sysCLOCK PLL Design and Usage Guide for Nexus Platform

Technical Note

FPGA-TN-02095-1.3

November 2020

Disclaimers

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

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
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
MIB	Memory Interface Block
PLC	Programmable Logic Cell
PLL	Phase Locked Loop
SED	Soft Error Detect

1. Introduction

This usage guide describes the clock resources available in the Lattice Nexus™ Platform architecture, which includes CrossLink™-NX and Certus™-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-40 LFD2NX-40	LIFCL-17 LFD2NX-17
Number of PLLs	General purpose Phase Locked Loops.	3	2
Number of Edge Clocks	Edge Clocks for high speed applications.	12	12
Number of Edge Clock Dividers	Edge Clock Dividers for high speed applications.	12	12
Number of Primary Clock Dividers	Programmable Clock dividers for domain crossing applications.	1	1
Number of DDRDLLs	DDRDLL used for DDR memory and High Speed I/O interfaces	2	2

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

2. Clock/Control Distribution Network

Nexus devices provide global clock distribution in the form of 32 global primary clocks. There are two clock regions and each clock region can accommodate 16 primary clocks. However, 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 the Nexus devices is shown in Figure 3.1. The shaded blocks (PCIe, upper left PLL, and I/O Bank 2/Bank 6/Bank 7) are not available in the LIFCL-17 and LFD2NX-17 device. The MIPI_DPHY0 and MIPI_DPHY1 on the top are only available for the CrossLink-NX family.

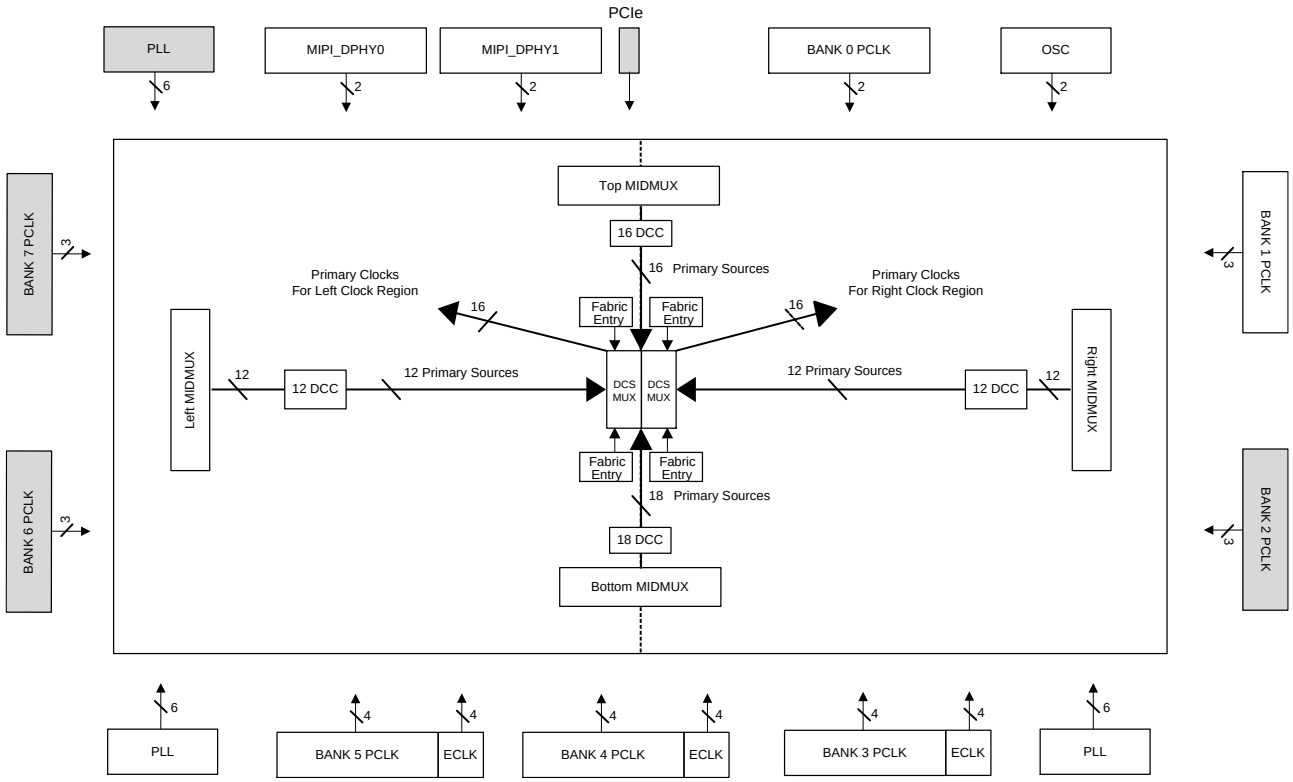

Figure 3.1. Nexus Clocking Structure

4. Clocking Architecture Overview

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

4.1. Primary Clock Network

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

The Primary Clock Network provides low-skew, high fan-out clock distribution to all synchronous elements in the FPGA fabric. The Primary Clock Network is divided into two clocking regions, 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. You 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 Other Clocking Elements

5.1. Edge Clock Dividers (ECLKDIV)

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

5.2. Primary Clock Divider (PCIKDIV)

A programmable Primary Clock Divider is provided to create the divided down clocks.

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 Nexus device.

5.4. Dynamic Clock Control (DCC)

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

5.5. Edge Clock Sync (ECLKSYNC)

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 you to synchronize the edge clock to an event or external signal, if desired. It also allows the design to dynamically disable a clock and its associated logic in the design when is it 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, having a +/- 15% variation in its output frequency. It is mainly used for circuits that do not require a high degree of clock accuracy. Examples of usage are asynchronous logic blocks such as a timer or reset generator, or other logic that require a constantly running clock.

6. 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 get to the primary clock routing are:

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

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

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. The primary clock routing network is divided into two sections, left and right, called regions. Figure 6.1 is the simplified view of Figure 3.1.

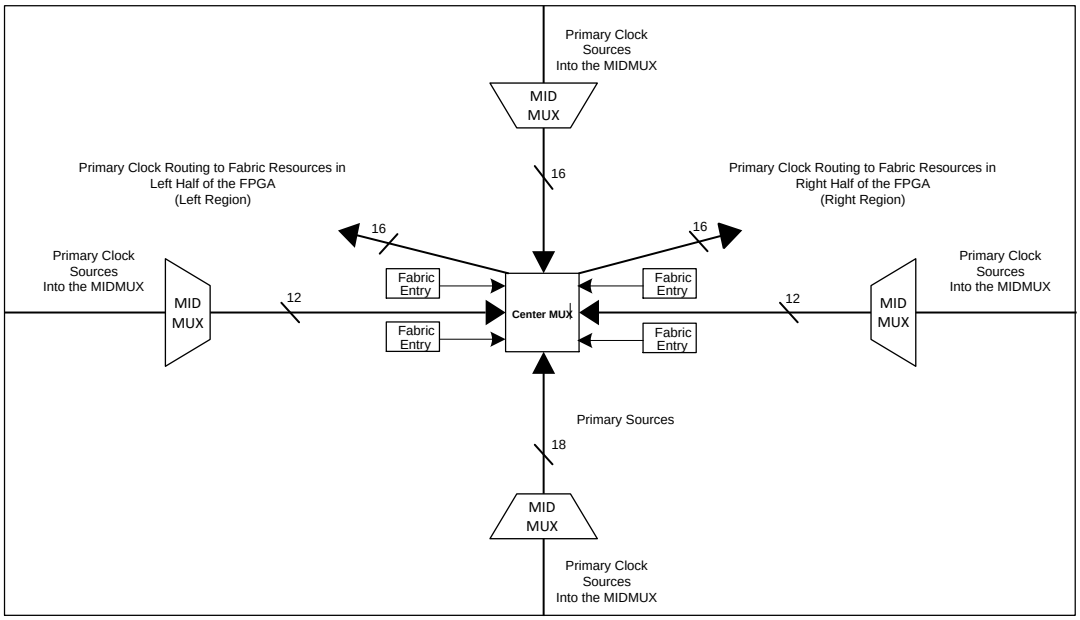

Figure 6.1. Primary Clock Routing Architecture

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

If an external input clock is being sourced to a PLL, then in most cases, the input clock should use a dedicated PLL input pin 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 (PCLKDIV)

There is one Primary Clock Divider available in the Nexus device, located inside the centermux. The 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 disable 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. PCLKDIV Component Definition

The PCLKDIV 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 PCLKDIV component. Verilog and VHDL instantiations are included.

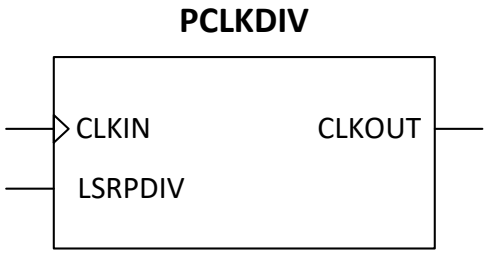

Figure 7.1. PCLKDIV Component Symbol

Table 7.1. PCLKDIV 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	Divide by 1, 2, 4, 8, 16, 32, 64, or 128 Output Port

Table 7.2. PCLKDIV Component Attribute Definition

Name	Value	Default	Description
DIV_PCLKDIV	÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, ÷128	÷1	Primary Clock Divide Ratio Selection
GSR	ENABLE DISABLE	ENABLED	GSR ENABLE/DISABLE Selection

7.2. PCLKDIV Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

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

PCLKDIV Instantiation

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

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

7.3. PLKDIVF Usage in Verilog

Component and Attribute Declaration

```
module PCLKDIV (CLKIN, LSRPDIV, CLKOUT);

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

input  CLKIN, LSRPDIV;
output CLKOUT;
endmodule
```

PCLKDIV Instantiation

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

8. Dynamic Clock Select (DCS)

The Nexus device has one dynamic clock select (DCS) block located at the center of the PLC array core, which can drive to any or all the regions. The DCS_CMUX Structure is shown in Figure 8.1.

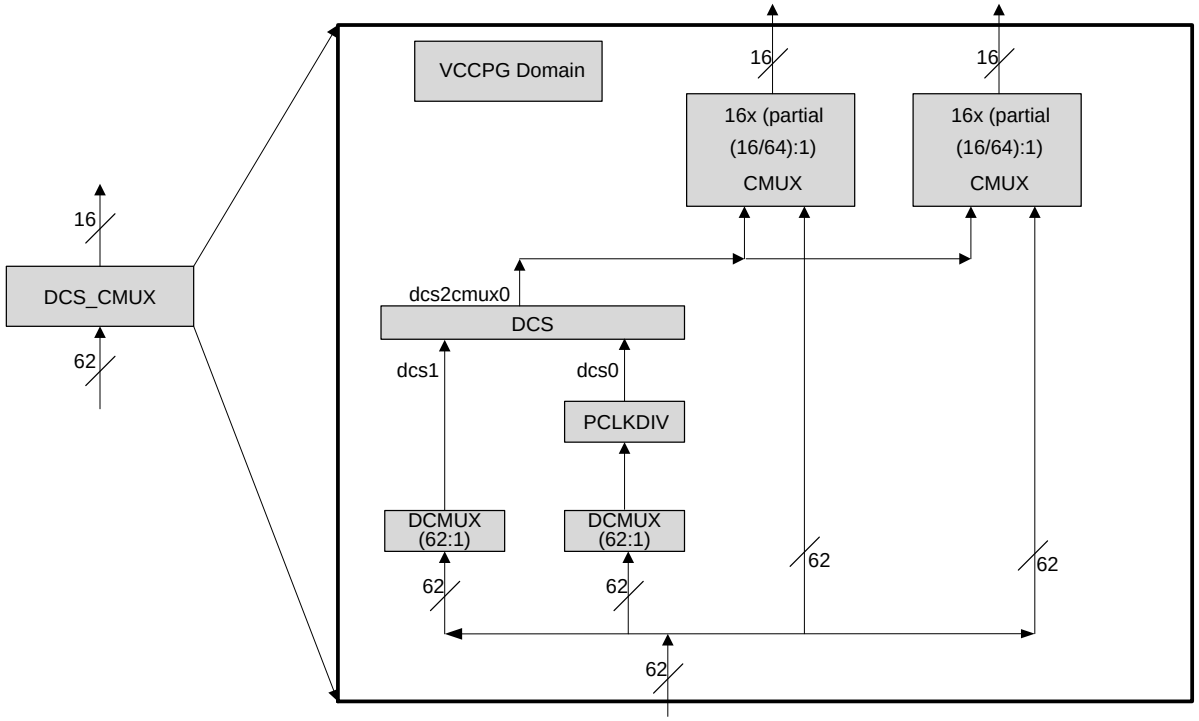

Figure 8.1. DCS_CMUX Structure

The DCS block allows dynamic and glitchless selection between two PCLK clock sources. The DCS block share the same clock resource as any PCLK CMUX. This way 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 core. The output of the DCS is connected to the inputs of Primary Clock Center MUXs. The DCS logic structure is shown in Figure 8.2.

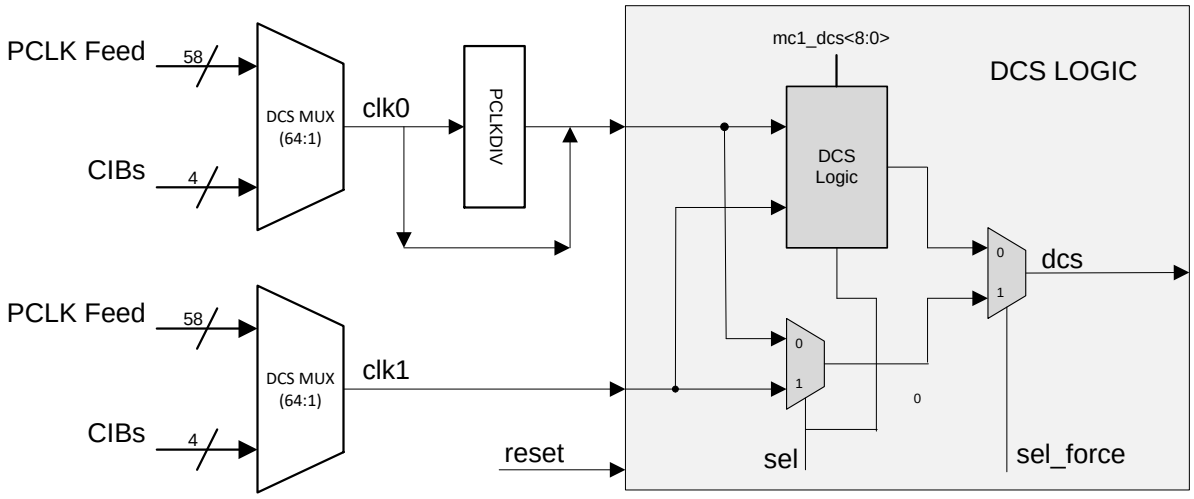

Figure 8.2. DCS Logic Structure

For glitchless operation, the *DCSMODE* attribute sets the behavior of the DCS output. Additional attribute values and their functions are shown 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 share the same clock resource as any PCLK CMUX. Therefore the DCS function can be performed on any two primary clock sources.

Figure 8.3, Figure 8.4, and Figure 8.5 show the DCS in glitchless operation in conjunction with the DCSMODE attribute. Figure 8.6 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.

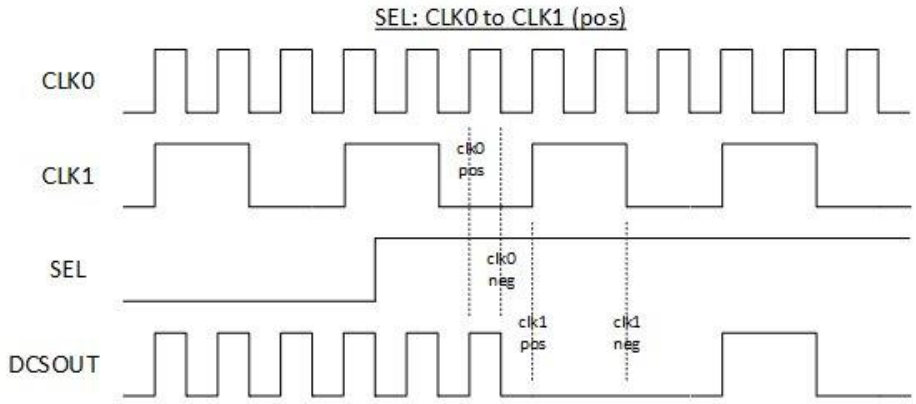

Figure 8.3. Posedge DCS Switch from sel: 0 => 1

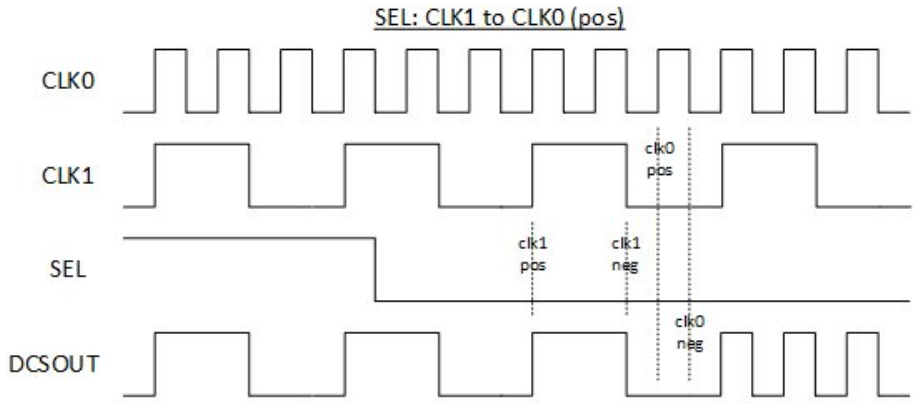

Figure 8.4. Posedge DCS Switch from sel: 1 => 0

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.

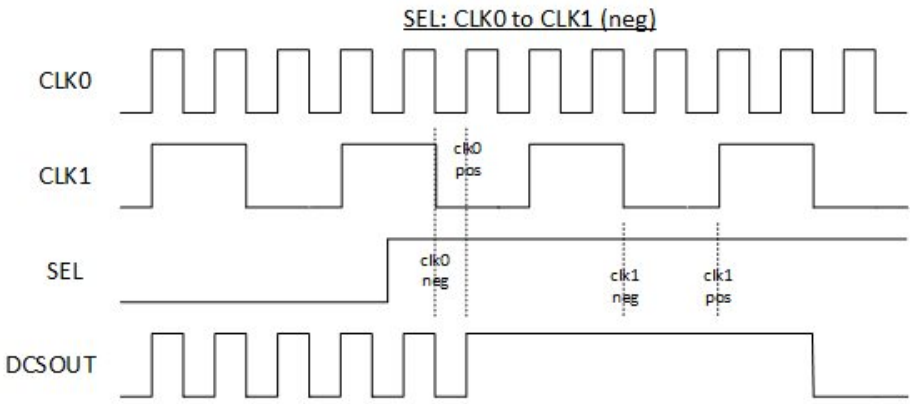

Figure 8.5. Negedge DCS Switch from sel: 0 => 1

8.1.3. Functionality – bypass

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

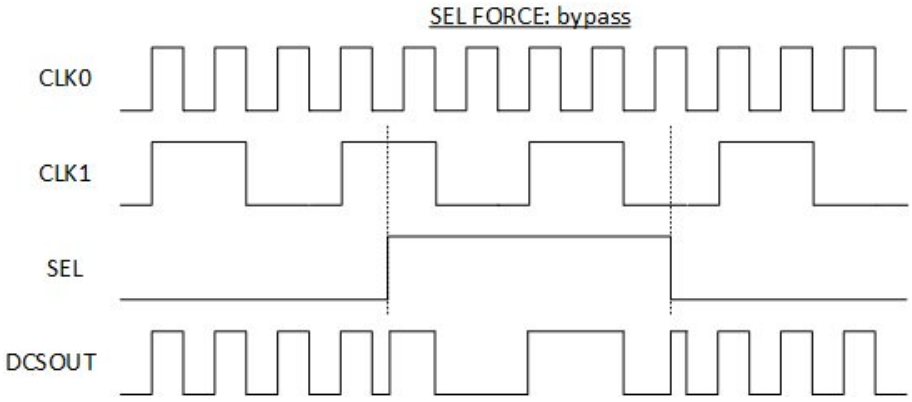

Figure 8.6. SELFORCE = 1 DCS Clock Switch Glitches

8.2. DCS Component Definition

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

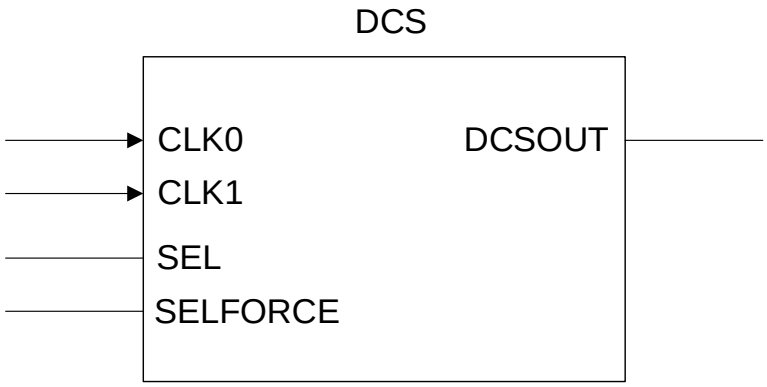

Figure 8.7. DCS Component Symbol

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
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 MODESEL = *0*.

Table 8.2. DCS – DCSMODE Attribute

Attribute Name	Attribute Value	Output		Description
		SEL = 0	SEL = 1	
DCSMODE (SELFORCE = 0)	VCC	Clk0	Clk1	Rising edge triggered. Latched state is high.
	GND	Clk0	Clk1	Falling edge triggered. Latched state is low.
	BUFGCECLK1_0	0	CLK1	SEL is active high. Disabled output is low
	BUFGCECLK1	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 := "VCC");
PORT   (CLK0 :IN STD_LOGIC;
  CLK1 :IN STD_LOGIC;
  SEL :IN STD_LOGIC;
  SELFORCE :IN STD_LOGIC;
  DCSOUT :OUT STD_LOGIC);
END COMPONENT;
```

DCS Instantiation

```
attribute DCSMODE : string;
attribute DCSMODE of DCSinst0 : label is "VCC";
I1: DCS
generic map(
   DCSMODE => "VCC")
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 = "VCC";
DCS DCSInst0 (
.CLK0     (CLK0),
.CLK1     (CLK1),
.SEL      (SEL),
.SELFORCE  (SELFORCE),
.DCSOUT   (DCSOUT));
```

9. Dynamic Clock Control (DCC)

9.1. 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.

This 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, all the logic fed by that clock does not toggle, hence, reducing the overall power consumption of the device.

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 GPLL. The DCC should remain enabled, otherwise it will cause the GPLL to loss of lock.

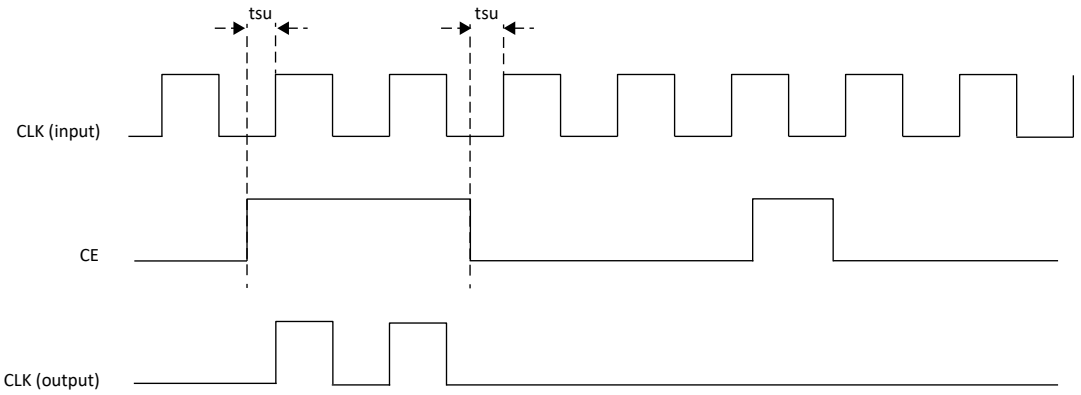

Figure 9.1. Glitchless DCC Functional Waveform

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

9.2. Component Definition

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

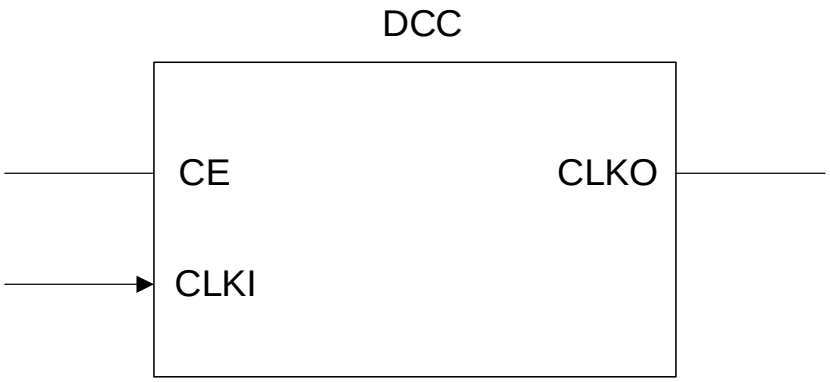

Figure 9.2. DCCA Component Symbol

Table 9.1. DCCA 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

9.3. 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;
```

DCCA 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
```

DCCA Instantiation

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

10. Internal Oscillator (OSCA)

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

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

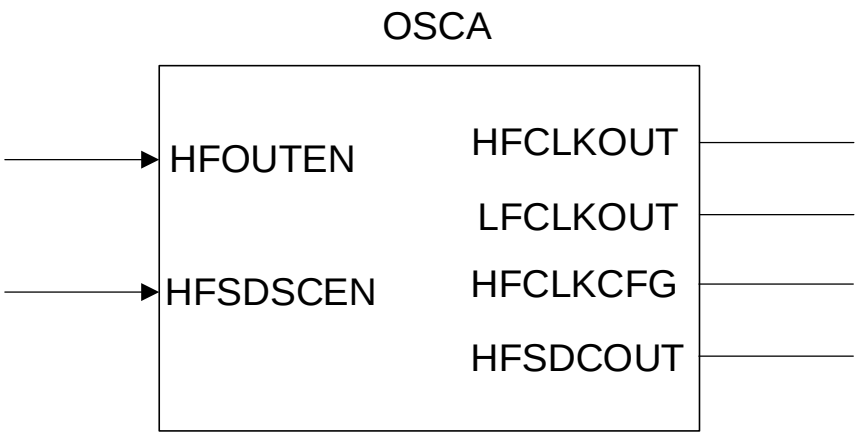

Figure 10.1. OSC Component Symbol

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

```
omponent 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)
);
```

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
- PLL outputs (CLKOP and CLKOS)
- ECLK Bridge
- Internal nodes

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

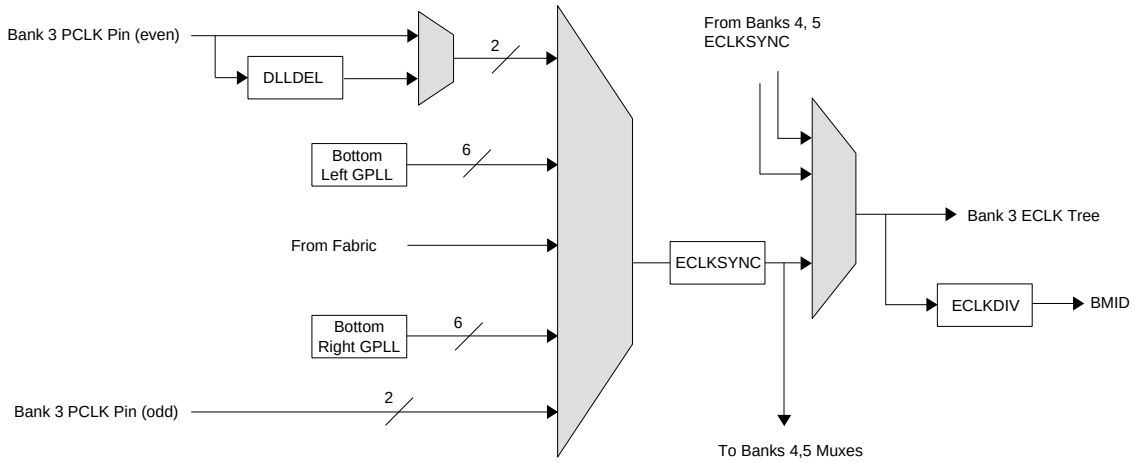

Figure 11.1. Edge Clock Sources Per Bank

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
GSRN_ECLK	ENABLE DISABLE	ENABLED	GSR ENABLE/DISABLE Selection
ECLK_DIV	“1” “2” “3P5” “4” “5”	1	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;
         GSRN_ECLK     : string);
Port    (DIVRST     : in STD_LOGIC;
         ECLKIN     : in STD_LOGIC;
         SLIP   : in STD_LOGIC;
         DIVOUT    : out STD_LOGIC);
end component;
```

ECLKDIV Instantiation

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

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

11.4. CLKDIVF Usage in Verilog

Component and Attribute Declaration

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

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

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

CLKDIVF Instantiation

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

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 you to synchronize the Edge Clock to an event or external signal if desired. It also allows the design to dynamically disable a clock and its associated logic in the design when is it 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.

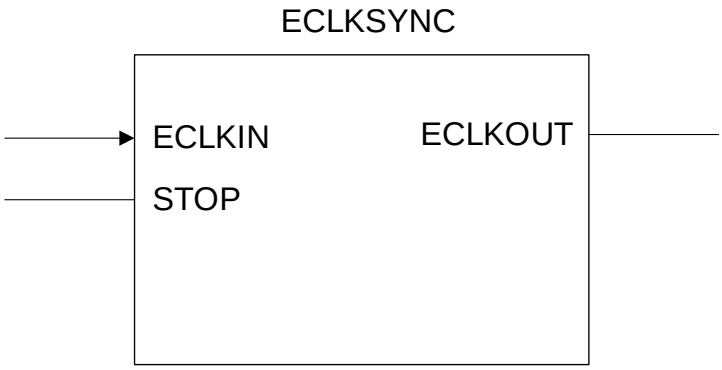

Figure 12.1. ECLKSYNC Component Symbol

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

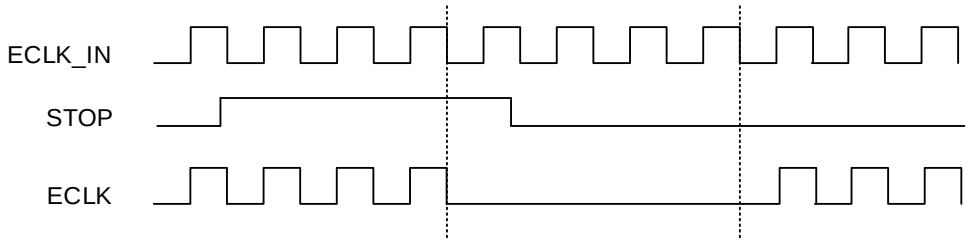

Figure 12.2. ECLKSYNC Functional Waveform

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);
ECLKSYNC Usage in Verilog
Component and Attribute Declaration
module ECLKSYNC (ECLKIN,STOP,ECLKOUT);
input   ECLKIN;
input   STOP;
output  ECLKOUT;
endmodule
```

ECLKSYNCB Instantiation

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

13. General Routing for Clocks

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

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

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

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

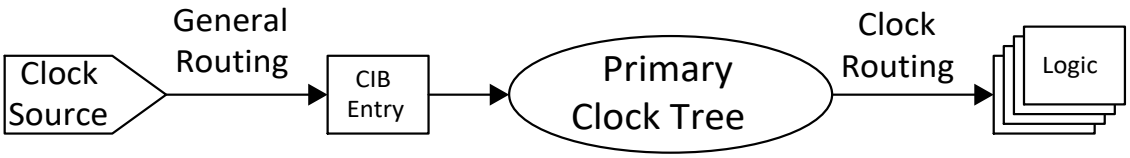

Figure 13.1. Gated Clock to the Primary Clock Routing

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

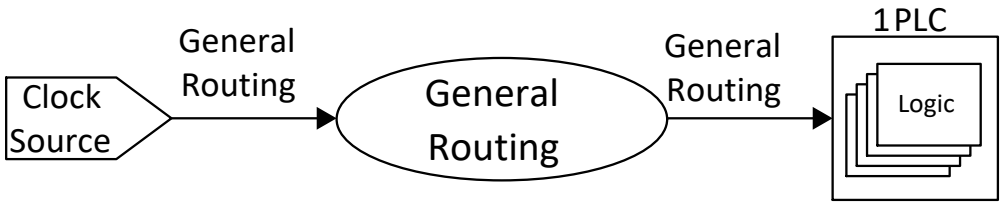

Figure 13.2. Gated Clock to Small Logic Domain

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

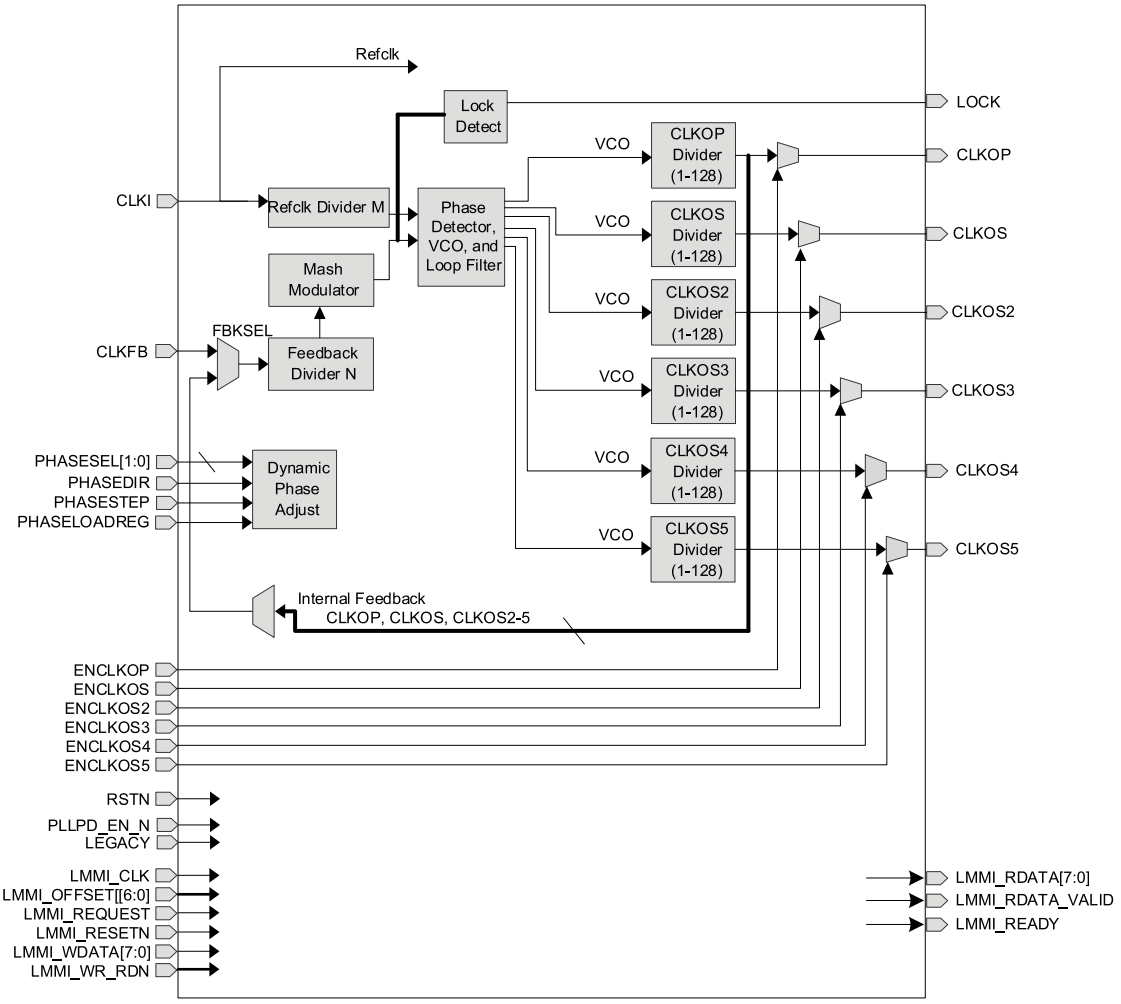

Figure 14.1. Nexus PLL Block Diagram

There are three PLLs on the larger density devices (LIFCL-40 and LFD2NX-40) and two PLLs on the smaller density (LIFCL-17 and LFD2NX-17) devices. There is a PLL on three corners (Upper Left, Lower Left and Lower Right) of the larger density devices and the smaller density device has two PLLs, one each in the Lower Left and Lower Right corners. Each PLL has six outputs. All six PLL outputs can feed the Primary Clock and Edge Clock networks.

14.2. PLL Features

14.2.1. Dedicated PLL Inputs

Every PLL has a dedicated low skew input (PLLCLK) that routes directly to its reference clock input. These are the recommended inputs for a PLL. It is possible to route a PLL input from the Primary clock routing, but it incurs more clock input injection delays, which are not natively compensated for using feedback, compared to a dedicated PLL input. There is one PLLs in each corner of the FPGA on bigger densities. Each PLL on the Nexus device has one pair dedicated PLL input pin. The top PLL, which is available only for LIFCL-40 and LFD2NX-40, has the dedicated input pin from the upper left bank. The bottom two PLLs each have a dedicated input pin from a bottom bank.

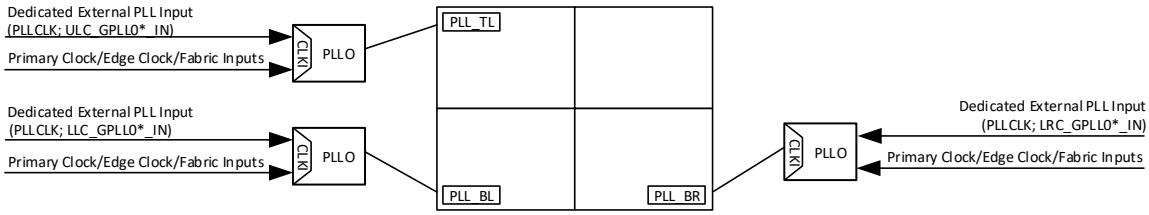

Figure 14.2. PLL Input Pins for LIFCL-40 and LFD2NX-40

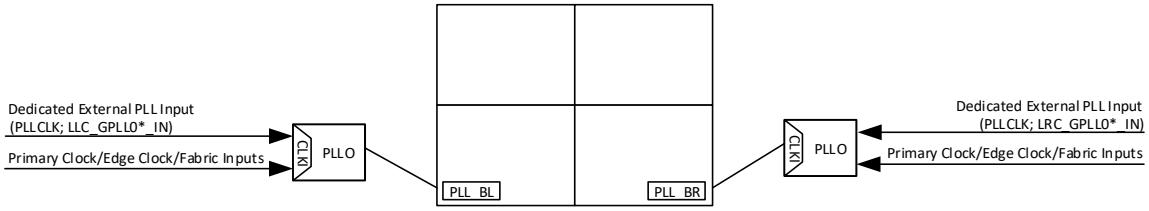

Figure 14.3. PLL Input Pins for LIFCL-17 and LFD2NX-17

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 clock to out timing and remove the delay differences between the PLL output clock and the data input. This feature is performed by aligning the input clock with a feedback clock from the clock tree. Optional delay may also be added to the feedback path to further reduce the clock injection time.

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 PLL, 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.

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

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 14.4 and Table 14.1 show the OSCA definitions.

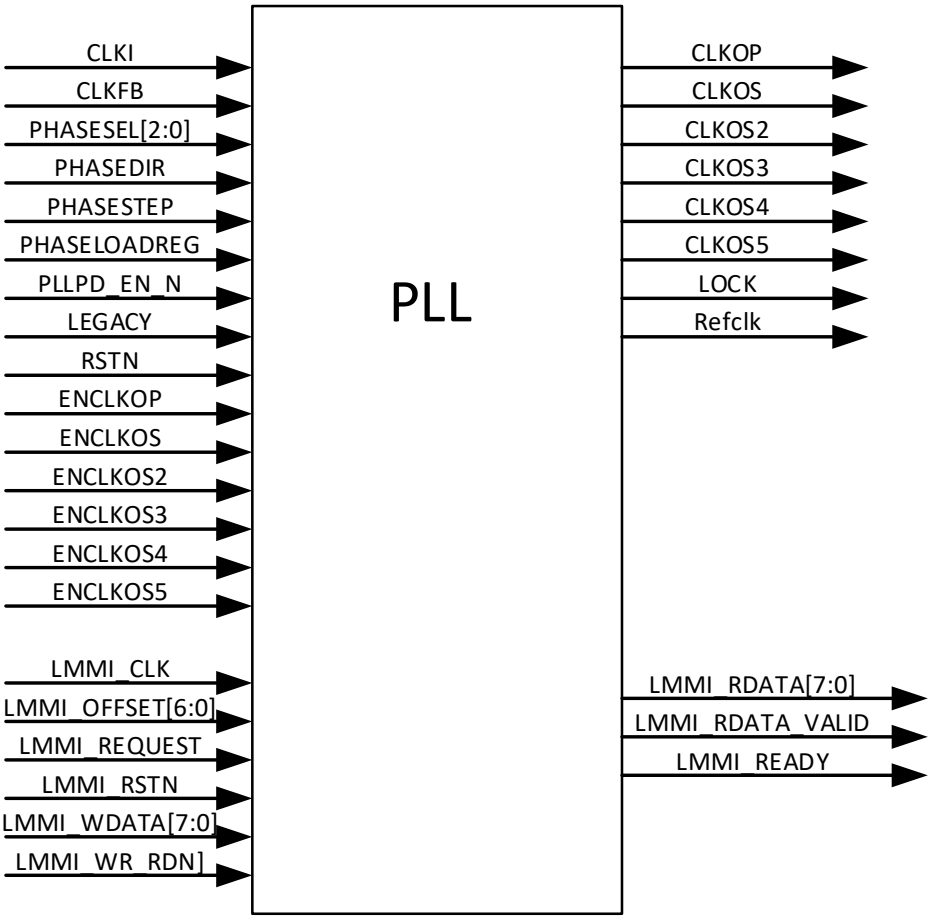

Figure 14.4. PLL Component Instance

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	Load dynamic phase adjustment values into PLL.
PLLPD_EN_N	I	Standby signal to power down the PLL.
LEGACY	I	Power mode setting to enable legacy mode
RST	I	Resets the whole 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	PLL LOCK 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 PLL block. The valid 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 you selected in the software.

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

The output Clock Dividers allow the VCO frequency to be scaled up to the maximum range to minimize jitter. Each of the output dividers is independent of the other dividers and each uses the VCO as the source by default. Each of the output dividers can be set to a value of 1 to 128.

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 device, phase adjustments are now calculated values in the software tools based on VCO clock frequency. This provides a finer phase shift depending on the required frequency. The clock output selected as the feedback cannot use the static phase adjustment feature since it causes the PLL to unlock.

14.4.5. Phase Adjustment (Dynamic Mode)

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

Similar restrictions apply to other clocks.

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.

14.5.2. CLKFB Input

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

14.5.3. RST Input

At power-up, an internal power-up reset signal from the configuration block resets the PLL. At runtime, an active high, asynchronous, user-controlled PLL reset signal can be provided as a part of the PLL module. The RST signal can be driven by an internally generated reset function or by an I/O pin. This RST signal resets the PLL core (VCO, phase detector, and charge pump) and the output dividers which cause the outputs to be logic *0*. In bypass mode, the output does not reset.

After the RST signal is deasserted, the PLL starts the lock-in process and takes tLOCK time, about 16 ms, to complete PLL lock. Figure 14.5 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.

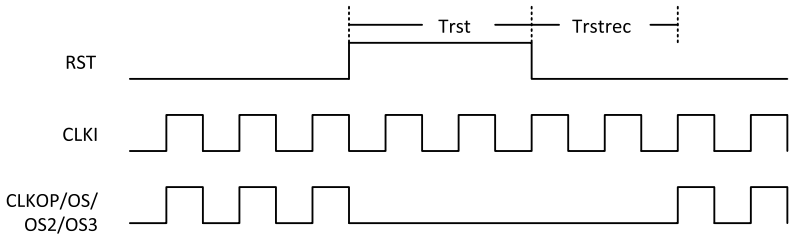

Figure 14.5. RST Input Timing Diagram

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

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 you select the *Enable Powerdown Mode* in the IP Catalog wizard. The PLLPD_EN_N signal is active low. When asserted, the PLL outputs are pulled to *0* and the PLL is reset. You need to stay in the Power Down mode for at least 1 ms to make sure the PLL analog circuits are fully reset and to have a stable analog startup.

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 you 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 or PHASELOADREG signals are pulsed. The PHASESEL signal is optional and is available if you select the *Dynamic Phase Ports* option in IP Catalog Wizard.

Table 14.3. PHASESEL Signal Settings Definition

PHASESEL[1:0]	PLL Output Shifted
000	CLKOS
001	CLKOS2
010	CLKOS3
011	CLKOS4
100	CLKOS5
101	CLKOP

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 or PHASELOADREG signals are pulsed. The PHASEDIR signal is optional and is available if you select the Dynamic Phase ports option in IP Catalog Wizard.

Table 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 cycles of VCO running period. The PHASESTEP signal is optional and is available if you select the Dynamic Phase ports option in IP Catalog Wizard. The PHASESEL and PHASEDIR are required to have a setup time of 5 ns prior to PHASESTEP falling edge.

14.5.10. PHASELOADREG Input

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

14.5.11. PLL Clock Outputs

The PLL has four outputs, listed in Table 14.5. All four outputs can be routed to the Primary clock routing of the FPGA. All four outputs can be phase shifted statically or dynamically if external feedback on the clock is not used. They can also statically or dynamically adjust their output duty cycle. The outputs can come from their output divider or the reference clock input (PLL bypass). In bypass mode, the output divider can be bypassed or used to divide the reference clock.

Table 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 via IP Catalog
CLKOS2	No ECLK Connection	Selectable via IP Catalog
CLKOS3	No ECLK Connection	Selectable via IP Catalog
CLKOS4	No ECLK Connection	Selectable via IP Catalog
CLKOS5	No ECLK Connection	Selectable via IP Catalog

14.5.12. LOCK Output

The LOCK output provides information about the status of the PLL. After the device is powered up and the input clock is valid, the PLL achieves lock within 16 ms. Once lock is achieved, the PLL LOCK signal is asserted. The LOCK signal can be set in IP Catalog Wizard in either the default *unsticky* frequency lock mode by checking the *Provide PLL Lock Signal* or sticky lock mode by selecting *PLL Lock is Sticky*. In sticky lock mode, once the LOCK signal is asserted (logic *1*), it stays asserted until a PLL reset is asserted. In the default lock mode of *unsticky* frequency lock, if during operation the input clock or feedback signals to the PLL become invalid, the PLL loses lock and the LOCK output de-asserts (logic *0*). It is recommended to assert PLL RST to re-synchronize the PLL to the reference clock when the PLL loses lock. The LOCK signal is available to the FPGA routing to implement the generation of the RST signal if requested by the designer. The LOCK signal is optional and is available if you select the Provide PLL Lock signal option in IP Catalog Wizard.

14.6. Dynamic Phase Adjustment

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

All six output clocks, CLKOP, CLKOS, CLKOS2, CLKOS3, CLKOS4, and CLKOS5 have the dynamic phase adjustment feature but only one output clock can be adjusted at a time. Table 14.5 shows the output clock selection settings available for the PHASESEL[1:0] signal. The PHASESEL signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed.

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

14.6.1. VCO Phase Shift

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

```
(CLKO<n>_FPHASE/ (8*CLKO<n>_DIV)]*360
```

Where <n> is the clock output specified by PHASESEL (CLKOP/OS/OS2/OS3). Values for CLKO<n>_FPHASE and CLKO<n>_DIV are located in the HDL source file.

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

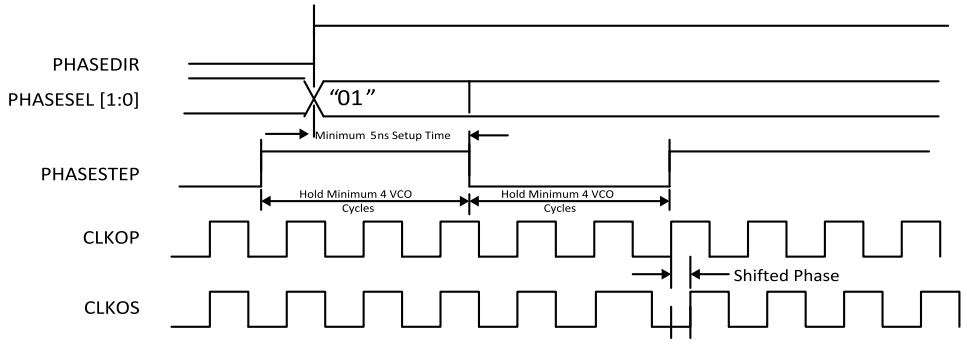

Figure 14.6. PLL Phase Shifting Using the PHASESTEP Signal

For Example:

PHASESEL[2:0]=3'b000 to select CLKOS for phase shift

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

Assume the output is divided by 2, CLKOS_DIV = 2

The CLKOS_FPHASE is set to 1.

The above signals need to be stable for 5 ns before the falling edge of PHASESTEP and the minimum pulse width of PHASESTEP should be four VCO clock cycles. It should also stay low for four VCO Clock Cycles.

For each toggling of PHASESTEP, you are getting [1/(8*2)]*360 = 22.5 degree phase shift (delayed).

14.6.2. Divider Phase Shift

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

```
[(CLKO<n>_CPHASE – CLKO<n>_DIV)/( CLKO<n>_DIV + 1)] * 360
```

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

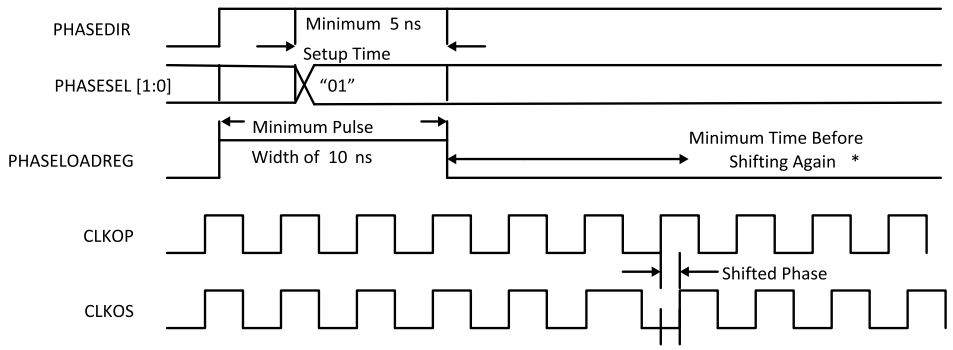

Figure 14.7. Divider Phase Shift Timing Diagram

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

14.7. Low Power Features

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

14.7.1. Dynamic Clock Enable

The Dynamic Clock Enable feature allows you to glitchlessly enable and disable selected output clocks during periods when not used in the design. A disabled output clock is logic *0*. Re-enabled clocks start on the falling edge of CLKOP. To support this feature, each output clock has an independent Output Enable signal that can be selected. The Output Enable signals are ENCLKOP, ENCLKOS, ENCLKOS2, ENCLKOS3, ENCLKOS4, and ENCLKOS5. Each clock enable port has an option in the IP Catalog user interface to bring the signal to the top level ports of the PLL. If external feedback is used on a port or if the clock output is not enabled, its dynamic clock enable port is unavailable.

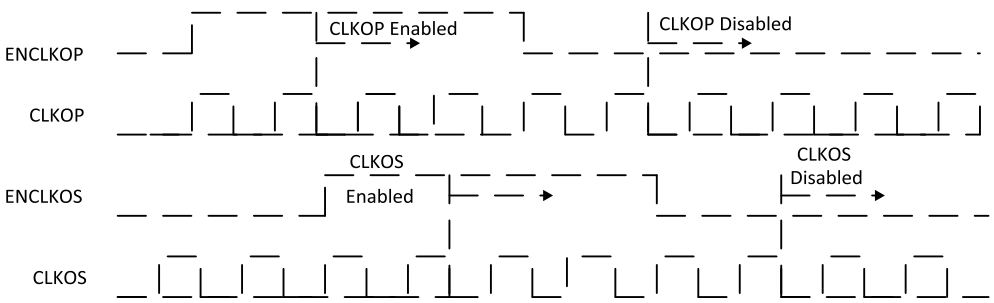

Figure 14.8. Dynamic Clock Enable for PLL Outputs

14.7.2. Standby Mode

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

14.8. 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 14.9. 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 14.9.

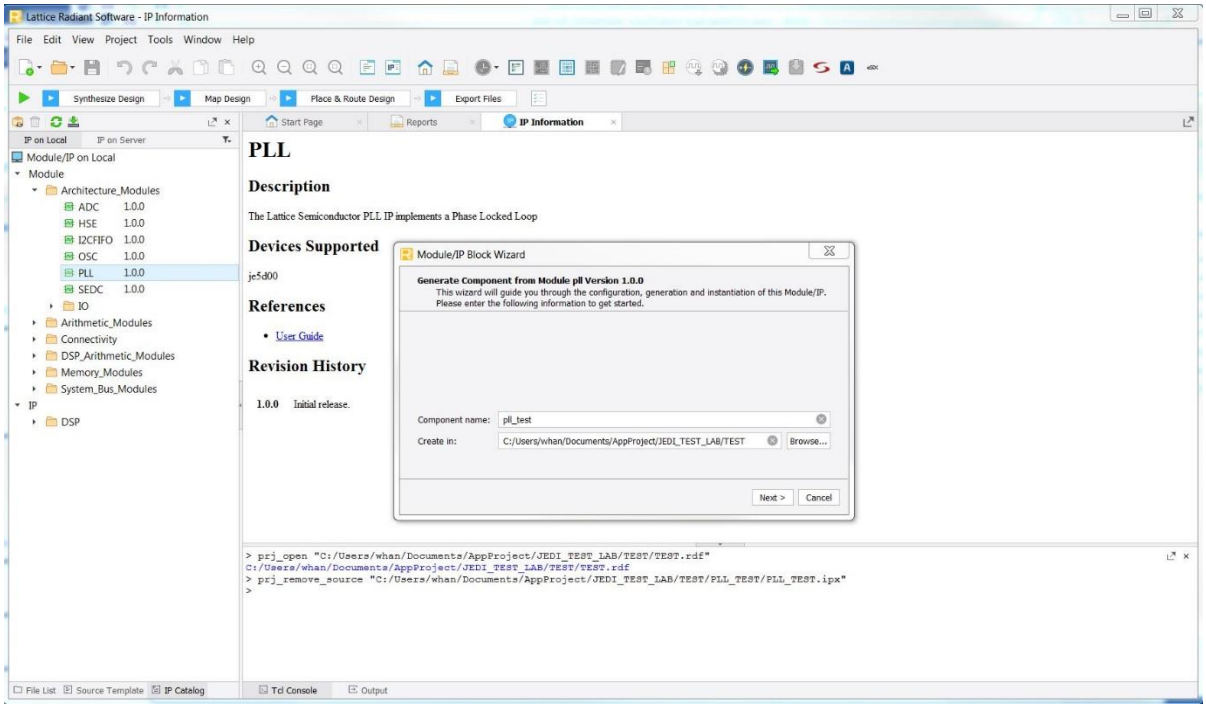

Figure 14.9. IP Catalog Main Window for PLL Module

14.8.1. Configuration Tab

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

14.8.2. PLL Frequency and Phase Configuration

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

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

General Tab

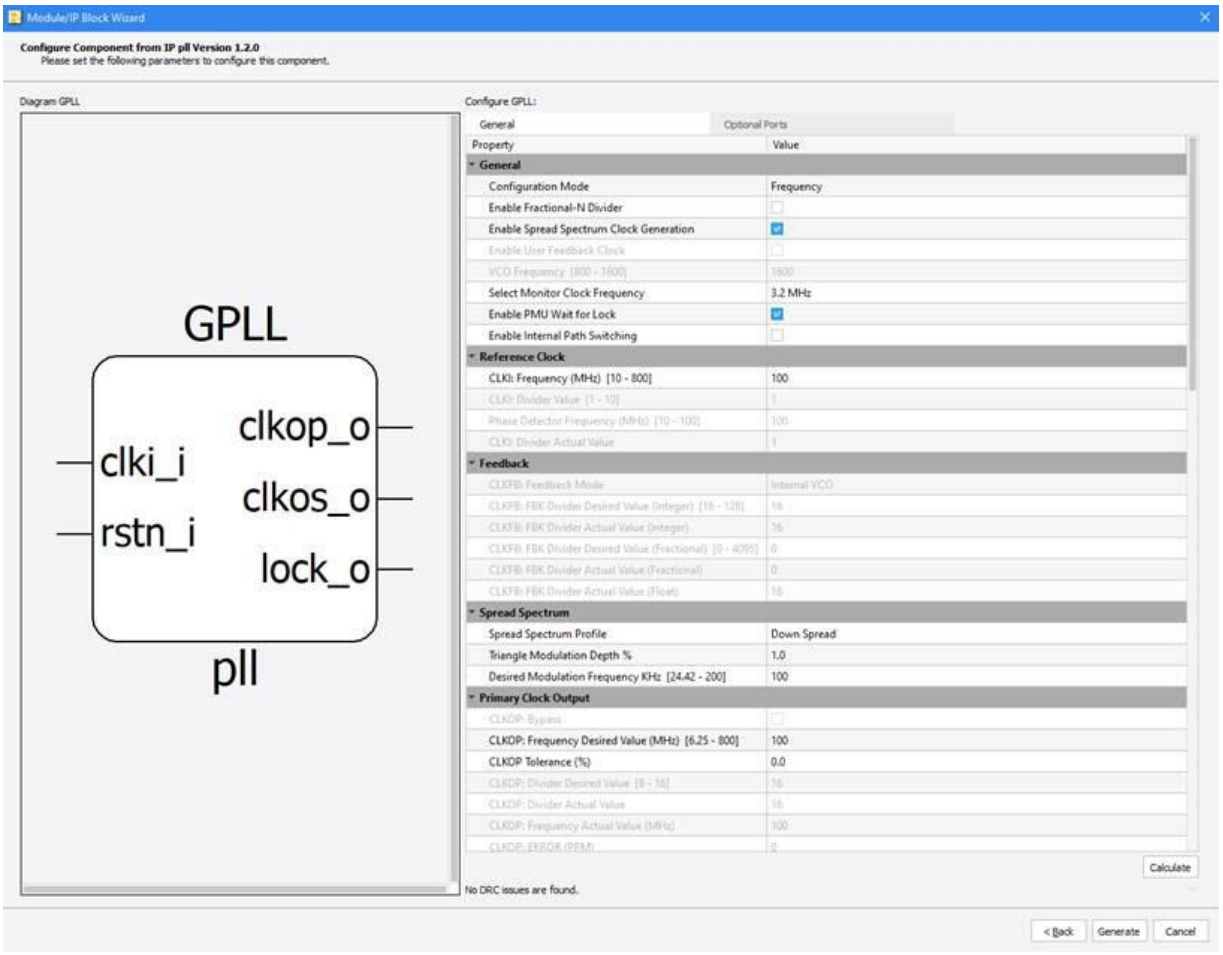

Figure 14.10. Nexus PLL Frequency Configuration in General Tab

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 frequency. 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 Spectrum Clock Generation	Checked, Unchecked	Unchecked	Enable/Disable the Spread Spectrum Clock Generation.
Enable User Feedback Clock	Checked, Unchecked	Unchecked	When enabled, feedback clock will be 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)	10 – 800	100	Set the Reference Clock frequency. (applicable for Frequency mode only)
CLKI: Divider Desired Value	1 to 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	1 to 128	1	Set the Feedback Clock divider. (applicable for Divider mode only)
CLKFB: FBK Divider Actual Value	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)
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)

User Parameters	Range	Default	Description
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 to 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

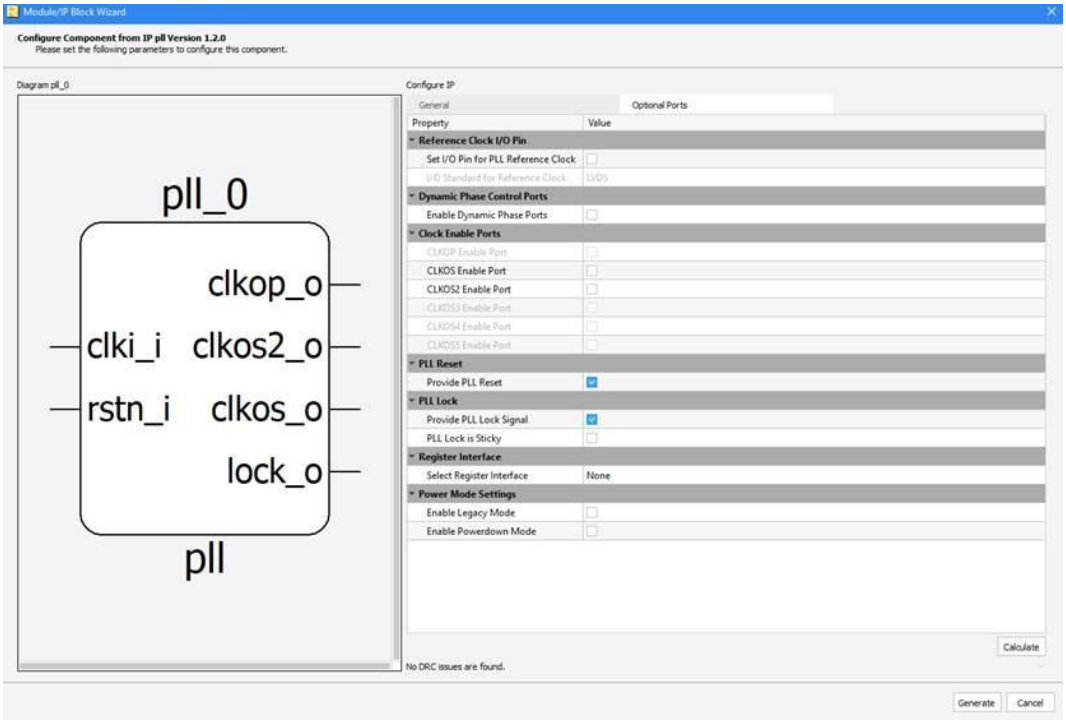

Figure 14.11. Nexus PLL Optional Ports Configuration Tab

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	LVDS	Select type of I/O pin.
Dynamic Phase Control Ports			
Enable Dynamic Phase ports	Checked, Unchecked	Unchecked	Enable/Disable dynamic phase control ports.
Clock Enable Ports			
CLKOP/CLKOS[n] Enable Port	Checked, Unchecked	Unchecked	Set to provide clock enable port.
PLL Reset			
Provide PLL Reset	Checked, Unchecked	Unchecked	Set to provide PLL reset port.
PLL Lock			
Provide PLL Lock Signal	Checked, Unchecked	Unchecked	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 Power Down 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, FREQUENCY preferences is included in the preference file to automatically constrain the clocks produced by the PLL. For a step-by-step guide to using IP Catalog, refer to the IP Catalog user manual.

Appendix A. Primary Clock Sources and Distribution

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

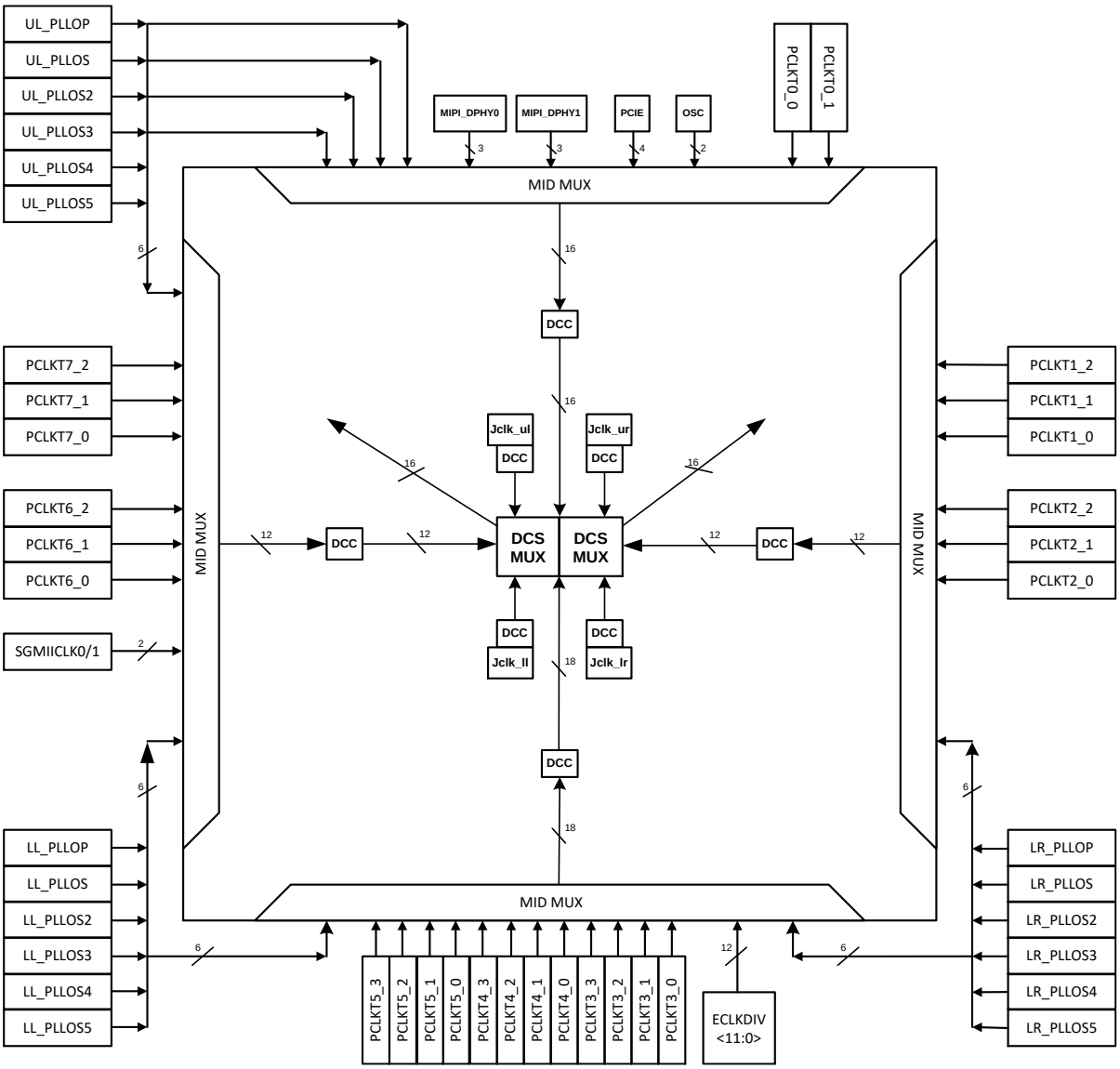

Figure A.1. Nexus Primary Clock Sources and Distribution, LIFCL-40 and LFD2NX-40 Devices

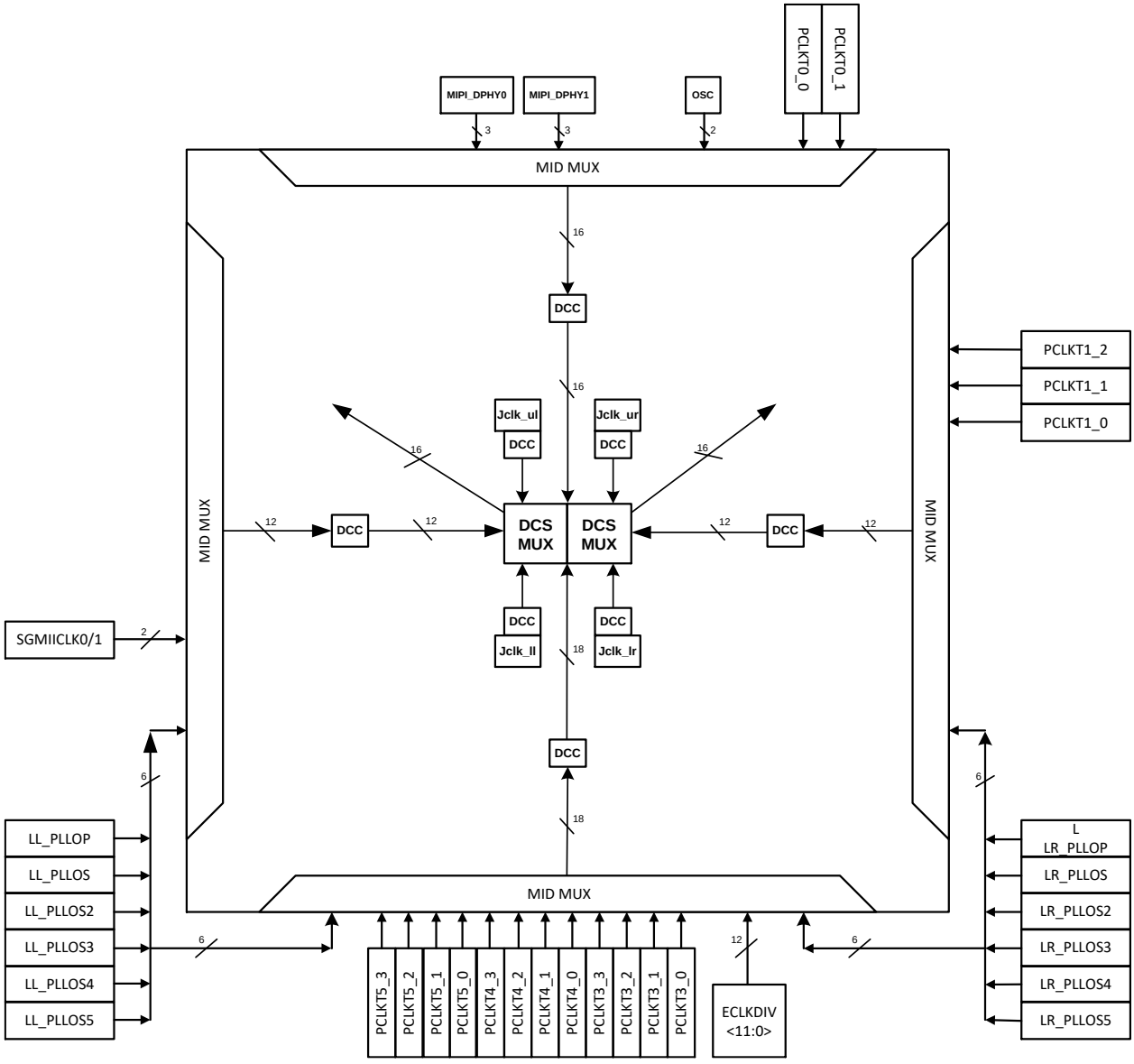

Figure A.2. Nexus Primary Clock Sources and Distribution, LIFCL-17 and LFD2NX-17 Devices

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, you can see the PCLK and PLL input pins listed as below:

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

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.

Technical Support Assistance

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

Revision History

Revision 1.3, 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

www.latticesemi.com