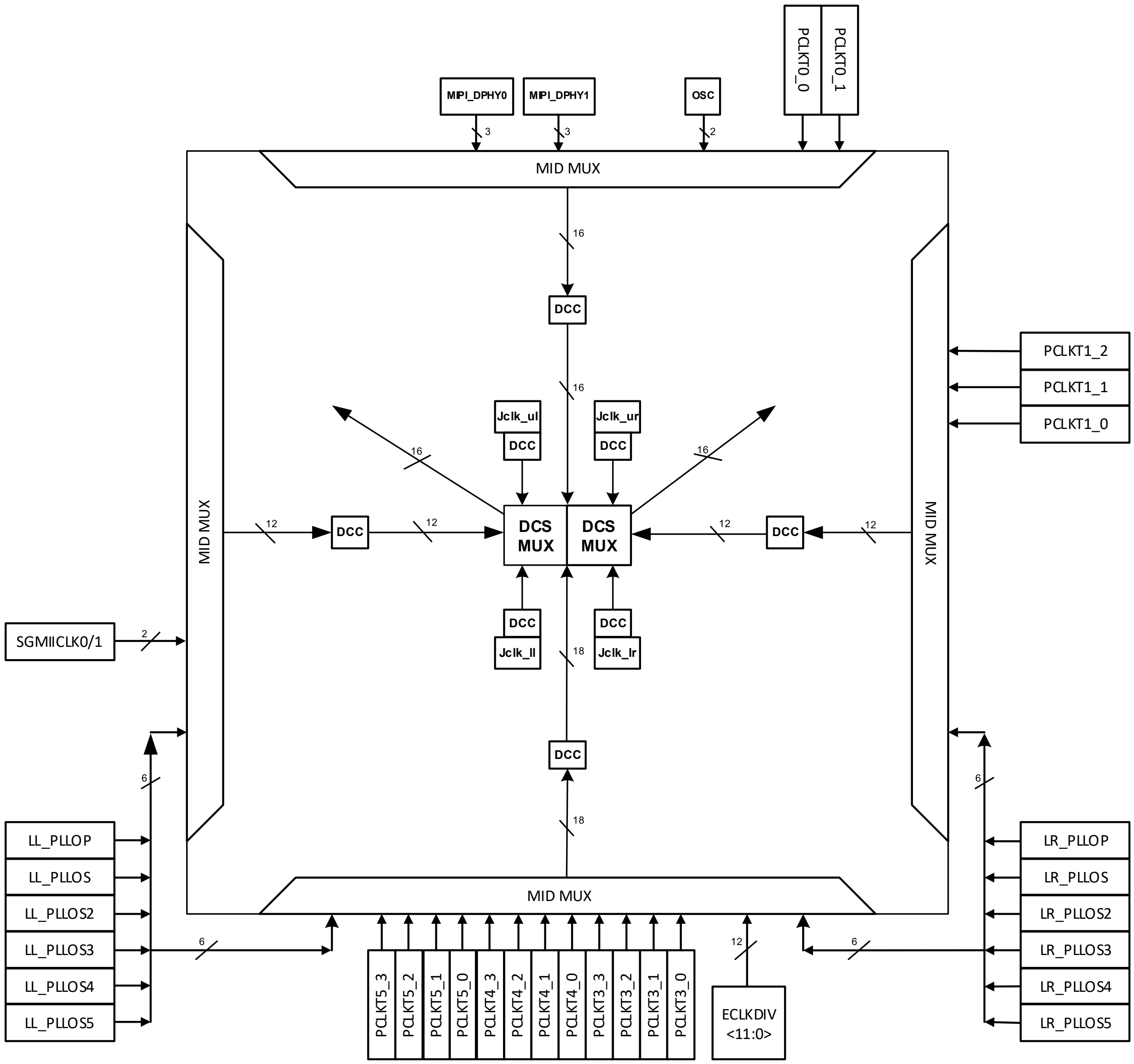

Figure A.2. Nexus Primary Clock Sources and Distribution, LIFCL-17, LFD2NX-17, and LFDD2NX-9 Devices

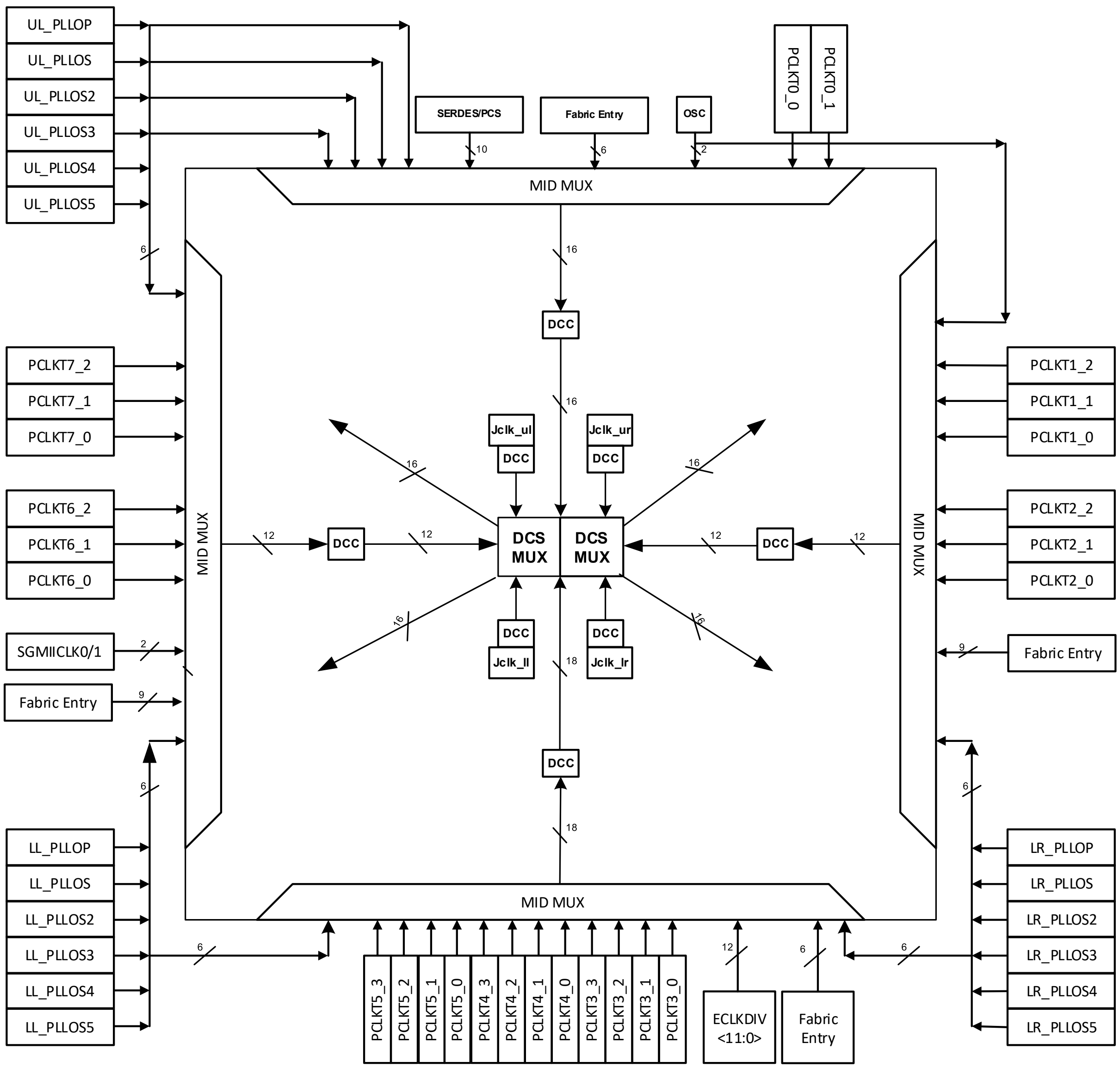

Figure A.3. Nexus Primary Clock Sources and Distribution, LFCPNX-50 Devices

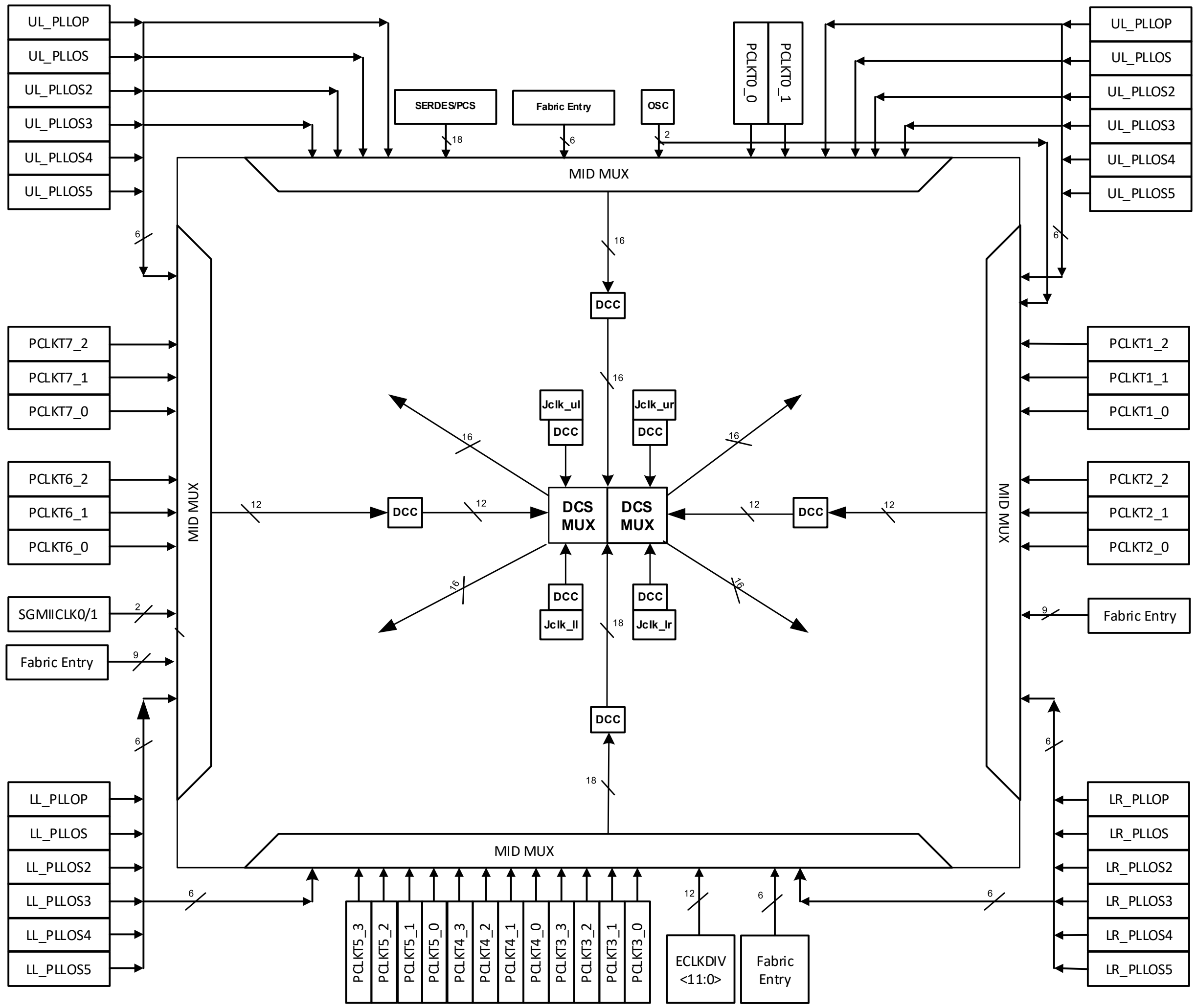

Figure A.4. Nexus Primary Clock Sources and Distribution, LFCPNX-100 and LFMXO5-55TDQ Devices

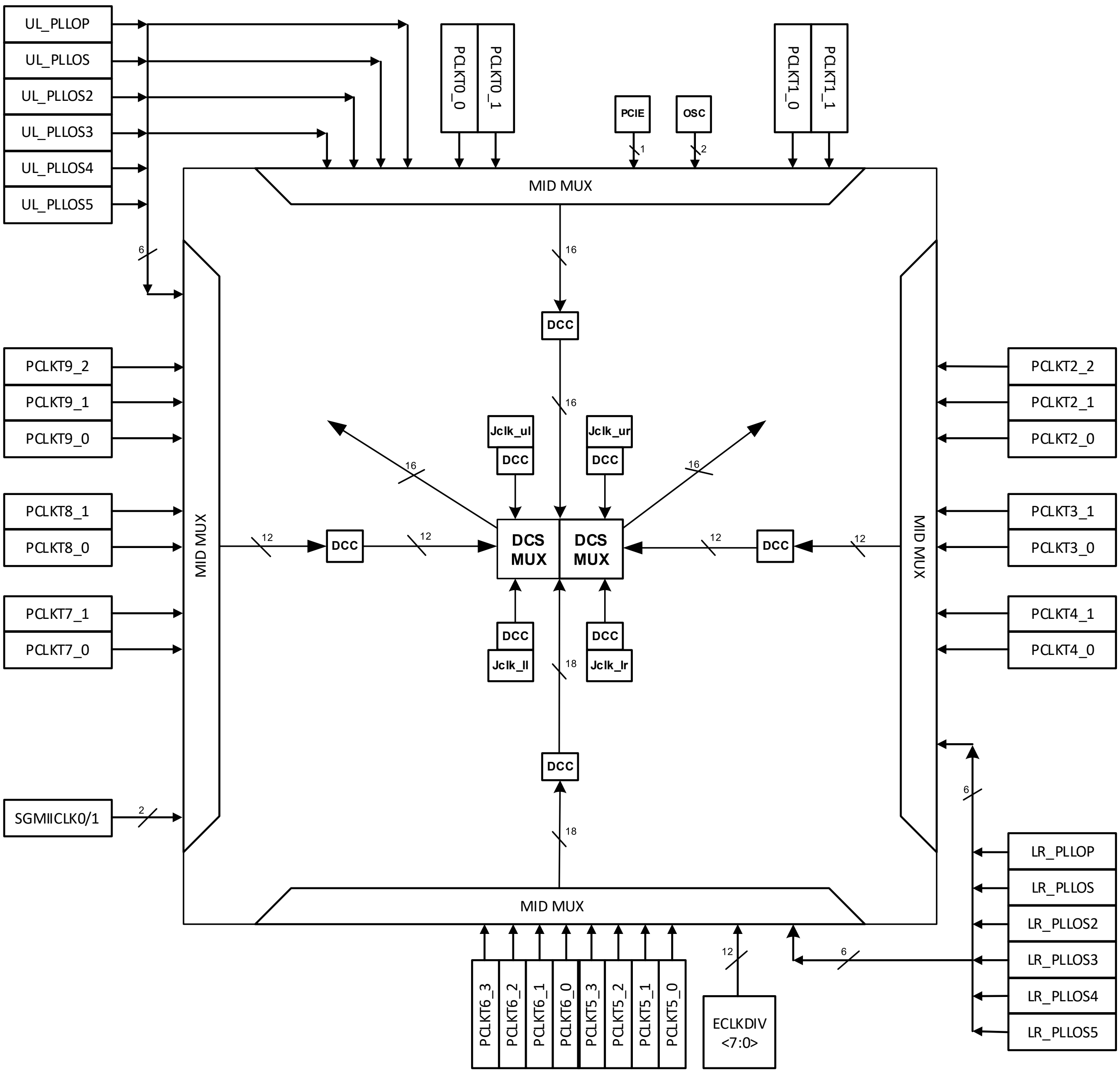

Figure A.5. Nexus Primary Clock Sources and Distribution, LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T and LFD2NX-15/25/35/65 Devices

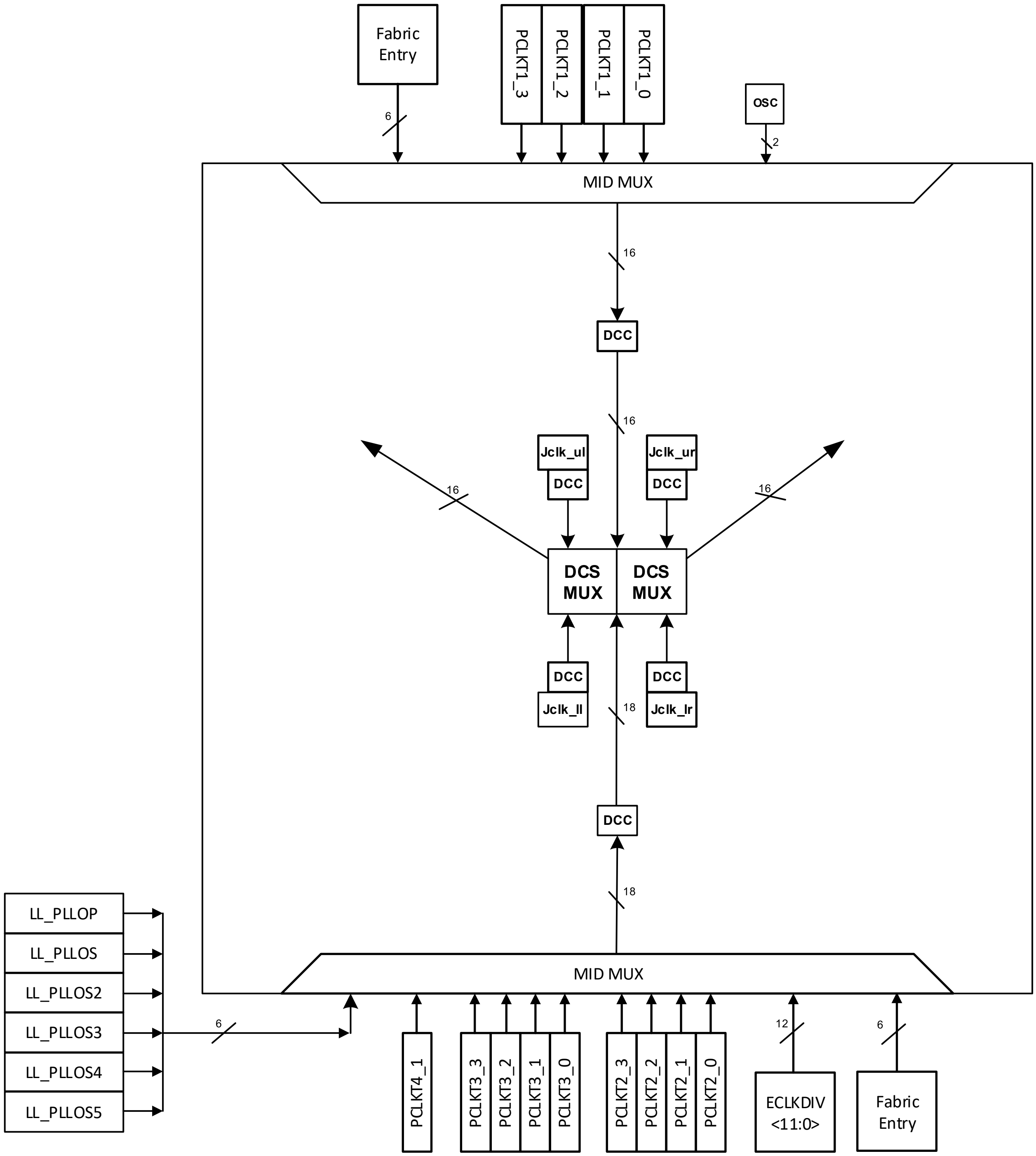

Figure A.6. Nexus Primary Clock Sources and Distribution, LIFCL-33 and LIFCL-33U Devices

Appendix B. Pinout Rules for Clocking in Nexus Devices

In the Nexus 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, user can see the PCLK and PLL input pins listed as below:

Primary Clock Input Pin — PCLK<T/C><Bank>_<0/1/2/3>

Dedicated PLL Input Pin — <LOC>_GPLL0<T/C>_IN

Table B.1. Clock Input Selection Table

Clock Input	Pin to Use	Clock Routing Resource
Clock Input to Logic Directly	PCLK Input Pin	Uses Primary Clock Routing for the Clock.
Clock Input to PLL Only	PLL Input Pin	Uses a Dedicated PLL Input. No Primary Clock Routing is used.
Clock Input to Logic and PLL	PCLK Input Pin	Uses Primary Clock Routing for the Clock.
Clock input to more than 2 PLLs	PCLK Input Pin	Uses Primary Clock Routing for the Clock.

Appendix C. PLL LMMI Operation

The Nexus PLL operating parameters can be changed dynamically through the LMMI bus or APB bus. This section uses LMMI nomenclature. All addresses and bit definitions in PLL Architecture and LMMI Register Map sections are used identically for APB interface applications. A hard-wired LMMI Bus is used to communicate between the LMMI host and the PLL. See Lattice Memory Mapped Interface (LMMI) and Lattice Interrupt Interface (LINTR) User Guide (FPGA-UG-02039) for more information about the LMMI bus.

The LMMI Bus on the PLL module provides support for functional operation and simulation. The user must connect the LMMI Bus to the LMMI host in their HDL design to make the operand and simulation working properly. The LMMI Bus ports and the corresponding LMMI connections are listed in Table C.1.

Table C.1. PLL Data Bus Port Definition

PLL Port Name	I/O	Description
lmmi_clk_i	I	LMMI clock.
lmmi_resetn_i	I	LMMI reset signal (Active Low). Only reset the bus, not register value.
lmmi_offset_i[6:0]	I	LMMI offset address.
lmmi_wr_rdn_i	I	LMMI WR/RD signal (write-high/read-low).
lmmi_request_i	I	LMMI request signal.
lmmi_wdata_i[7:0]	I	LMMI write data.
lmmi_ready_o	O	LMMI ready signal.
lmmi_rdata_o[7:0]	O	LMMI read data.
lmmi_rdata_valid	O	LMMI read data valid signal.

PLL Architecture

The Nexus PLL has six output sections with flexible configuration settings to support a variety of different applications. IP Catalog is able to support most of the common PLL configurations, but for those users with more complex needs the LMMI bus can be used to change the PLL configuration, which allows for more advanced support options.

Each of the six 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, D for the CLKOS3 output, E for the CLKOS4 output, and F for the CLKOS5 output section.
All LMMI addressable PLL Registers defined in Table C.2 have corresponding shadow registers. The output of the shadow register bits controls the PLL hard IP. To alter the PLL setting through the LMMI Registers, user should write the desired new setting to LMMI Registers when the *Shadow_Reg_Update* bit is at 0. After finishing writing all desired LMMI Registers, write 1 to the *Shadow_Reg_Update* bit, so the new setting takes effect at same time.

LMMI Register Map

The LMMI register map for the PLL registers is shown in Table C.2. The items shaded in grey in Table C.2 and Table C.3 are read only.

Table C.2. LMMI Offset Address Locations for PLL Registers

<table>
<tr><th>Addr</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>
<tr><td>00</td><td colspan="7">lmmi_reserved0[6:0]</td><td>Shadow_Reg_Update</td></tr>
<tr><td>01</td><td>lmmi_rotate</td><td colspan="3">lmmi_dyn_sel [2:0]</td><td>lmmi-sleep</td><td>lmmi_stdby</td><td>lmmi_pllreset_ena</td><td>lmmi_pllpd_n</td></tr>
<tr><td>02</td><td colspan="5">lmmi_enable_clk [4:0]</td><td>lmmi_dyn_source</td><td>lmmi_load_reg</td><td>lmmi_direction</td></tr>
<tr><td>03</td><td colspan="4">lmmi_fbk_cur_ble [3:0]</td><td>lmmi_legacy</td><td>lmmi_refin_reset</td><td>lmmi_enable_sync</td><td>lmmi_enable_clk[5]</td></tr>
<tr><td>04</td><td>lmmi_fbk_mask</td><td colspan="2">lmmi_fbk_if_timing_ctl[1:0]</td><td>lmmi_fbk_</td><td colspan="4">lmmi_fbk_cur_ble[7:4]</td></tr>
</table>

<table>
<tr><th>Addr</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>
<tr><td></td><td>[0]</td><td colspan="2"></td><td>edge_sel</td><td colspan="4"></td></tr>
<tr><td>05</td><td>lmmi_fbk_mmd_dig[0]</td><td colspan="7">lmmi_fbk_mask[7:1]</td></tr>
<tr><td>06</td><td>lmmi_fbk_mmd_plus_ctl[0]</td><td colspan="7">lmmi_fbk_mmd_dig[7:1]</td></tr>
<tr><td>07</td><td colspan="2">lmmi_fbk_pi_rc[1:0]</td><td>lmmi_fbk_pi_bypass</td><td colspan="2">lmmi_fbk_mode[1:0</td><td colspan="3">lmmi_fbk_mmd_plus_ctl[3:1]</td></tr>
<tr><td>08</td><td colspan="2">lmmi_fbk_pr_ic[1:0]</td><td colspan="4">lmmi_fbk_pr_cc[3:0]</td><td colspan="2">lmmi_fbk_pi_rc[3:2]</td></tr>
<tr><td>09</td><td colspan="6">lmmi_fbk_rsv[5:0]</td><td colspan="2">lmmi_fbk_pr_ic[3:2]</td></tr>
<tr><td>0A</td><td colspan="8">lmmi_fbk_rsv[13:6]</td></tr>
<tr><td>0B</td><td colspan="4">lmmi_ref_mask[3:0]</td><td>lmmi_ref_integer_mode</td><td>lmmi_fbk_integer_mode</td><td colspan="2">lmmi_fbk_rsv[15:14]</td></tr>
<tr><td>0C</td><td colspan="4">lmmi_ref_mmd_dig[3:0]</td><td colspan="4">lmmi_ref_mask[7:4]</td></tr>
<tr><td>0D</td><td colspan="4">lmmi_ref_mmd_in[3:0]</td><td colspan="4">lmmi_ref_mmd_dig[7:4]</td></tr>
<tr><td>0E</td><td colspan="4">lmmi_ref_mmd_plus_ctl[3:0]</td><td colspan="4">lmmi_ref_mmd_in[7:4]</td></tr>
<tr><td>0F</td><td>lmmi_ldt_int_lock_sticky</td><td colspan="3">lmmi_ldt_lock_sel[2:0]</td><td colspan="2">lmmi_ldt_lock[1:0]</td><td colspan="2">lmmi_ref_timing_ctl[1:0]</td></tr>
<tr><td>10</td><td colspan="4">lmmi_ssc_delta[3:0]</td><td>lmmi_flock_src_sel</td><td>lmmi_flock_en</td><td colspan="2">lmmi_flock_ctrl[1:0]</td></tr>
<tr><td>11</td><td colspan="8">lmmi_ssc_delta[11:4]</td></tr>
<tr><td>12</td><td>lmmi_ssc_en_sdm</td><td>lmmi_ssc_en_center_in</td><td>lmmi_ssc_dither</td><td colspan="2">lmmi_ssc_delta_ctl[1:0]</td><td colspan="3">lmmi_ssc_delta[14:12]</td></tr>
<tr><td>13</td><td colspan="7">lmmi_ssc_f_code[6:0]</td><td>lmmi_ssc_en_ssc</td></tr>
<tr><td>14</td><td colspan="8">lmmi_ssc_f_code[14:7]</td></tr>
<tr><td>15</td><td colspan="8">lmmi_ssc_n_code[7:0]</td></tr>
<tr><td>16</td><td>lmmi_ssc_step_in[0]</td><td>lmmi_ssc_square_mode</td><td colspan="3">lmmi_ssc_reg_weighting_sel[2:0]</td><td>lmmi_ssc_pi_bypass</td><td>lmmi_ssc_order</td><td>lmmi_ssc_n_code[8]</td></tr>
<tr><td>17</td><td colspan="2">lmmi_ssc_tbase[1:0]</td><td colspan="6">lmmi_ssc_step_in[6:1]</td></tr>
<tr><td>18</td><td colspan="8">lmmi_ssc_tbase[9:2]</td></tr>
<tr><td>19</td><td>lmmi_bw_ctl_bias[0]</td><td>lmmi_delay_ctrl</td><td>lmmi_float_cp</td><td colspan="3">lmmi_i_ctrl[2:0]</td><td colspan="2">lmmi_ssc_tbase[11:10]</td></tr>
<tr><td>1A</td><td colspan="3">lmmi_ipi_cmpn[2:0]</td><td>Not Used</td><td>Not Used</td><td colspan="3">lmmi_bw_ctl_bias[3:1]</td></tr>
<tr><td>1B</td><td colspan="3">lmmi_ipp_sel[2:0]</td><td colspan="4">lmmi_ipp_ctrl[3:0]</td><td>lmmi_ipi_cmpn[3]</td></tr>
<tr><td>1C</td><td>lmmi_v2i_pp_ictrl[0]</td><td>lmmi_v2i_1v_en</td><td colspan="4">lmmi_v2i_kvco_sel[3:0]</td><td>lmmi_fast_lock_en</td><td>lmmi_ipp_sel[3]</td></tr>
<tr><td>1D</td><td>lmmi_openloop_en</td><td colspan="3">lmmi_v2i_pp_res[2:0]</td><td colspan="4">lmmi_v2i_pp_ictrl[4:1]</td></tr>
<tr><td>1E</td><td colspan="3">lmmi_cripple[2:0]</td><td colspan="4">lmmi_cset[3:0]</td><td>lmmi_reset_lf</td></tr>
<tr><td>1F</td><td colspan="3">lmmi_mfg_ctrl[2:0]</td><td colspan="5">lmmi_kp_vco[4:0]</td></tr>
<tr><td>20</td><td>lmmi_mfg_en</td><td>lmmi_force_filter</td><td>lmmi_ipi_comp_en</td><td colspan="4">lmmi_ipi_cmp[3:0]</td><td>lmmi_mfg_ctrl[3]</td></tr>
<tr><td>21</td><td colspan="2">lmmi_phib[1:0]</td><td colspan="3">lmmi_phia[2:0]</td><td colspan="3">lmmi_mfg_sel[2:0]</td></tr>
<tr><td>22</td><td>lmmi_phie[0]</td><td colspan="3">lmmi_phid[2:0]</td><td colspan="3">lmmi_phic[2:0]</td><td>lmmi_phib[2]</td></tr>
<tr><td>23</td><td>lmmi_sel_outc</td><td>lmmi_sel_outb</td><td>lmmi_sel_outa</td><td colspan="3">lmmi_phif[2:0]</td><td colspan="2">lmmi_phie[2:1]</td></tr>
<tr><td>24</td><td colspan="5">lmmi_dela[4:0]</td><td>lmmi_sel_outf</td><td>lmmi_sel_oute</td><td>lmmi_sel_outd</td></tr>
<tr><td>25</td><td colspan="6">lmmi_delb[5:0]</td><td colspan="2">lmmi_dela[6:5]</td></tr>
<tr><td>26</td><td colspan="7">lmmi_delc[6:0]</td><td>lmmi_delb[6]</td></tr>
</table>

<table>
<tr><th>Addr</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>
<tr><td>27</td><td>lmmi_dele[0]</td><td colspan="7">lmmi_deld[6:0]</td></tr>
<tr><td>28</td><td colspan="2">lmmi_delf[1:0]</td><td colspan="6">lmmi_dele[6:1]</td></tr>
<tr><td>29</td><td colspan="3">lmmi_diva[2:0]</td><td colspan="5">lmmi_delf[6:2]</td></tr>
<tr><td>2A</td><td colspan="4">lmmi_divb[3:0]</td><td colspan="4">lmmi_diva[6:3]</td></tr>
<tr><td>2B</td><td colspan="5">lmmi_divc[4:0]</td><td colspan="3">lmmi_divb[6:4]</td></tr>
<tr><td>2C</td><td colspan="6">lmmi_divd[5:0]</td><td colspan="2">lmmi_divc[6:5]</td></tr>
<tr><td>2D</td><td colspan="7">lmmi_dive[6:0]</td><td>lmmi_divd[6]</td></tr>
<tr><td>2E</td><td>lmmi_mfgout1_ sel[0]</td><td colspan="7">lmmi_divf[6:0]</td></tr>
<tr><td>2F</td><td colspan="3">lmmi_clkop_trim[2:0]</td><td colspan="3">lmmi_mfgout2_sel[2:0]</td><td colspan="2">lmmi_mfgout1_sel[2:1]</td></tr>
<tr><td>30</td><td colspan="3">lmmi_clkos2_trim[2:0]</td><td colspan="4">lmmi_clkos_trim[3:0]</td><td>lmmi_clkop_ trim[3]</td></tr>
<tr><td>31</td><td colspan="3">lmmi_clkos4_trim[2:0]</td><td colspan="4">lmmi_clkos3_trim[3:0]</td><td>lmmi_clkos2_ trim[3]</td></tr>
<tr><td>32</td><td>lmmi_trimos2_ bypass_n</td><td>lmmi_trimos_ bypass_n</td><td>lmmi_trimop _bypass_n</td><td colspan="4">lmmi_clkos5_trim[3:0]</td><td>lmmi_clkos4_ trim[3]</td></tr>
<tr><td>33</td><td colspan="4">lmmi_div_del[3:0]</td><td>lmmi_pllpdn_ en</td><td>lmmi_trimos5 _bypass_n</td><td>lmmi_trimos4 _bypass_n</td><td>lmmi_trimos3_ bypass_n</td></tr>
<tr><td>34</td><td colspan="2">lmmi_phase_sel_del_p1[1:0]</td><td colspan="3">lmmi_phase_sel_del[2:0]</td><td colspan="3">lmmi_div_del[6:4]</td></tr>
<tr><td>35</td><td colspan="7">lmmi_reserved[6:0]</td><td>lmmi_phase_sel _del_p1[2]</td></tr>
</table>

Table C.3. PLL Registers Descriptions

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access 1
lmmi_reserved0[6:0]	00[7:1]	6	Reserved	7'b0000000	RO
Shadow_Reg_Update	00[0]	1	1'b0: Allows LMMI register bit value updates. 1'b1: Set to 1 to transfer the updated LMMI register content to the active shadow registers. All changed register bit settings become active simultaneously.	1'b0	R/W
lmmi_pllpd_n	01[0]	1	1'b0 – PLL is not used.	1'b0	R/W
lmmi_pllreset_ena	01[1]	1	Active HIGH; Enable PLLRESET CIB signal	1'b0	R/W
lmmi_stdby	01[2]	1	NOT Supported. Must be set to 1'b0.	1'b0	RO
lmmi_sleep	01[3]	1	1'b1 – enable PLL to support sleep/stop mode	1'b0	R/W
lmmi_dyn_sel[2:0]	01[6:4]	3	Output clock phase shift selection, only one output is shifted at one time. 000 – CLKOS 001 – CLKOS2 010 – CLKOS3 011 – CLKOS4 100 – ClKOS5 101 – CLKOP	3'b000	R/W
lmmi_rotate	01[7]	1	LMMI equivalent of CIB rotate signal. Valid if lmmi_dyn_source = 0. Initiate a change from current VCO clock phase to an earlier or later phase on the negative edge of lmmi_rotate.	1'b0	R/W

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access 1
lmmi_direction	02[0]	1	LMMI equivalent of CIB direction signal. Valid if lmmi_dyn_source = 0. Specify direction that lmmi_rotate changes VCO phase. 0 – Phase rotates to later phase. 1 – Phase rotates to earlier phase.	1’b0	R/W
lmmi_load_reg	02[1]	1	LMMI equivalent of CIB load signal. Valid if lmmi_dyn_source = 0. Initiate a divider output phase shift on negative edge of LMMI_LOAD_REG. After two consecutive output clock cycles, one divider cycle use lmmi_del[A/B/C/D/E/F] instead of lmmi_div[A/B/C/D/E/F]. Only the output which is addressed by LMMI_DYN_SEL is shifted. On the falling edge of the LMMI_LOAD_REG, the internal phase shift starts. The minimum pulse width is 10 ns. The setting control signals need to have 5 ns setup time with respect to the LMMI_LOAD_REG falling edge.	1’b0	R/W
lmmi_dyn_source	02[2]	1	1'b0 select LMMI signals for dynamic phase shift. 1'b1 select CIB signals for dynamic phase shift.	1’b0	R/W
lmmi_enable_clk[5:0]	02[7:3] 03[0]	6	1'b1 enables corresponding output mapped as <clkos5, clkos4, clkos3, clkos2, clkos, clkop>.	6’b000000	R/W
lmmi_enable_sync	03[1]	1	Active HIGH; Enable synchronous disable/enable of secondary clocks CLKOS, CLKOS2, CLKOS3, CLKOS4, CLKOS5 with respect to CLKOP.	1’b0	R/W
lmmi_refin_reset	03[2]	1	Active High. Enable PLL internal reset generated after reference mux dynamic selection from CIB.	1’b0	R/W
lmmi_legacy	03[3]	1	Active HIGH; enable Legacy mode.	1’b0	R/W
lmmi_fbk_cur_ble[7:0]	03[7:4] 04[3:0]	8	Bleeding current for PI to adjust the linearity.	8’H00	RO
lmmi_fbk_edge_sel	04[4]	1	Select the positive or negative phase of PI output. 0: positive phase. 1: negative phase.	1’b0	RO
lmmi_fbk_if_timing_ctl[1:0]	04[6:5]	2	Interface timing control for feedback divider.	1’b0	RO
lmmi_fbk_mask[7:0]	04[7] 05[6:0]	8	Minimum divider ratio control word for feedback divider. For example, if n_pll<7:0> or mmd_dig<7:0> is less than lmmi_fbk_mask<7:0>, then the MMD divider ratio is determined by lmmi_fbk_mask<7:0>. Otherwise, the divider ratio is determined by n_pll<7:0> or mmd_dig<7:0>	8’b00001000	RO
lmmi_fbk_mmd_dig[7:0]	05[7] 06[6:0]	8	MMD divider ratio setting in integer mode	8’b00001000	RO
lmmi_fbk_mmd_puls_ctl[3:0]	06[7] 07[2:0]	4	Pulse width control for MMD output clock. If lmmi_fbk_mmd_puls_ctl<3:0>=4'b0110, it means that there's 6 VCO cycles in the MMD output clock. If divider value > 2, 4'b0001.	4’b0000	RO
lmmi_fbk_mode[1:0]	07[4:3]	2	Reserved floating control bits.	2’b00	RO
lmmi_fbk_pi_bypass	07[5]	1	PI bypass control bit. It should be same as lmmi_ssc_pi_bypass. 0: PI not bypass. 1: PI bypass;	1’b0	R/W
lmmi_fbk_pi_rc[3:0]	07[7:6] 08[1:0]	4	RC time constant control in PI.	4'b1100	RO
lmmi_fbk_pr_cc<3:0>	08[5:2]	4	Current control for PI to adjust the linearity.	4’b0000	RO

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access 1
lmmi_fbk_pr_ic<3:0>	08[7:6] 09[1:0]	4	Bias current control for PI.	4'b1000	RO
lmmi_fbk_rsv[15:0]	09[7:2] 0A[7:0] 0B[[1:0]	16	Reserved control bit for feedback divider.	16'H0000	RO
lmmi_fbk_integer_mode	0B[2]	1	Enable the integer mode for feedback divider.	1'b0	R/W
lmmi_ref_integer_mode	0B[3]	1	Integer mode control bit for reference clock pre-divider	1'b0	R/W
lmmi_ref_mask[7:0]	0B[7:4] 0C[3:0]	8	Minimum divider ratio control word for reference pre-divider.	8'H00	R/W
lmmi_ref_mmd_dig[7:0]	0C[7:4] 0D[3:0]	8	MMD divider ratio setting for reference pre-divider when lmmi_ref_integer_mode=1	8'H08	R/W
lmmi_ref_mmd_in[7:0]	0D[7:4] 0E[3:0]	8	MMD divider ratio setting for reference pre-divider when lmmi_ref_integer_mode=0.	8'H08	R/W
lmmi_ref_mmd_puls_ctl[3:0]	0E[7:4]	4	Pulse width control for MMD output clock in reference pre-divider. If divider value > 2, 4'b0001.	4'b0000	RO
lmmi_ref_timing_ctl[1:0]	0F[1:0]	2	Interface timing control for reference divider. Default setting is 2'b00.	2'b00	RO
lmmi_ldt_lock[1:0]	0F[3:2]	2	Frequency lock-detector resolution sensitivity 00 = takes about 98304 PFDFBK cycles to lock 01 = takes about 24576 PFDFBK cycles to lock 10 = takes about 6144 PFDFBK cycles to lock 11 = takes about 1536 PFDFBK cycles to lock	2'b11	R/W
lmmi_ldt_lock_sel[2:0]	0F[6:4]	3	Lock-detector type select: 000 = UNSTICKY freq lock[1] 001 = STICKY phase lock (freq lock first detected then phase lock 1st detected) * 010 = STICKY frequency lock (freq lock first detected) [1] 011 = UNSTICKY freq and STICKY phase lock[1] 100 = UNSTICKY freq lock 101 = UNSTICKY phase lock 110 = STICKY freq lock 111 = UNSTICKY freq and STICKY phase lock	3'b000	R/W
lmmi_ldt_int_lock_sticky	0F[7]	1	Active HIGH to have INT_LOCK STICKY. Default to be 1'b1. 1'b0 for PDE/DE purpose.	1'b1	Read Only
lmmi_flock_ctrl[1:0]	10[1:0]	2	2 bits control the fast lock period. 00 is 1x, 01 is 2x,10 is 4x,11 is 8x	2'b01	R/W
lmmi_flock_en	10[2]	1	Active high. To enable fast lock.	1'b1	R/W
lmmi_flock_src_sel	10[3]	1	fast lock source selection: 0 is ref clock. 1 is feedback clock.	1'b0	R/W
lmmi_ssc_delta[14:0]	10[7:4] 11[7:0] 12[2:0]	15	RSVD	15'H0000	RO
lmmi_ssc_delta_ctl[1:0]	12[4:3]	2	RSVD	2'b00	RO
lmmi_ssc_dither	12[5]	1	Dither enable or disable for SDM. 0: disable; 1: enable	1'b0	RO
lmmi_ssc_en_center_in	12[6]	1	Down triangle or central triangle control bit in SSC profile generator. 0: down-triangle; 1: central-triangle.	1'b0	RO

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access 1
lmmi_ssc_en_sdm	12[7]	1	Enable or disable the SDM. 0: disable the SDM. 1: enable the SDM. In stair-wave SSC mode, it should be set to 0.	1'b0	RO
lmmi_ssc_en_ssc	13[0]	1	Enable or disable the SSC profile generator. 0: disable the SSC generator. 1: enable the SSC generator.	1'b0	RO
lmmi_ssc_f_code[14:0]	13[7:1] 14[7:0]	15	Fractional part of the feedback divider ratio.	15'H0000	RO
lmmi_ssc_n_code[8:0]	15[7:0] 16[0]	9	Integer part of feedback divider ratio	9'b00001 0100	RO
lmmi_ssc_order	16[1]	1	SDM order control bit. 0: SDM order=1; 1: SDM order=2, MASH1-1	1'b0	RO
lmmi_ssc_pi_bypass	16[2]	1	PI bypass control bit. 0: PI not bypass; 1: PI bypass; it should be same as lmmi_fbk_pi_bypass.	1'b0	RO
lmmi_ssc_reg_weighting_sel[2:0]	16[5:3]	3	Weighting control bit for lmmi_ssc_step_in<6:0>	3'b000	RO
lmmi_ssc_square_mode	16[6]	1	Two-point FSK modulation control bit. 0: disable; 1: enable 2-point FSK.	1'b0	RO
lmmi_ssc_step_in[6:0]	16[7] 17[5:0]	7	SSC modulation depth control bit.	7'b0000000	RO
lmmi_ssc_tbase[11:0]	17[7:6] 18[7:0] 19[1:0]	12	SSC modulation frequency control. The frequency should be 30~33kHz.	12'H000	RO
lmmi_I_ctrl[2:0]	19[4:2]	3	current tuning: 000:10 µA; 001:8.3 µA; 010:14.9 µA;011:12.4 µA; 100:19.8 µA;101:17.3 µA; 110:24.8 µA;111:22.3 µA	3'b000	RO
lmmi_float_cp	19[5]	1	Active HIGH to tri-state the ICP output.	1'b0	RO
lmmi_delay_ctrl	19[6]	1	Control signal to adjust the delay of the PFD; default 1b0=200 ps;1b1=300 ps	1'b0	RO
lmmi_bw_ctl_bias[3:0]	19[7] 1A[2:0]	4	Input control signal to tune the bias current of ppath cp, When the bit increase, the bias current increase. This current branch is combined with the Ivco_fb current.	4'b0000	RO
lmmi_ipi_cmpn[3:0]	1A[7:5] 1B[0]	4	Input control bits to compensate i-path bias current	4'b0000	RO
lmmi_ipp_ctrl[3:0]	1B[4:1]	4	Input control signal to tune the bias current of ppath cp. ipp_ctrl<3:2> was used to control the bias voltage of the cpp_bias; ipp_ctrl<1:0> was used to tune the current of bias, start from 5uA to 20uA with 5uA step from each bit.	4'b0110	RO
lmmi_ipp_sel[3:0]	1B[7:5] 1C[0]	4	Input control signal to select which ppath cp is on,there are 4 branchs at max p-path current function: [5uA+ipp_ctrl<1:0>×5uA]/3×bw_ctl_bias<3:0>×[ipp_sel<3>+ipp_sel<2>+ipp_sel<1>+ipp_sel<0>]	4'b1111	RO
lmmi_fast_lock_en	1C[1]	1	Enable signal for fast lock, default to 1'b1	1'b0	RO
lmmi_v2i_kvco_sel[3:0]	1C[5:2]	4	v2i kvco slope control, 10 + Dec(lmmi_v2i_kvco_sel) × 5	4'b1001 (for 0.9V) 4'b0100 (for 1V)	RO

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access 1
lmmi_v2i_1v_en	1C[6]	1	1 V supply enable or disable. 0: disable. 1: enable	1'b0 (for 0.9v) 1'b1 (for 1v)	RO
lmmi_v2i_pp_ictrl<4:0>	1C[7] 1D[3:0]	5	P-path v2i gm control	5'b00110	RO
lmmi_v2i_pp_res<2:0>	1D[6:4]	3	P-path high frequency pole resistor control 000: 11.3K 001: 11K 010:10.7K 011:10.3K 100: 10K 101:9.7K 110: 9.3K 111:9K	3’b000	RO
lmmi_openloop_en	1D[7]	1	open loop mode enable for mfg testing.	1’b0	RO
lmmi_reset_lf	1E[0]	1	lpf reset enable, default to 1’b0	1’b0	RO
lmmi_cset[3:0]	1E[4:1]	4	LPF cap control 0000:8p; 0001:12p 0010:16p 0011:20p 0100:24p 0101:28p 0110:32p 0111:36p 1000:40p 1001:44p 1010:48p 1011:52p 1100:56p 1101:60p 1110:64p 1111:68p	4'b1000	RO
lmmi_cripple[2:0]	1E[7:5]	3	LPF cap control 000:1p; 001:3p 010:5p 011:7p 100:9p 101:11p 110:13p 111:15p	3'b010	RO
lmmi_kp_vco[4:0]	1F[4:0]	5	90 + Dec (lmmi_kp_vco) × 10;	5’b11001 (for 0.9V) 5’b00011 (for 1.0v)	RO
lmmi_mfg_ctrl[3:0]	1F[7:5] 20[0]	4	mfg internal vctrl selection 0000:0; 1111:vdd, step is 55 mV	4’b0000	RO
lmmi_ipi_cmp[3:0]	20[4:1]	4	i-path CP compensate up/dn mismatch at process variation	4'b1000	RO
lmmi_ipi_cmp_en	20[5]	1	Enable ipi_cmp combine with lmmi_en_ipi_cmp to adjust the delay of the PFD,this bit is LSB; 2'b00=160ps; 2'b10=200ps; 2'b10=230ps; 2'b11=300ps	1’b0	RO
lmmi_force_filter	20[6]	1	force internal vctrl=analog pad	1’b0	RO
lmmi_mfg_en	20[7]	1	mfg feature enable pin	1’b0	RO
lmmi_mfg_sel[2:0]	21[2:0]	3	mfg current mux selection 000:I path CP up current; 001:P path CP up current; 010:I path V2I current 011:P path V2I current 100:I path CP dn current 101:P path CP dn current 110:NA 111:NA	3’b000	RO
lmmi_phia [2:0]	21[5:3]	3	Select VCO phase-shift (0..7) for A section	3’b000	R/W
lmmi_phib [2:0]	21[7:6] 22[0]	3	Select VCO phase-shift (0..7) for B section	3’b000	R/W
lmmi_phic [2:0]	22[3:1]	3	Select VCO phase-shift (0..7) for C section	3’b000	R/W
lmmi_phid [2:0]	22[6:4]	3	Select VCO phase-shift (0..7) for D section	3’b000	R/W
lmmi_phie [2:0]	22[7] 23[1:0]	3	Select VCO phase-shift (0..7) for E section	3’b000	R/W
lmmi_phif [2:0]	23[4:2]	3	Select VCO phase-shift (0..7) for F section	3’b000	R/W

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access 1
lmmi_sel_outa	23[5]	1	Select output to CLKOP	1’b0	R/W
lmmi_sel_outb	23[6]	1	Select output to CLKOS	1’b0	R/W
lmmi_sel_outc	23[7]	1	Select output to CLKOS2	1’b0	R/W
lmmi_sel_outd	24[0]	1	Select output to CLKOS3	1’b0	R/W
lmmi_sel_oute	24[1]	1	Select output to CLKOS4	1’b0	R/W
lmmi_sel_outf	24[2]	1	Select output to CLKOS5	1’b0	R/W
lmmi_dela[6:0]	24[7:3] 25[1:0]	7	Delay A section output DELA VCO clock cycles with respect to VCO phase 0, or REFBUF if in VCO bypass mode (post-divider phase-shift)	7’H00	R/W
lmmi_delb[6:0]	25[7:2] 26[0]	7	Delay B section output DELB VCO clock cycles with respect to VCO phase 0, or REFBUF if in VCO bypass mode (post-divider phase-shift)	7’H00	R/W
lmmi_delc[6:0]	26[7:1]	7	Delay C section output DELC VCO clock cycles with respect to VCO phase 0, or REFBUF if in VCO bypass mode (post-divider phase-shift)	7’H00	R/W
lmmi_deld[6:0]	27[6:0]	7	Delay D section output DELD VCO clock cycles with respect to VCO phase 0, or REFBUF if in VCO bypass mode (post-divider phase-shift)	7’H00	R/W
lmmi_dele[6:0]	27[1] 28[5:0]	7	Delay E section output DELE VCO clock cycles with respect to VCO phase 0, or REFBUF if in VCO bypass mode (post-divider phase-shift)	7’H00	R/W
lmmi_delf[6:0]	28[7:6] 29[4:0]	7	Delay F section output DELF VCO clock cycles with respect to VCO phase 0, or REFBUF if in VCO bypass mode (post-divider phase-shift)	7’H00	R/W
lmmi_diva [6:0]	29[7:5] 2A[3:0]	7	Output dividers setting for clkop: divide value = DIVA + 1	7’H00	R/W
lmmi_divb [6:0]	2A[7:4] 2B[2:0]	7	Output dividers setting for clkos: divide value = DIVB + 1	7’H00	R/W
lmmi_divc [6:0]	2B[7:3] 2C[1:0]	7	Output dividers setting for clkos2: divide value = DIVC + 1	7’H00	R/W
lmmi_divd [6:0]	2C[7:2] 2D[0]	7	Output dividers setting for clkos3: divide value = DIVD + 1	7’H00	R/W
lmmi_dive[6:0]	2D[7:1]	7	Output dividers setting for clkos4: divide value = DIVE + 1	7’H00	R/W
lmmi_divf[6:0]	2E[6:0]	7	Output dividers setting for clkos5: divide value = DIVF + 1	7’H00	R/W
lmmi_mfgout1_sel[2:0]	2E[7] 2F[1:0]	3	PLL_MFGOUT1 selection bits	3’b000	RO
lmmi_mfgout2_sel[2:0]	2F[4:2]	3	PLL_MFGOUT2 selection bits	3’b000	RO
lmmi_clkop _trim[3:0]	2F[7:5] 30[0]	4	CLKOP output edge trim	4’b0000	RO
lmmi_clkos _trim[3:0]	30[4:1]	4	CLKOS output edge trim	4’b0000	RO
lmmi_clkos2 _trim[3:0]	30[7:5] 31[0]	4	CLKOS2 output edge trim	4’b0000	RO
lmmi_clkos3 _trim[3:0]	31[4:1]	4	CLKOS3 output edge trim	4’b0000	RO
lmmi_clkos4 _trim[3:0]	31[7:5] 32[0]	4	CLKOS4 output edge trim	4’b0000	RO
lmmi_clkos5_trim[3:0]	32[4:1]	4	CLKOS5 output edge trim	4’b0000	RO
lmmi_trimop _bypass_n	32[5]	1	CLKOP output edge trim bypass	1’b0	RO
lmmi_trimos _bypass_n	32[6]	1	CLKOS output edge trim bypass	1’b0	RO
lmmi_trimos2 _bypass_n	32[7]	1	CLKOS2 output edge trim bypass	1’b0	RO

Register Name	Register Addr (Hex)	Size (Bits)	Description	Default Value	User Access 1
lmmi_trimos3 _bypass_n	33[0]	1	CLKOS3 output edge trim bypass	1'b0	RO
lmmi_trimos4_bypass_n	33[1]	1	CLKOS4 output edge trim bypass	1'b0	RO
lmmi_trimos5_bypass_n	33[2]	1	CLKOS5 output edge trim bypass	1'b0	RO
lmmi_pllpdn_en	33[3]	1	Active high to enable pllpd_n CIB control	1'b0	R/W
lmmi_div_del[6:0]	33[7:4] 34[2:0]	7	The internal delay path divider	7'b0000001	R/W
lmmi_phase_sel_del[2:0]	34[5:3]	3	The internal phase delay selection path	3'b000	R/W
lmmi_phase_sel_del_p1[2:0]	34[7:6] 35[0]	3	The internal phase delay selection path1	3'b000	R/W
lmmi_reserved[6:0]	35[7:1]	7	Reserved	7'b0000000	RO

Notes:

1. Gated with pll_wakeup_sync pin.
2. R/W = Read and Write; RO = Read Only
3. ppath (p-path) stands for proportional path, I-path stands for Integral path which is a dual tuning PLL using PI tuning loop, and CP is abbreviated for Charge Pump.

Appendix D. Edge Clock Input Source Connection

The possible Edge Clock input sources are shown in the table below.

Table D.1. ECLK Input Source Connections

Source	Ports		I/O Bank 3				I/O Bank 4				I/O Bank 5			
			eclk0	eclk1	eclk2	eclk3	eclk0	eclk1	eclk2	eclk3	eclk0	eclk1	eclk2	eclk3
GPLL	llgpll	clkop	✓	✓	—	—	✓	✓	—	—	✓	✓	—	—
		clkos	—	✓	✓	—	—	✓	✓	—	—	✓	✓	—
		clkos2	—	—	✓	✓	—	—	✓	✓	—	—	✓	✓
		clkos3	✓	—	—	✓	✓	—	—	✓	✓	—	—	✓
		clkos4	—	✓	—	✓	—	✓	—	✓	—	✓	—	✓
		clkos5	✓	—	✓	—	✓	—	✓	—	✓	—	✓	—
	lrgpll	clkop	—	—	✓	✓	—	—	✓	✓	—	—	✓	✓
		clkos	✓	—	—	✓	✓	—	—	✓	✓	—	—	✓
		clkos2	✓	✓	—	—	✓	✓	—	—	✓	✓	—	—
		clkos3	—	✓	✓	—	—	✓	✓	—	—	✓	✓	—
		clkos4	✓	—	✓	—	✓	—	✓	—	✓	—	✓	—
		clkos5	—	✓	—	✓	—	✓	—	✓	—	✓	—	✓
LVDS PIO	Bank 3	pclk3_0	✓	✓	✓	✓	—	—	—	—	—	—	—	—
		pclk3_1	✓	✓	✓	✓	—	—	—	—	—	—	—	—
		pclk3_2	✓	✓	✓	✓	—	—	—	—	—	—	—	—
		pclk3_3	✓	✓	✓	✓	—	—	—	—	—	—	—	—
	Bank 4	pclk4_0	—	—	—	—	✓	✓	✓	✓	—	—	—	—
		pclk4_1	—	—	—	—	✓	✓	✓	✓	—	—	—	—
		pclk4_2	—	—	—	—	✓	✓	✓	✓	—	—	—	—
		pclk4_3	—	—	—	—	✓	✓	✓	✓	—	—	—	—
	Bank 5	pclk5_0	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		pclk5_1	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		pclk5_2	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		pclk5_3	—	—	—	—	—	—	—	—	✓	✓	✓	✓

References

- Lattice Memory Mapped Interface (LMMI) and Lattice Interrupt Interface (LINTR) User Guide (FPGA-UG-02039)
- CrossLink-NX-33 and CrossLink-NX-33U Data Sheet (FPGA-DS-02104)
- CrossLink-NX Family Data Sheet (FPGA-DS-02049)
- Certus-NX Family Data Sheet (FPGA-DS-02078)
- CertusPro-NX Family Data Sheet (FPGA-DS-02086)
- MachXO5-NX Family Datasheet (FPGA-DS-02102)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- CrossLink-NX web page
- Development Kits & Boards for CrossLink-NX web page
- IP & Reference Designs for CrossLink-NX web page
- Certus-NX web page
- Development Kits & Boards for Certus-NX web page
- IP and Reference Designs for Certus-NX web page
- CertusPro-NX web page
- Development Kits & Boards for CertusPro-NX web page
- IP and Reference Designs for CertusPro-NX web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

Revision 2.7, October 2025

Section	Change Summary
All	• Renamed document from *sysCLOCK PLL Design and User Guide for Nexus Platform* to *sysCLOCK PLL Design User Guide for Nexus Platform*. • Performed minor formatting and editorial edits.
Acronyms in This Document	Updated list of acronyms.
Introduction	Added LFMXO5-20TD, LFMXO5-20TDQ, LFMXO5-30TD, LFMXO5-30TDQ, and LFMXO5-55TDQ devices in Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers.
Nexus Top-Level View	Added device support in the following figures: • Figure 3.2. CertusPro-NX (LFCPNX-100) and MachXO5-NX (LFMXO5-55TDQ) Clocking Structure • Figure 3.3. MachXO5-NX (LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T) and Certus-NX (LFD2NX-15/25/35/65) Clocking Structure
Overview of Clocking Components	• Updated the Dynamic Clock Control (DCC) section. • Removed the oscillator output specifications in the Oscillator (OSC) section.
Primary Clocks	Added device support in the following figures: • Figure 6.1. Primary Clock Routing Architecture for CrossLink-NX, Certus-NX, and MachXO5-NX (LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T) • Figure 6.2. Primary Clock Routing Architecture for CertusPro-NX and MachXO5-NX (LFMXO5-55TDQ)
Primary Clock Divider (PCLKDIVSP)	Updated the description for CLKOUT in Table 7.1. PCLKDIVSP Component Port Definition.
Dynamic Clock Select (DCS)	Updated the description for SELFORCE in Table 8.1. DCS Component Port Definition.
Dynamic Clock Control (DCC)	Changed feed line to clock in the Dynamic Clock Control (DCC) section.

Internal Oscillator (OSCA, OSCD)	• Renamed section title from *Internal Oscillator (OSCA)* to Internal Oscillator (OSCA, OSCD), and added support for the OSCD component.
	• Added the following sections:
	  • OSCD Component Definition
	  • OSCD Usage in VHDL
	  • OSCD Usage in Verilog

sysCLOCK PLL	• Added device support in the following figures:
	  • Figure 14.5. PLL Input Pins for LFCPNX-100 and LFMXO5-55TDQ
	  • Figure 14.6. PLL Input Pins for LFD2NX-15/25/35/65 and LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T
	• Added description on the reference clock if the optional reset port is added to the PLL in the CLKI Input section.
	• Updated the RST Input section as follows:
	  • Added description on the reference clock if the optional reset port is added to the PLL.
	  • Removed the t_{LOCK} specification.
	• Removed the t_{LOCK} specification in the LOCK Output section.
	• Updated the VCO Phase Shift section as follows:
	  • Updated the phase shift operation and equation.
	  • Removed the description on the glitchless and non-glitchless dynamic phase adjustment.
	  • Updated Figure 14.10. PLL Phase Shifting Using the PHASESTEP Signal.
	• Updated the spread spectrum range in the Spread Spectrum Clock Generation section.
	• Updated the range for the **I/O Standard for Reference Clock** parameter in Table 14.7. Tab 2, PLL Optional Ports, IP Catalog User Interface.

Section	Change Summary
Appendix A. Primary Clock Sources and Distribution	Added device support in the following figures: • Figure A.4. Nexus Primary Clock Sources and Distribution, LFCPNX-100 and LFMXO5-55TDQ Devices • Figure A.5. Nexus Primary Clock Sources and Distribution, LFMXO5-15/20TD/20TDQ/25/30TD/30TDQ/35/35T/65/65T and LFD2NX-15/25/35/65 Devices

Revision 2.6, July 2025

Section	Change Summary
All	• Added the following note where applicable: *The SGMII interface using LVDS I/O has limitations when operating across the full specified temperature range. Lattice recommends using alternative interfaces, such as SERDES or RGMII, for designs requiring Gigabit Ethernet. Refer to the Knowledge Base article for details. Contact your local Lattice sales representative for more information*. • Changed *SerDes* to *SERDES*.
Acronyms in This Document	Added LVDS, RGMII, and SERDES.
Introduction	Added LFD2NX-15, LFD2NX-25, LFD2NX-35, LFD2NX-65, LFMXO5-15, LFMXO5-25, LFMXO5-35/T, and LFMXO5-65/T devices in Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers.
Nexus Top-Level View	• Updated device support in Figure 3.1. CrossLink-NX and Certus-NX (LFD2NX-9, LFD2NX-17, LFD2NX-28, and LFD2NX-40) Clocking Structure. • Updated Figure 3.3. MachXO5-NX and Certus-NX (LFD2NX-35 and LFD2NX-65) Clocking Structure and added device support.
sysCLOCK PLL	Added reference for details on changing the clock routing preference in the Dedicated PLL Inputs section.
Appendix A. Primary Clock Sources and Distribution	Added Figure A.5. Nexus Primary Clock Sources and Distribution, MachXO5-NX and Certus-NX (LFD2NX-35 and LFD2NX-65) Devices.
Appendix D. Edge Clock Input Source Connection	Added this section.
References	Added reference to the Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059).

Revision 2.5, July 2024

Section	Change Summary
Introduction	Added LFD2NX-9 and LFD2NX-28 devices in Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers.
Nexus Top-Level View	Added support for LFD2NX-9 and LFD2NX-28 devices in the Nexus Top-Level section.
Overview of Clocking Components	Changed from *PCLKDIV* to *PCLKDIVSP* in the Primary Clock Divider (PCLKDIVSP) section.
Primary Clocks	Changed from *PCLKDIV* to *PCLKDIVSP* in the Primary Clock Sources section.

Primary Clock Divider (PCLKDIVSP)	• Renamed section from *Primary Clock Divider (PCLKDIV)* to Primary Clock Divider (PCLKDIVSP).
	• Changed from *PCLKDIV* to *PCLKDIVSP* in the following subsections:
	  • PCLKDIVSP Component Definition
	  • PCLKDIVSP Usage in VHDL
	  • PCLKDIVSP Usage in Verilog
	• Updated the codes in the PCLKDIVSP Usage in VHDL section.

Dynamic Clock Select (DCS)	Changed from *PCLKDIV* to *PCLKDIVSP* in the following diagrams: • Figure 8.1. DCS_CMUX Structure for CrossLink-NX, Certus-NX, and MachXO5-NX • Figure 8.2. DCS_CMUX Structure for CertusPro-NX • Figure 8.3. DCS Logic Structure

Section	Change Summary
sysCLOCK PLL	• Added support for LFD2NX-9 and LFD2NX-28 devices in the sysCLOCK PLL Overview and Dedicated PLL Inputs sections. • Removed description on dedicated external dual-purpose I/O pin in the CLKFB Input section.
Appendix A. Primary Clock Sources and Distribution	• Added support for LFD2NX-28 devices in Figure A.1. Nexus Primary Clock Sources and Distribution, LIFCL-40, LFD2NX-40, and LFD2NX-28 Devices. • Added support for LFD2NX-9 devices in Figure A.2. Nexus Primary Clock Sources and Distribution, LIFCL-17, LFD2NX-17, and LFDD2NX-9 Devices.

Revision 2.4, February 2024

Section	Change Summary
Disclaimers	Updated disclaimers.
Inclusive Language	Added inclusive language boilerplate.
sysCLOCK PLL	Updated the lower range for the CLKI: Frequency (MHz) parameter from 10 MHz to 18 MHz in Table 14.6. Tab 1, General Settings, IP Catalog User Interface to match the data sheet specifications.
References	Updated references.
Technical Support Assistance	Updated link to submit a technical support case.

Revision 2.3, September 2023

Section	Change Summary
Introduction	• Added LIFCL-33U in Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers. • Rephrased *usage guide* to *user guide*.
Nexus Top-Level View	• Added CrossLink-NX-33 and CrossLink-NX-33U support. • Updated title of Figure 3.4. CrossLink-NX-33 and CrossLink-NX-33U Clocking Structure.
Clocking Architecture Overview	Added CrossLink-NX-33U devices in Primary Clock Network section.
sysCLOCK PLL	Updated title of Figure 14.7. PLL Input Pins for LIFCL-33 and LIFCL-33U.
Appendix A. Primary Clock Sources and Distribution	Updated title of Figure A.5. Nexus Primary Clock Sources and Distribution, LIFCL-33 and LIFCL-33U Devices.
References	Added this section.

Revision 2.2, February 2023

Section	Change Summary
Dynamic Clock Select (DCS)	• Updated Table 8.2. DCS – DCSMODE Attribute. • Replaced VCC with DCS in DCS Usage in VHDL and DCS Usage in Verilog sections.
All	Removed references to Divider Phase Shift method of dynamic PLL adjustment.

sysCLOCK PLL	• Deleted below sections:
	  • PHASELOADREG Input Section
	  • Divider Phase Shift Section
	  • Total Phase Shift Section
	• Deleted Figure 14.11. Divider Phase Shift Timing Diagram

Technical Support Assistance	Added FAQ website link in Technical Support Assistance.

Revision 2.1, July 2022

Section	Change Summary
sysCLOCK PLL	Added sentence "The reference clock must be stable before the RST signal is deasserted." in CLKI Input and RST Input sections.

Section	Change Summary
General Routing for Clocks	Changed the paragraph from "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." to "The user must group this logic (UGROUP) with a BBOX (see Lattice Radiant Help > Constraints Reference Guide > Preferences > UGROUP) and specify a PROHIBIT PRIMARY on the generated clock. The PROHIBIT_PRIMARY constraint allows the pin to be used as a clock source while the BBOX constraint is also included to ensure that timing closure can be obtained even without using a dedicated PCLK pin."

Revision 2.0, June 2022

Section	Change Summary
Introduction	Updated Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers to add LIFCL-33.
Nexus Top-Level View	Added Figure 3.4. CrossLink-NX-33 Clocking Structure.
sysCLOCK PLL	• Added information on one PLL for LIFCL-33 in sysCLOCK PLL Overview. • Added Figure 14.7. PLL Input Pins for LIFCL-33.
Appendix A. Primary Clock Sources and Distribution	Added Figure A.5. Nexus Primary Clock Sources and Distribution, LIFCL-33 Devices.

Revision 1.9, May 2022

Section	Change Summary
All	• Added MachXO5-NX support across the document. • Changed SERDES to SerDes across the document.
Introduction	Updated Table 1.1. Number of PLLs, Edge Clocks, and Clock Dividers to add LFMXO5.
sysCLOCK PLL	• Added information on two PLLs for LFMXO5 in sysCLOCK PLL Overview. • Added Figure 14.6. PLL Input Pins for LFMXO5.

Revision 1.8, February 2022

Section	Change Summary
Overview of Clocking Components	Changed oscillator output frequency from +/-15% to +/-7% in Section 5.6.
Dynamic Clock Control (DCC)	Added Table 9.2. DCC Component Attribute Definition for DCCEN information.

Revision 1.7, December 2021

Section	Change Summary
All	• Minor adjustments in formatting across the document. • Changed document title from sysCLOCK PLL Design and Usage Guide for Nexus Platform to sysCLOCK PLL Design and User Guide for Nexus Platform.
Introduction	Updated Table 1.1.
Nexus Top-Level View	Added Figure 3.3.
sysCLOCK PLL	Updated sysCLOCK PLL Overview content to add info on two PLLs and added Figure 14.6.

Revision 1.6, September 2021

Section	Change Summary
Nexus Top-Level View	Changed '4' to '6' for PLL blocks in Figure 3.2.

Revision 1.5, July 2021

Section	Change Summary
Acronyms in This Table	Added LMMI definition.
Dynamic Clock Control	Changed GPLL to PLL instance.
General Routing for Clocks	Added entry for PROHIBIT PRIMARY as a workaround for non-PCLK located clock nets.
sysCLOCK PLL	• Changed PLLCLK to PLLCK in Dedicated PLL Inputs, including Figure 14.2 to Figure 14.5. • Updated Table 14.3 to correct the values in PLL Output Shifted column. • Changed PHASESEL[2:0]=3'b000 to PHASESEL[2:0]=3'b001 in VCO Phase Shift. • Added Total Phase Shift, Fractional-N Synthesis Operation, and Spread Spectrum Clock Generation sections.
Appendix C.PLL LMMI Operation	Added this section.

Revision 1.4, June 2021

Section	Change Summary
All	• Minor formatting across the document. • Added CertusPro-NX support across the document
Introduction	Updated section content, including Table 1.1.
Clock/Control Distribution Network	Updated section content to add CertusPro-NX.
Nexus Top-Level View	Updated section content, including Figure 3.2.
Clocking Architecture Overview	Updated section content to add CertusPro-NX.
Overview of Clocking Components	• Changed section title from Overview of Other Clocking Elements to Overview of Clocking Components. • Changed PCIKDIV to PCLKDIV in Primary Clock Divider (PCLKDIV). • Updated content to add CertusPro-NX in Dynamic Clock Select (DCS).
Primary Clocks	• Updated content, including adding two bullet points in Primary Clock Sources. • Updated content in Primary Clock Routing, including Figure 6.1 and Figure 6.2.
Primary Clock Divider	• Updated Table 7.2. • Changed PLKDIVF to PCLKDIV in PCLKDIV Usage in Verilog.
Dynamic Clock Select	Updated section content, including Figure 8.1 and Figure 8.2.
Dynamic Clock Control	Updated section content, including adding DCC Usage in Verilog.
Internal Oscillator	Updated section content to include CertusPro-NX.
Edge Clocks	• Added bullet point for Bottom PLL Outputs. • Updated Table 11.2. Updated codes in ECLKDIV Usage in VHDL and ECLKDIV Usage in Verilog.
Edge Clock Synchronization	Updated codes in ECLKSYNC Usage in Verilog.
sysCLOCK PLL	• Updated section content, including Figure 14.4 and Figure 14.5. • Updated Table 14.1, Table 14.3, Table 14.6, and Table 14.7. Updated equations in VCO Phase Shift and Divider Phase Shift.
Appendix A. Primary Clock Sources and Distribution	Added Figure A.3 and Figure A.4.

Revision 1.3, November 2020

Section	Change Summary
All	Changed some OSC instances to *OSCA*.
Internal Oscillator	• Updated Table 10.1 and Table 10.2. • Updated codes in OSCA Usage in VHDL and OSCA Usage in Verilog section.
sysCLOCK PLL	Updated Table 14.6.

Revision 1.2, June 2020

Section	Change Summary
All	• Changed document name to sysCLOCK PLL Design and Usage Guide for Nexus Platform. • Changed CrossLink-NX to Nexus across the document.
Nexus Top-Level View	Updated content.
sysCLOCK PLL	• Updated content to add LFD2NX-17 and LFD2NX-40. • Moved PLL Features to this section. • Updated Table 14.7.

Revision 1.1, April 2020

Section	Change Summary
PLL Features	Updated Figure 7.1 and 7.2.
Primary Clocks	Updated Figure 6.1.
sysCLOCK PLL	Updated Figure 16.3.

Revision 1.0, November 2019

Section	Change Summary
All	Initial release