CrossLinkPlus sysCLOCK PLL/DLL Design and Usage Guide

Technical Note

FPGA-TN-02109-1.0

August 2019

Disclaimers

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

Contents

Acronyms in This Document 6
1. Introduction 7
2. Clock/Control Distribution Network 7
3. CrossLinkPlus Device Top-Level View 7
4. Clocking Architecture Overview 8
4.1. Primary Clock Network 8
4.2. Edge Clock Network 8
5. Overview of Other Clocking Elements 8
5.1. Edge Clock Dividers (CLKDIVG) 8
5.2. Edge Clock Synchronization Control (ECLKSYNCB) 8
5.3. Oscillator (OSCI) 8
6. sysCLOCK™ PLL 9
6.1. Functional Block Description 10
6.1.1. Refclk (CLKI) Divider 10
6.1.2. Feedback Clock (CLKFB) Divider 10
6.1.3. Output Clock Dividers (CLKOP, CLKOS, CLKOS2, CLKOS3) 10
6.1.4. Phase Adjustment (Static Mode) 10
6.1.5. Phase Adjustment (Dynamic Mode) 11
6.2. PLL Features 11
6.2.1. Dedicated PLL Inputs 11
6.2.2. PLL Input Clock Mux (PLLREFCS) 11
6.2.3. Standby Mode 11
6.2.4. ECLK Dedicated Feedback to PLL 11
6.3. PLL Inputs and Outputs 12
6.3.1. CLKI Input 12
6.3.2. CLKFB Input 12
6.3.3. USRSTDBY Input 12
6.3.4. RST Input 12
6.3.5. Dynamic Clock Enable 13
6.3.6. Dynamic Phase Shift Inputs 13
6.3.7. PLL Clock Outputs 14
6.4. Dynamic Phase Adjustment 14
6.4.1. VCO Phase Shift 15
6.4.2. Divider Phase Shift 16
6.5. Low Power Features 16
6.6. PLL Usage in Clarity Designer 17
6.6.1. Configuration Tab 18
6.6.2. PLL Frequency and Phase Configuration 18
7. Primary Clocks 23
7.1. Primary Clock Sources 23
7.2. Primary Clock Routing 23
7.3. Dedicated Clock Inputs 23
8. Internal Oscillator (OSCI) 24
8.1. OSCI Component Definition 24
8.2. OSCI Usage in VHDL 25
8.3. OSCI Usage in Verilog 25
9. Edge Clocks 26
9.1. Edge Clock Dividers (CLKDIVG) 26
9.2. CLKDIVG Component Definition 27
9.3. CLKDIVG Usage in VHDL 27
9.4. CLKDIVG Usage in Verilog HDL 28
10. Edge Clock Synchronization (ECLKSYNCB) 29

10.1. ECLKSYNCB Component Definition 29
10.2. ECLKSYNCB Usage in VHDL 30
10.3. ECLKSYNCB Usage in Verilog HDL 30
11. General Routing for Clocks 31
12. General Routing PCLK Pins 31
13. PLL Reference Clock Switch Primitive (PLLREFCS) 32
13.1. PLLREFCS Usage in VHDL 32
13.2. PLLREFCS Usage in Verilog HDL 33
References 34
Technical Support Assistance 35
Revision History 36

Figures

Figure 3.1. CrossLinkPlus Clocking Structure 7
Figure 6.1. CrossLinkPlus PLL Block Diagram 9
Figure 6.2. PLL Component Instance 9
Figure 6.3. PLL Dedicated Inputs to the PLLREFCS Component 11
Figure 6.4. RST Input Timing Diagram 12
Figure 6.5. PLL Phase Shifting Using the PHASESTEP Signal 15
Figure 6.6. Divider Phase Shift Timing Diagram 16
Figure 6.7. Dynamic Clock Enable for PLL Outputs 17
Figure 6.8. Clarity Designer Main Window for PLL Module 17
Figure 6.9. CrossLinkPlus PLL Frequency Configuration Tab 18
Figure 6.10. CrossLinkPlus PLL Phase Configuration Tab 20
Figure 6.11. CrossLinkPlus PLL Optional Ports Configuration Tab 21
Figure 8.1. OSCI Component Symbol 24
Figure 9.1. Edge Clock Sources per Bank 26
Figure 9.2. CLKDIVG Component Symbol 27
Figure 10.1. ECLKSYNCB Component Symbol 29
Figure 10.2. ECLKSYNCB Functional Waveform 29
Figure 11.1. Gated Clock to the Primary Clock Routing 31
Figure 11.2. Gated Clock to Small Logic Domain 31
Figure 13.1. PLLREFCS Component Symbol 32

Tables

Table 1.1. Key Clocking Resources 7
Table 6.1. EHXPLLM Component Port Definition 10
Table 6.2. USRSTDBY Signal Settings Definition 12
Table 6.3. PLL Clock Output Enable Signal List 13
Table 6.4. PHASESEL Signal Settings Definition 13
Table 6.5. PHASEDIR Signal Settings Definition 13
Table 6.6. PLL Clock Outputs and ECLK Connectivity 14
Table 6.7. Page 1, PLL Frequency Settings, Clarity Designer GUI 19
Table 6.8. Tab 2, PLL Phase Settings, Clarity Designer GUI 20
Table 6.9. Tab 2, PLL Optional Ports, Clarity Designer GUI 22
Table 8.1. OSCI Component Port Definition 24
Table 8.2. OSCI Component Attribute Definition 24
Table 9.1. CLKDIVG Component Port Definition 27
Table 9.2. CLKDIVG Component Attribute Definition 27
Table 10.1. ECLKSYNCB Component Port Definition 29
Table 13.1. PLLREFCS Component Port Definition 32

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CLKDIVG	Edge Clock Dividers
DDR	Double Data Rate
ECLKSYNCB	Edge Clock Synchronization Control
GPIO	General Purpose Input/Output
GUI	Graphic User Interface
LVDS	Low-Voltage Differential Signaling
MIPI	Mobile Industry Processor Interface
OSCI	Oscillator
PLL	Phase Locked Loop
VCO	Voltage-Controlled Oscillator

1. Introduction

This usage guide describes the clock resources available in the CrossLinkPlus™ device architecture. Details are provided for primary clocks, edge clocks, PLLs, internal oscillator, and clocking elements such as clock dividers, clock multiplexers, and clock stop blocks.

The number of PLLs, edge clocks, clock dividers, and DDRDLLs for each device is listed in Table 1.1.

Table 1.1. Key Clocking Resources

Parameter	Description	Number
Number of PLLs	General purpose PLLs.	1
Number of Edge Clocks	Edge Clocks for high-speed applications.	4
Number of Clock Dividers	Edge Clock Dividers for DDR applications.	4
Number of DDRDLLs	DDRDLL used for high-speed I/O interfaces.	2

It is important to note that you need to validate the pinout so that correct pin placement is used. The Lattice Diamond® tool should be used to validate the pinout while designing the printed circuit board.

2. Clock/Control Distribution Network

CrossLinkPlus devices provide global clock distribution in the form of eight global primary clocks. Each of the Bank 1 and Bank 2 at the bottom also has low skew and high speed edge clocks.

3. CrossLinkPlus Device Top-Level View

A top-level view of the major clocking resources for the CrossLinkPlus device is shown in Figure 3.1.

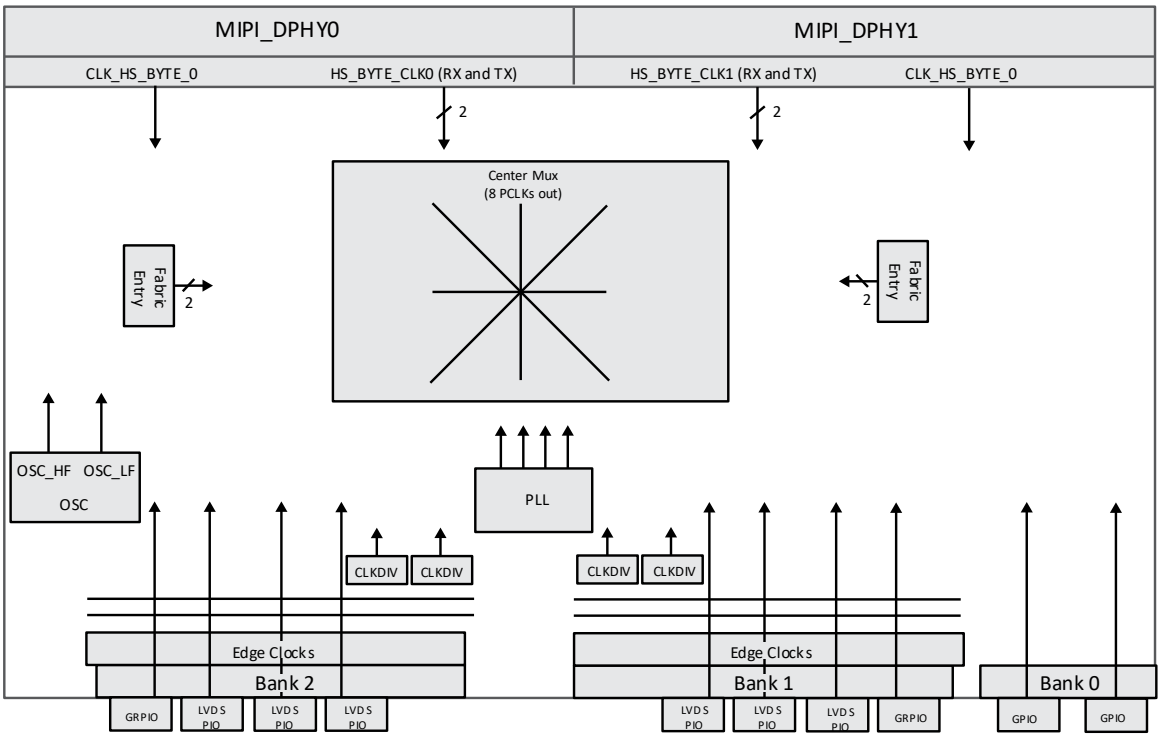

Figure 3.1. CrossLinkPlus Clocking Structure

4. Clocking Architecture Overview

Below is a brief overview of the clocking structure, elements, and PLL.

4.1. Primary Clock Network

Up to eight primary clock sources are selected and routed to the primary clock network. This gives you eight available unique clock domains in the CrossLinkPlus device. Clock sources are from LVDS PIO pins, GPIO pins, PLL, clock dividers, fabric internal generated clock signal, divided down clock from DPHY, and OSCI. All clock sources are fed into the PCLK center MUX which generates eight clock sources.

The primary clock network provides low-skew, high fanout clock distribution to all synchronous elements in the FPGA fabric.

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 the CrossLinkPlus device. There are two edge clocks located on Bank 1 and two edge clocks located on Bank 2.

5. Overview of Other Clocking Elements

5.1. Edge Clock Dividers (CLKDIVG)

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 drive to the primary clock routing to the fabric. There are four clock dividers on the CrossLinkPlus device.

5.2. Edge Clock Synchronization Control (ECLKSYNCB)

Each ECLK has a block to allow dynamic start and stop of the edge clock. You can start and stop the clock synchronous to an event or external signal. These are important for applications requiring exact clock timing relationships on inputs, such as DDR interfaces and video applications.

5.3. Oscillator (OSCI)

An internal programmable rate oscillator is provided. The oscillator can be used as a user logic clock source that is available after FPGA configuration. There is one OSCI resource on the CrossLinkPlus device. The oscillator clock outputs have access to primary clock routing resources. The CrossLinkPlus OSCI generates two clock outputs, a high frequency and a low frequency clock. OSCI port LFCLKOUT runs at 10 kHz and port HFCLKOUT runs at a maximum of 48 MHz with output divider 1, 2, 4, or 8.

6. sysCLOCK™ PLL

The CrossLinkPlus PLL provides features such as clock injection delay removal, frequency synthesis, and phase adjustment. Figure 6.1 shows a block diagram of the CrossLinkPlus PLL. Figure 6.2 shows the PLL component instance.

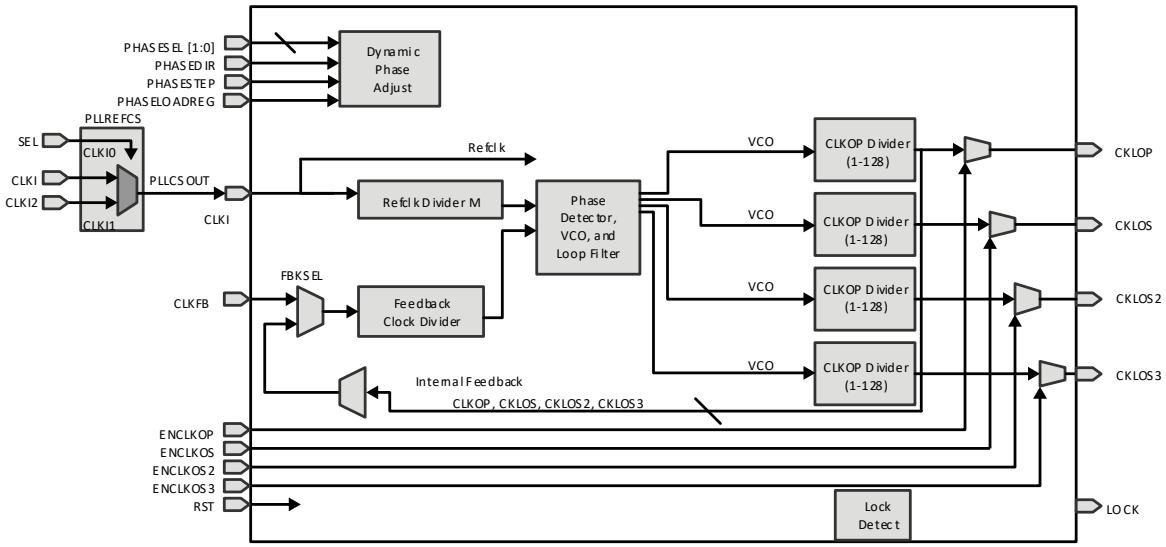

Figure 6.1. CrossLinkPlus PLL Block Diagram

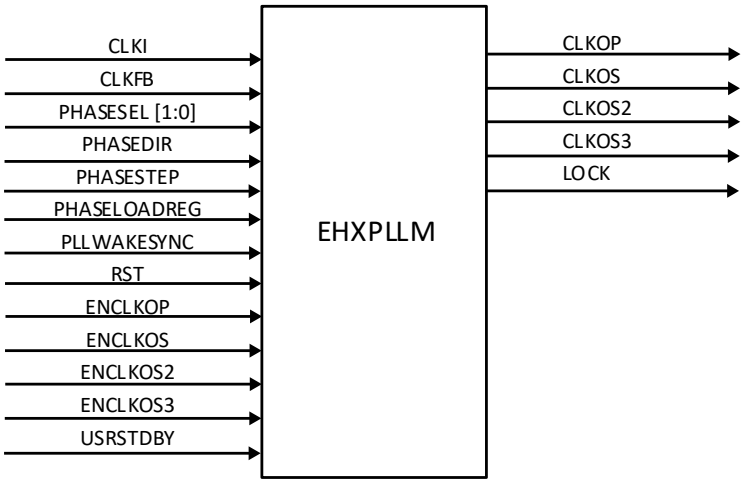

Figure 6.2. PLL Component Instance

Table 6.1. EHXPLLM Component Port Definition

Signal	I/O	Description
CLKI	I	Input clock to PLL
CLKFB	I	Feedback clock
USRSTDBY	I	User port to put the PLL in sleep mode
PHASESEL[1:0]	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
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
PLLWAKESYNC	I	Enable PLL switching from internal feedback to user feedback path when PLL wakes up
CLKOP	O	PLL main output clock
CLKOS	O	PLL output clock
CLKOS2	O	PLL output clock
CLKOS3	O	PLL output clock
LOCK	O	PLL LOCK to CLKI, Asynchronous signal. Active high indicates PLL lock

6.1. Functional Block Description

6.1.1. Refclk (CLKI) Divider

The Refclk 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.

6.1.2. Feedback Clock (CLKFB) Divider

The Feedback Clock 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 divided 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 when the **user clock** option is selected for feedback clock. Otherwise, this port is automatically connected by the tool to the appropriate signal based on the selection from the drop-down menu in the GUI.

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

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

6.1.4. Phase Adjustment (Static Mode)

The CLKOP, CLKOS, CLKOS2, and CLKOS3 outputs can be phase adjusted relative to the enabled unshifted output clock. Phase adjustments are 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. For example, if the FB_MODE is INT_OP or CLKOP, there should be no phase shift on CLKOP. A similar restriction applies on other clocks.

6.1.5. Phase Adjustment (Dynamic Mode)

The phase adjustments can also be controlled in a dynamic mode using the PHASESEL, PHASEDIR, PHASESTEP, and PHASELOADREG ports. The clock output selected as 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. For example, if the FB_MODE is INT_OP or CLKOP, there should be no phase shift on CLKOP. Similar restriction applies on other clocks.

6.2. PLL Features

6.2.1. Dedicated PLL Inputs

PLL has a dedicated low skew input that routes directly to its reference clock input. These are the recommended inputs for a PLL. It is possible to route a PLL input from the primary clock routing, but the routings are not designed and optimized for high-speed clock.

6.2.2. PLL Input Clock Mux (PLLREFCS)

CrossLinkPlus PLL contains an input mux to dynamically switch between two input reference clocks. The output of the PLLREFCS is routed directly into the PLL. To enhance the clock muxing capability of the CrossLinkPlus device, the dedicated clock inputs for the PLL are routed to the PLLREFCS component along with the other potential PLL sources such as edge clocks and primary clocks. This structure is shown in Figure 6.3.

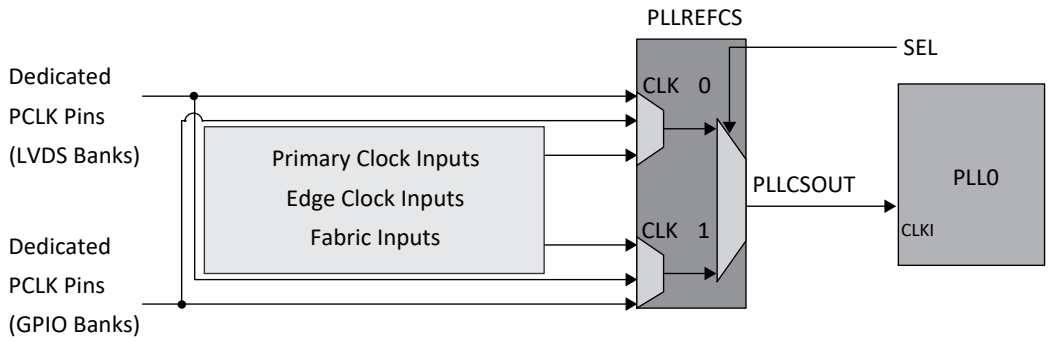

Figure 6.3. PLL Dedicated Inputs to the PLLREFCS Component

This adds a lot of flexibility for designs that need to switch between two external clocks.

Note: While switching between two external clocks, two frequencies need to be the same and output jitter may increase momentarily causing PLL to unlock.

6.2.3. Standby Mode

The PLL contains a Standby Mode that allows the PLL to be placed into a standby state to save power when not needed in the design. Standby mode is very similar to holding the PLL in reset since the VCO is turned off and needs to regain lock when exiting standby. In both reset and standby modes, the PLL retains its programming. You need to stay in the USRSTDBY mode for at least 1 ms to make sure the PLL analog circuits are fully reset and to have a stable analog startup. USRSTDBY is an active high signal.

6.2.4. ECLK Dedicated Feedback to PLL

When PLL is used as clock source for the ECLK network, such as in the application of the video function, clock feedback is needed from the ECLK network back to the PLL to ensure clock phase alignment and to cancel any clock tree delay. The way to nullify the effect of the PVT variation on the clock path is to add a dedicated identical ECLK network to provide the feedback clock source to the PLL. By doing that, no edge clock resource is lost due to this feedback and the whole ECLK tree insertion delay is compensated.

6.3. PLL Inputs and Outputs

6.3.1. CLKI Input

The CLKI signal is the reference clock for the PLL. It must conform to the specifications in the data sheet for the PLL to operate correctly. The CLKI signal can come from a dedicated PLL input pin or from internal routing. The dedicated dual-purpose I/O pin provides a low skew input path and is the recommended source for the PLL. The reference clock can be divided by the input (M) divider to create one input to the phase detector of the PLL.

6.3.2. CLKFB Input

The CLKFB signal is the feedback signal to the PLL. The feedback signal is used by the Phase Frequency Detector inside the PLL to determine if the output clock needs adjustment to maintain the correct frequency and phase. The CLKFB signal can come from a primary clock net (feedback mode = CLKO[P/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 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. A bypassed PLL output cannot be used as the feedback signal.

6.3.3. USRSTDBY Input

The USRSTDBY signal is used to put the PLL into a low power standby mode when it is not required.

Table 6.2. USRSTDBY Signal Settings Definition

USRSTDBY	PLL State
0	Normal Operation
1	Standby Mode

6.3.4. 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*.

After the RST signal is deasserted, the PLL starts the lock-in process and takes tLOCK time to complete PLL lock. Figure 6.4 shows the timing diagram of the RST input. The RST signal is active high. The RST signal is optional. Trst is the minimum reset pulse width, while Trstrec is the time after a reset before the divider output starts counting again. See the device data sheet for the specification of tLOCK, Trst, and Trstrec.

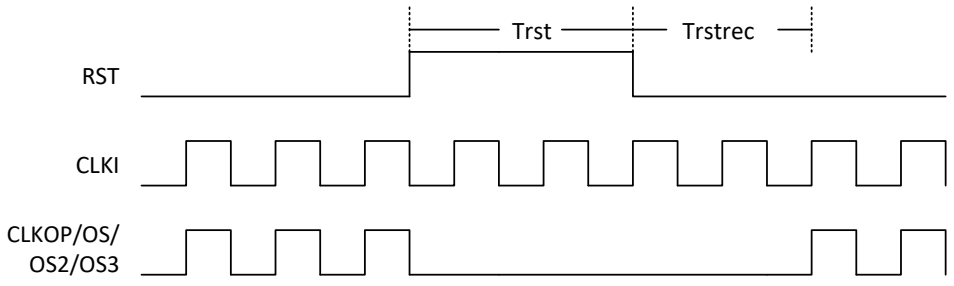

Figure 6.4. RST Input Timing Diagram

6.3.5. Dynamic Clock Enable

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 6.3. PLL Clock Output Enable Signal List

Clock Enable Signal Name	Corresponding PLL Output	Clarity Designer Option Name
ENCLKOP	CLKOP	Clock Enable OP
ENCLKOS	CLKOS	Clock Enable OS
ENCLKOS2	CLKOS2	Clock Enable OS2
ENCLKOS3	CLKOS3	Clock Enable OS3

This allows you to save power by stopping the corresponding output clock when not in use. The clock enable signals are optional and are available only when you select the corresponding option in Clarity Designer. 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 Clarity Designer when using it for external feedback to avoid shutting off the feedback clock input.

6.3.6. Dynamic Phase Shift Inputs

The PLL has five ports to allow for dynamic phase adjustment from FPGA logic. The Dynamic Phase Adjustment section provides details on how to drive these ports.

PHASESEL Input

The PHASESEL[1:0] 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 1 ns before the PHASESTEP signal is pulsed. The PHASESEL signal is optional and is available if you select the **Dynamic Phase Ports** option under **Optional Ports** tab in Clarity Designer.

Table 6.4. PHASESEL Signal Settings Definition

PHASESEL[1:0]	PLL Output Shifted
00	CLKOS
01	CLKOS2
10	CLKOS3
11	CLKOP

PHASEDIR Input

The PHASEDIR input is used to specify the direction in which the dynamic phase shift occurs, advanced (leading) or delayed (lagging). When PHASEDIR = 0, the phase shift is delayed. When PHASEDIR = 1, the phase shift is advanced. The PHASEDIR signal must be stable for 1 ns before the PHASESTEP signal is pulsed. The PHASEDIR signal is optional and is available if you select the **Dynamic Phase Ports** option under **Optional Ports** tab in Clarity Designer.

Table 6.5. PHASEDIR Signal Settings Definition

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

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 PHASESTEP pulse width, both high and low, is four cycles of the VCO running period. The PHASESTEP signal is optional and is available if you have selected the **Dynamic Phase Ports** option under **Optional Ports** tab in Clarity Designer. The PHASESEL and PHASEDIR are required to have a setup time of 1 ns prior to PHASESTEP rising edge.

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 specified by the PHASESEL input. 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 deassertion. The PHASESEL is required to have a setup time of 5 ns prior to PHASELOADREG falling edge. The PHASELOADREG signal is optional and is available if you selected the **Dynamic Phase Ports** option under **Optional Ports** tab in Clarity Designer.

6.3.7. PLL Clock Outputs

The PLL has four outputs, listed in Table 6.6. 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. The outputs can come from their output divider or the reference clock input (PLL bypass).

Table 6.6. PLL Clock Outputs and ECLK Connectivity

Clock Output Name	Edge Clock Connectivity	Selectable Output
CLKOP	Bank 1, Bank 2 ECLKs	Always Enabled
CLKOS	Bank 1, Bank 2 ECLKs	Selectable through Clarity Designer
CLKOS2	No ECLK Connection	Selectable through Clarity Designer
CLKOS3	No ECLK Connection	Selectable through Clarity Designer

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 tLOCK time. When lock is achieved, the PLL LOCK signal is asserted. The LOCK signal can be set in Clarity Designer in either the default unsticky frequency lock mode by checking the **Provide PLL Lock Signal** or in sticky lock mode by selecting **Provide PLL Lock Signal** and **PLL Lock is Sticky** under **Optional Ports** tab (Figure 6.11). In sticky lock mode, when the LOCK signal is asserted (logic *1*) it stays asserted until a PLL reset is asserted. In the default unsticky lock mode, when the PLL loses lock due to an invalid input clock or feedback signal, the LOCK output is deasserted (logic *0*) immediately. It is recommended to assert PLL RST to resynchronize 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 Signa**l option under **Optional Ports** tab in Clarity Designer. Refer to Table 6.9 for more details about the PLL optional ports.

6.4. Dynamic Phase Adjustment

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

All four output clocks, CLKOP, CLKOS, CLKOS2, and CLKOS3 have the dynamic phase adjustment feature but only one output clock can be adjusted at a time. Table 6.4 shows the output clock selection settings available for the PHASESEL[1:0] signal. The PHASESEL signal must be stable 1 ns before the PHASESTEP signal is pulsed, and remain stable for at least 5 VCO cycles afterward.

The selected output clock phase is either advanced or delayed depending upon the value of the PHASEDIR port or signal. Table 6.5 shows the PHASEDIR settings available. The PHASEDIR signal must be stable for 1 ns before the PHASESTEP signal is pulsed, and remain stable or at least 5 VCO cycles afterward.

6.4.1. VCO Phase Shift

Once the PHASESEL and PHASEDIR are set, a VCO phase adjustment is made by toggling the PHASESTEP signal from the current setting. Each Positive 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 latches PHASEDIR and PHASESEL on its rising edge. PHASESTEP must remain asserted for a minimum of four VCO cycles and is subject to a minimum wait of four VCO cycles prior to pulsing the signal again, if PHASEDIR and PHASESEL remain constant. 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. Glitches, however, may result when advancing the output clock.

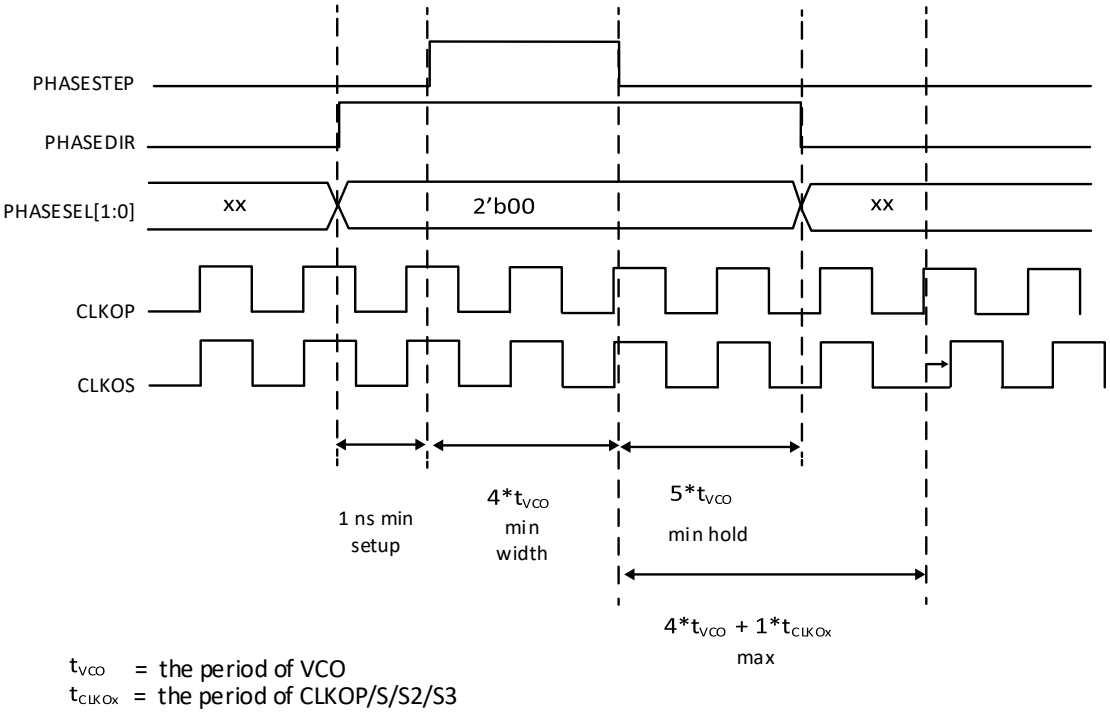

Figure 6.5. PLL Phase Shifting Using the PHASESTEP Signal

For Example:

PHASESEL[1:0]=2'b00 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 1 ns before the rising edge of PHASESTEP and the minimum pulse width of PHASESTEP should be four VCO clock cycles. It should also stay low for at least four VCO Clock Cycles if PHASEDIR and PHASESEL remain constant.

For each toggling of PHASESTEP, you can get a [1/(8*2)]*360 = 22.5° phase shift (delayed).

6.4.2. Divider Phase Shift

When the PHASESEL is 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. Note that if these values are both *1*, no shift is made.

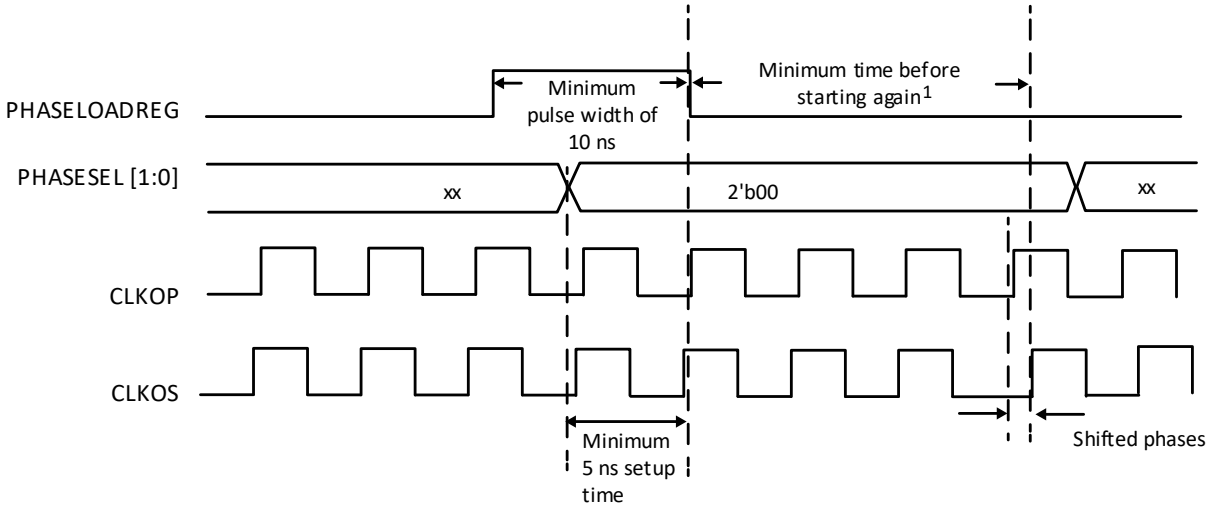

Figure 6.6. Divider Phase Shift Timing Diagram

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

6.5. Low Power Features

The CrossLinkPlus PLL contains several features that allow you to reduce the power usage of a design including Standby mode support and Dynamic Clock Enable.

Dynamic Clock Enable

The Dynamic Clock Enable feature allows you to glitchlessly enable and disable selected output clocks during periods when they are 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, and ENCLKOS3. Each clock enable port has an option in the Clarity Designer GUI 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.

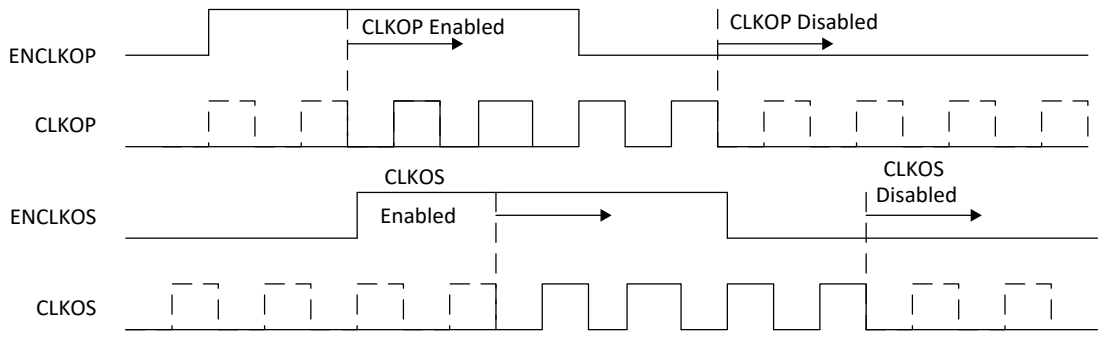

Figure 6.7. Dynamic Clock Enable for PLL Outputs

6.6. PLL Usage in Clarity Designer

Clarity Designer is used to create and configure the PLL. The general purpose PLL can be found in the **Catalog** tab of Clarity Designer under **Module** – **Architecture_Modules**, as shown in Figure 6.8. You can use the GUI to select parameters for the PLL. The result is an HDL block to be used in the simulation and synthesis flow.

When the PLL is selected, the only entry required is the file name, as other entries are set to the project settings. After entering the module name of choice, click **Customize** to open the PLL configuration window.

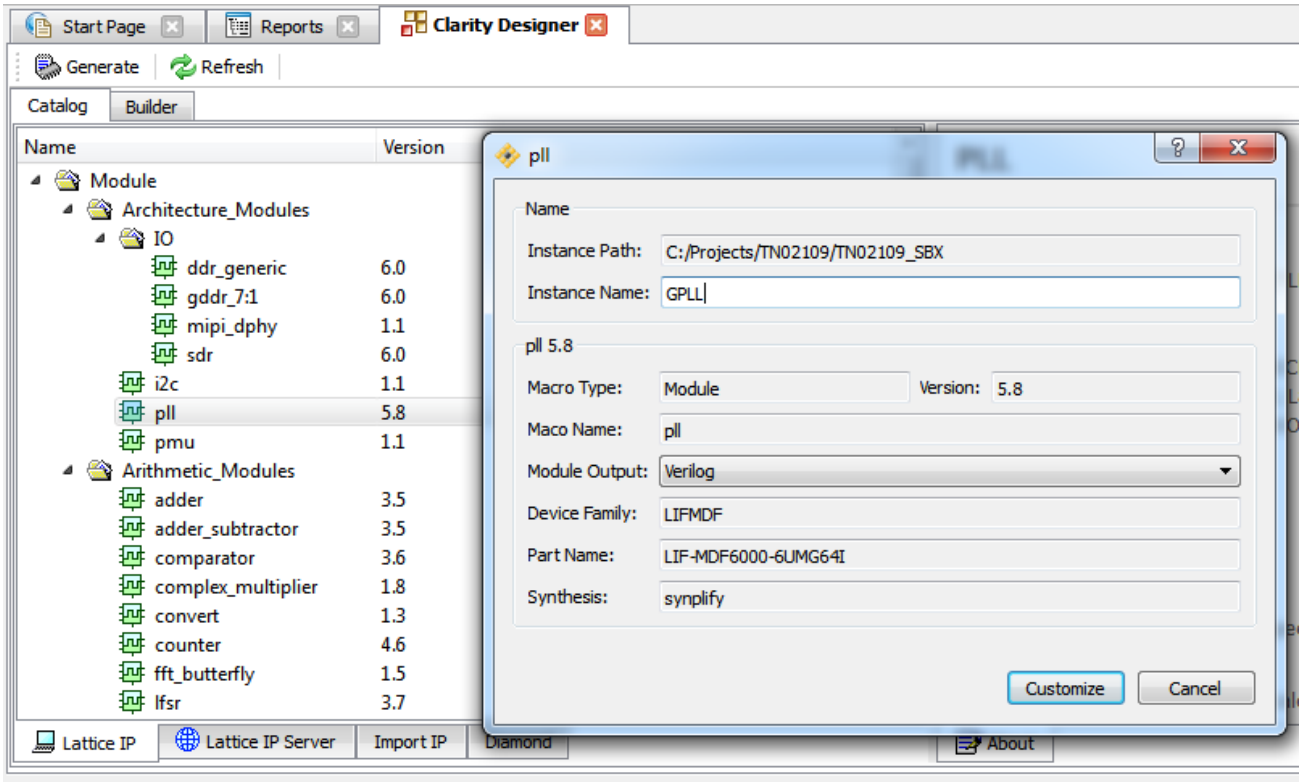

Figure 6.8. Clarity Designer Main Window for PLL Module

6.6.1. Configuration Tab

The Configuration tab (Figure 6.9) lists all accessible attributes with default values set. Upon completion, clicking **Configure** generates the PLL module for the design.

6.6.2. PLL Frequency and Phase Configuration

Enter the input and output clock frequencies, the software calculates the divider settings. After the input and output frequencies are entered, click the **Calculate** button to display the divider values and the closest achievable frequency in the **Actual Frequency** text box. If an entered value is out of range, it is displayed in red, and an error message appears. You can also select a tolerance value from the "Tolerance %" drop-down box. When the **Calculate** button is clicked, the calculation is considered accurate, if the result is within the entered tolerance percentage range.

In the PLL GUI, you enter the desired phase shift in the **Phase** tab (Figure 6.10). The software calculates the closest achievable shift. After the desired phase is entered, click the **Calculate** button to display 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 appears. Refer to Table 6.8 for more details on PLL phase settings.

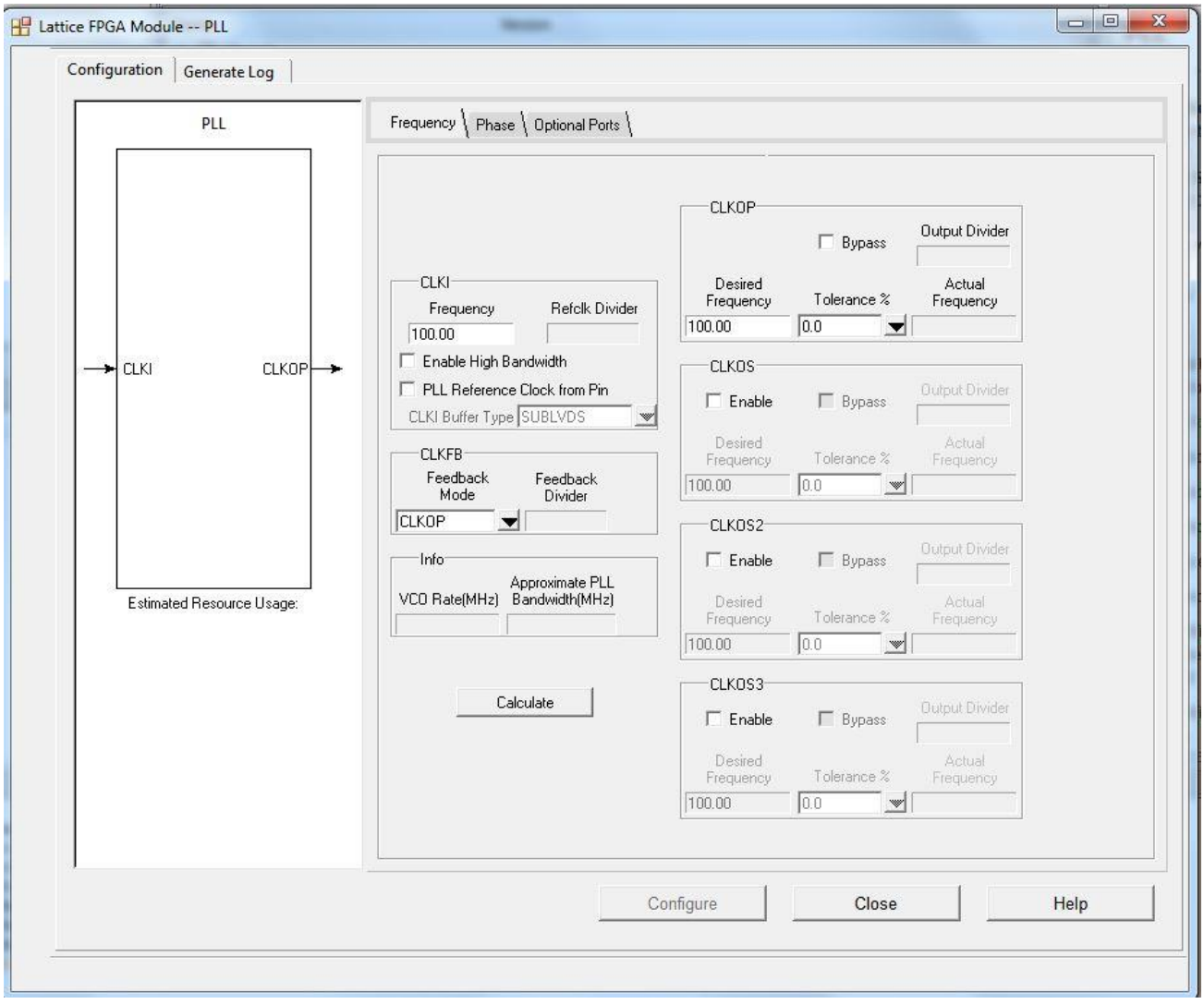

Figure 6.9. CrossLinkPlus PLL Frequency Configuration Tab

Table 6.7. Page 1, PLL Frequency Settings, Clarity Designer GUI

User Parameters	Description	Range	Default	Corresponding HDL Attribute
CLKI	Frequency Input	10 MHz — 400 MHz	100 MHz	FREQUENCY_PIN_CLKI
Refclk Divider – Read Only	Shows the reference clock divider value	—	—	CLKI_DIV
Enable High Bandwidth	Sets the PLL to high bandwidth mode	ON/OFF	OFF	—
PLL Reference Clock from Pin	Sets CLKI Buffer Type	LVDS, SUBLVDS, SLVS, MIPI, LVTTL33, LVCMOS33, LVCMOS25, LVCMOS25D, LVCMOS18, LVCMOS12	LVDS	buf_CLKI
CLKFB	Feedback mode	CLKOP, CLKOS, CLKOS2, CLKOS3, INT_OP, INT_OS, INT_OS2, INT_OS3, UserClock	CLKOP	FEEDBK_PATH
	Feedback Divider (read only)	1 — 128	1	CLKFB_DIV
CLKOP	Enable	ON	OFF	CLKOP_ENABLE
	Bypass	ON	OFF	OUTDIVIDER_MUXA
	Output Divider (read only)	—	—	CLKOP_DIV
	Desired Frequency*	4.6875 MHz — 600 MHz	100 MHz	FREQUENCY_PIN_CLKOP
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	—
	Actual Frequency (read only)	—	—	—
CLKOS	Enable	ON/OFF	ON/OFF	CLKOS_Enable
	Bypass	ON/OFF	ON/OFF	OUTDIVIDER_MUXB
	Clock Divider (read only)	—	—	CLKOS_DIV
	Desired Frequency*	3.125 MHz — 400 MHz	100 MHz	FREQUENCY_PIN_CLKOS
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	—
	Actual Frequency (read only)	—	—	—
CLKOS2	Enable	ON/OFF	ON/OFF	CLKOS2_Enable
	Bypass	ON/OFF	ON/OFF	OUTDIVIDER_MUXC
	Clock Divider (read only)	—	—	CLKOS2_DIV
	Desired Frequency*	3.125 MHz — 400 MHz	100 MHz	FREQUENCY_PIN_CLKOS2
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	—
	Actual Frequency (read only)	—	—	—
CLKOS3	Enable	ON/OFF	ON/OFF	CLKOS3_Enable
	Bypass	ON/OFF	ON/OFF	OUTDIVIDER_MUXD
	Clock Divider (read only)	—	—	CLKOS3_DIV
	Desired Frequency*	3.125 MHz — 400 MHz	100 MHz	FREQUENCY_PIN_CLKOS3
	Tolerance (%)	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0	—
	Actual Frequency (read only)	—	—	—

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

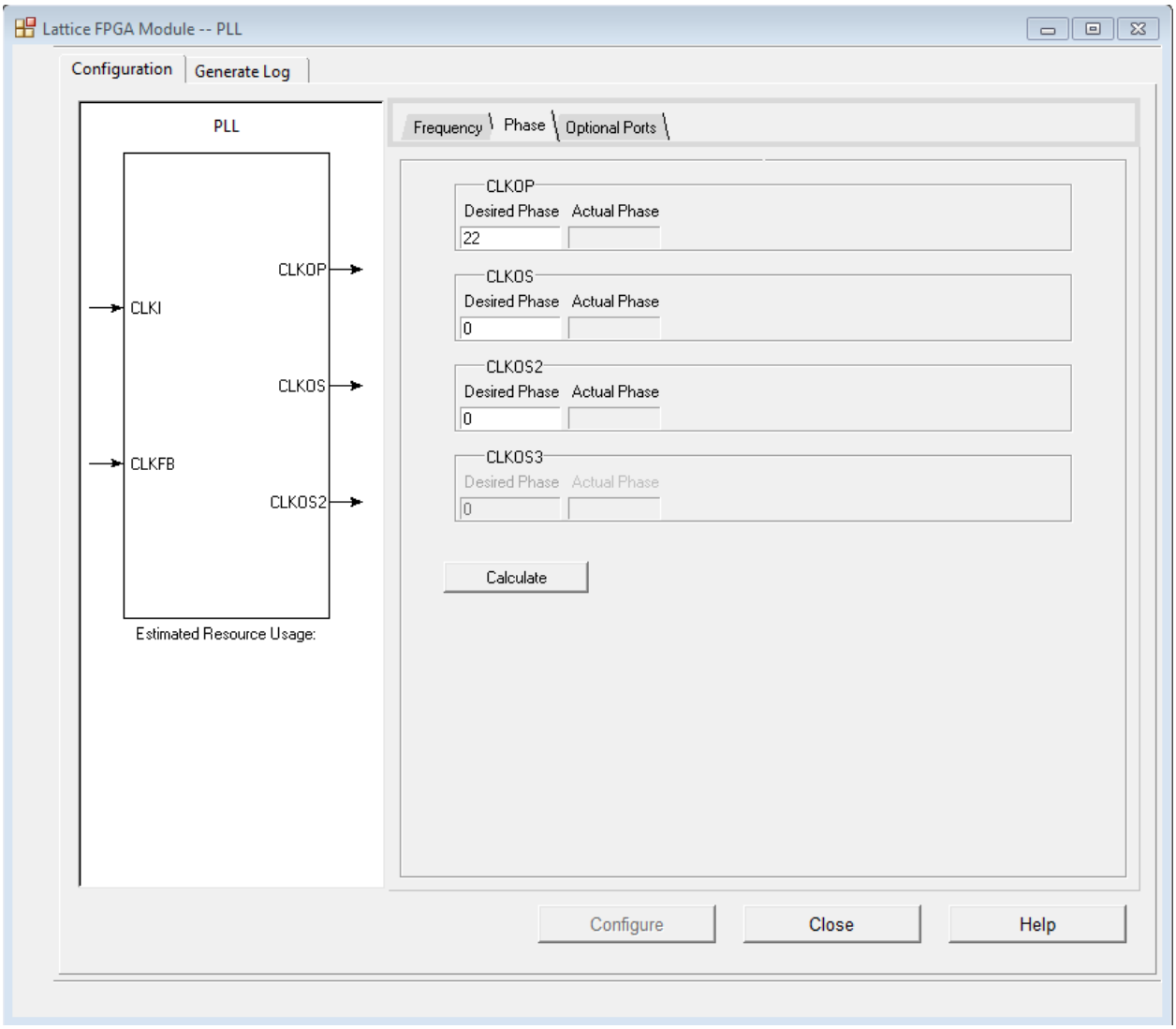

Figure 6.10. CrossLinkPlus PLL Phase Configuration Tab

Table 6.8. Tab 2, PLL Phase Settings, Clarity Designer GUI

User Parameters	Description	Range	Default	Corresponding HDL Attribute
CLKOP	Desired Phase*	Based on Frequency	100 MHz	CLKOP_CPHASE, CLKOP_FPHASE
	Actual Phase (read only)	—	—	—
CLKOS	Desired Phase*	Based on Frequency	100 MHz	CLKOS_CPHASE, CLKOS_FPHASE
	Actual Phase (read only)	—	—	—
CLKOS2	Desired Phase*	Based on Frequency	100 MHz	CLKOS2_CPHASE, CLKOS2_FPHASE
	Actual Phase (read only)	—	—	—
CLKOS3	Desired Phase*	Based on Frequency	100 MHz	CLKOS3_CPHASE, CLKOS3_FPHASE
	Actual Phase (read only)	—	—	—

***Note:** Phase is now a calculated value based on frequency parameters, which provides finer phase resolution.

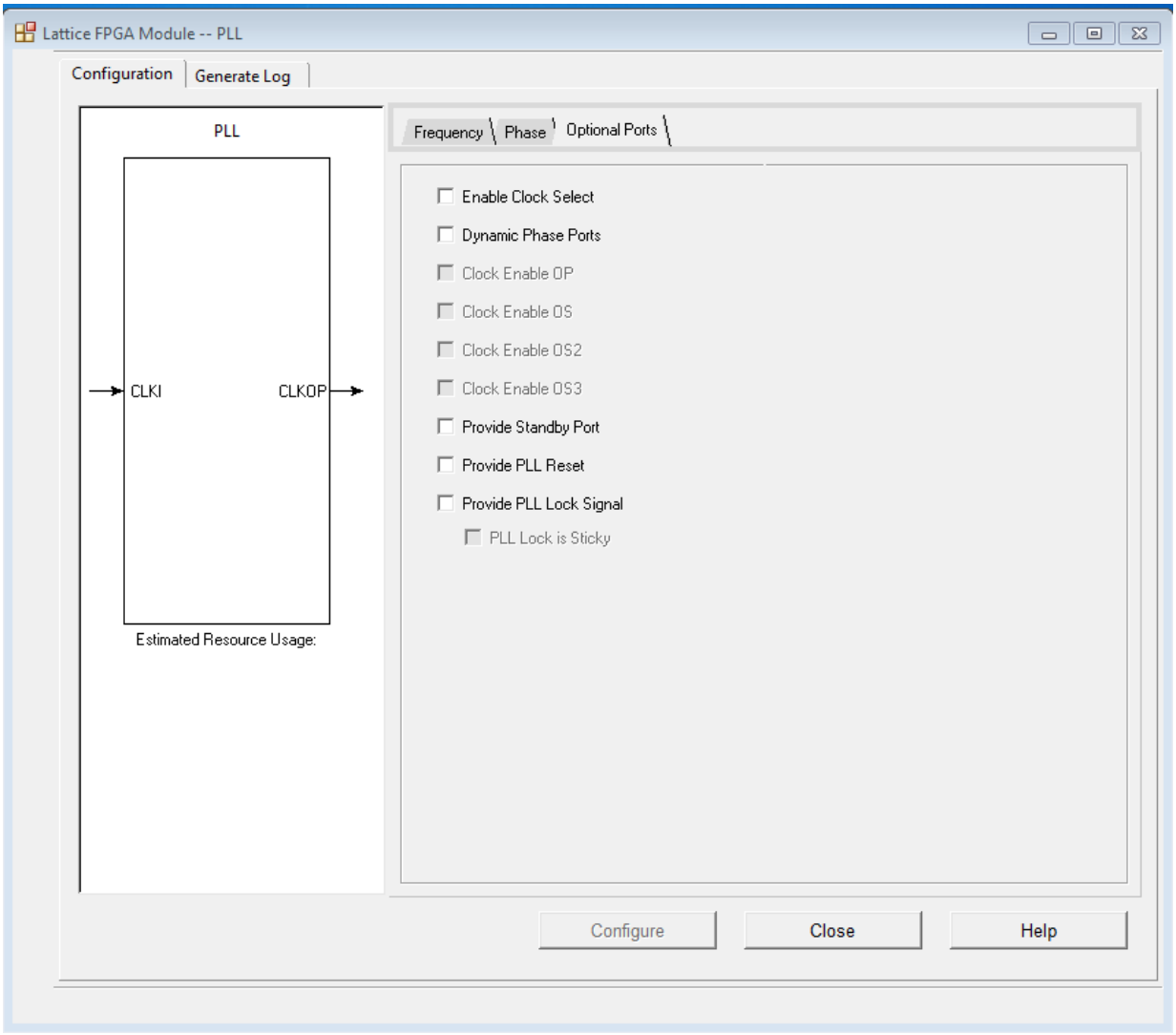

Figure 6.11. CrossLinkPlus PLL Optional Ports Configuration Tab

Table 6.9. Tab 2, PLL Optional Ports, Clarity Designer GUI

User Parameters	Description	Range	Default	Corresponding HDL Attribute
Enable Clock Select	Enables the input clock mux (PLLREFCS component).	ON/OFF	OFF	—
Dynamic Phase Ports	Provides Dynamic Phase Shift ports.	ON/OFF	OFF	DPHASE_SOURCE
Clock Enable OP	Provides ENCLKOP; clock enable port for dynamic clock output shutoff.	ON/OFF	OFF	—
Clock Enable OS	Provides ENCLKOS; clock enable port for dynamic clock output shutoff.	ON/OFF	OFF	—
Clock Enable OS2	Provides ENCLKOS2; clock enable port for dynamic clock output shutoff.	ON/OFF	OFF	—
Clock Enable OS3	Provides ENCLKOS3; clock enable port for dynamic clock output shutoff.	ON/OFF	OFF	—
Provide Standby Port	Provides USRSTDBTY signal.	ON/OFF	OFF	—
Provide PLL Reset	Provides RST signal.	ON/OFF	OFF	PLLRST_ENA
Provide PLL Lock Signal	Provides the LOCK signal.	ON/OFF	OFF	—
PLL Lock is Sticky	When LOCK goes high it does not deassert unless the PLL is reset.	ON/OFF	OFF	PLL_LOCK_MODE

For the PLL, Clarity Designer sets attributes in the HDL module that are specific to the data rate selected. These attributes can be easily changed. However, they should only be modified by re-running the GUI so that the performance of the PLL is maintained. After the MAP stage in the design flow, FREQUENCY preferences are included in the preference file to automatically constrain the clocks produced by the PLL. For a step by step guide on using the Clarity Designer, refer to the Clarity Designer User Manual.

7. Primary Clocks

7.1. 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 pins on LVDS and GPIO Banks
- PLL outputs
- CLKDIV outputs
- Internal FPGA fabric entries (with minimum general routing)
- OSCI clocks
- MIPI DPHY Byte RX and TX clocks from each DPHY block

Prior to primary clock routing, all potential primary clock sources are multiplexed by 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 basic clocking structure is shown in Figure 3.1.

7.2. 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 in the centermux and distributed on the primary clock routing to clock the synchronous elements in the FPGA fabric.

7.3. Dedicated Clock Inputs

The CrossLinkPlus device has dedicated PCLK pins that allow an external clock source to be routed into the FPGA and be used as the FPGA primary clock. These inputs route directly to the primary clock network, or to edge clock routing resources. A dedicated PCLK 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, the input clock should use one of the dedicated PCLK pins.

8. Internal Oscillator (OSCI)

The OSCI element performs multiple functions on the CrossLinkPlus device. Optionally, it is used in user mode for configuration as well. In user mode, the OSCI element has the following features:

- The OSCI permits a design to be fully self-clocked, as long as the accuracy of the OSCI element silicon-based oscillator characteristics are adequate.
- The OSCI allows you to generate a high frequency clock as well as a low frequency clock at the same time.
- Low frequency clock output, LFCLKOUT, is always enabled with 10 kHz output frequency.
- High frequency clock output, HFCLKOUT, can be enabled or disabled using HFOUTEN input. It can also be selected from 48 MHz, 24 MHz, 12 MHz, and 6 MHz depending on HFCLKDIV parameter.
- Both output clocks have a direct connection to primary clock routing.
- When OSCI is instantiated, LFCLKOUT is always ON, and there is no user control. HFCLKOUT is enable and disable through HFOUTEN.
- When OSCI is not instantiated, LFCLKOUT is always ON and cannot be disabled. HFCLKOUT is disabled.

8.1. OSCI Component Definition

The OSCI component can be instantiated in the source code of a design as defined in this section. Figure 8.1 and Table 8.1 below show the OSCI definitions.

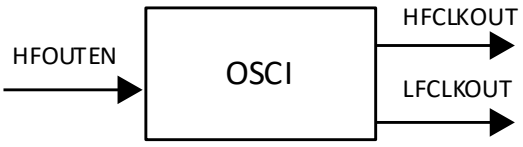

Figure 8.1. OSCI Component Symbol

Table 8.1. OSCI Component Port Definition

Port Name	I/O	Description	Notes
HFOUTEN	I	High frequency clock output enable	0 = disable; 1 = enable; default = 0
HFCLKOUT	O	High frequency clock output	—
LFCLKOUT	O	Low frequency clock output	—

Table 8.2. OSCI Component Attribute Definition

Defparam Name	Description	Value	Default
HFCLKDIV	Configure HF oscillator output divider	1, 2, 4, 8	1

8.2. OSCI Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
component OSCI
Generic (HFCLKDIV   : Integer);
Port  (
      HFOUTEN  : in STD_LOGIC;
      HFCLKOUT : out STD_LOGIC;
      LFCLKOUT : out STD_LOGIC);
end component;
attribute HFCLKDIV : Integer;
attribute HFCLKDIV of I1 : label is 1; -- 1,2,4,8
```

OSCI Instantiation

```
I1: OSCI
generic map (HFCLKDIV => 1)
port map
      HFOUTEN  => HFOUTEN,

      HFCLKOUT => HFCLKOUT,
      LFCLKOUT => LFCLKOUT);
```

8.3. OSCI Usage in Verilog

Module

```
module OSCI (HFOUTEN, HFCLKOUT,
LFCLKOUT);
parameter HFCLKDIV = 1;
input HFOUTEN;

output HFCLKOUT;
output LFCLKOUT;

endmodule
```

OSCI Instantiation

```
defparam I1.HFCLKDIV = 1; // 1,2,4,8
OSCI I1 (

.HFOUTEN(HFOUTEN),
.HFCLKOUT(HFCLKOUT),
.LFCLKOUT(LFCLKOUT));
```

9. Edge Clocks

The CrossLinkPlus device has Edge Clock (ECLK) resources at the bottom two banks, Bank 1 and Bank 2, 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.

The sources of edge clocks are:

- Dedicated Clock (PCLK) pins muxed with the DLLDEL output
- PLL outputs, CLKOP, and CLKOS
- Internal nodes

The ECLK system consists of:

- ECLK in MUX
- ECLKSYNC module to perform clock resynchronize function
- ECLK cascading MUX to widen the ECLK network to span two LVDS I/O banks
- Balanced ECLK tree to ensure the maximum clock skew of the ECLK tree which drives the PIC modules
- Clock Divider (CLKDIV)

to generate a frequency divided down clock which drives the PCLK bottom MIDMUX.

ECLK Input MUX collects all clock sources available as shown in Figure 9.1 below. There are two ECLK Input MUXs, one on each bank. It drives the ECLK SYNC module and the ECLK Clock Divider through a 2-to-1 MUX.

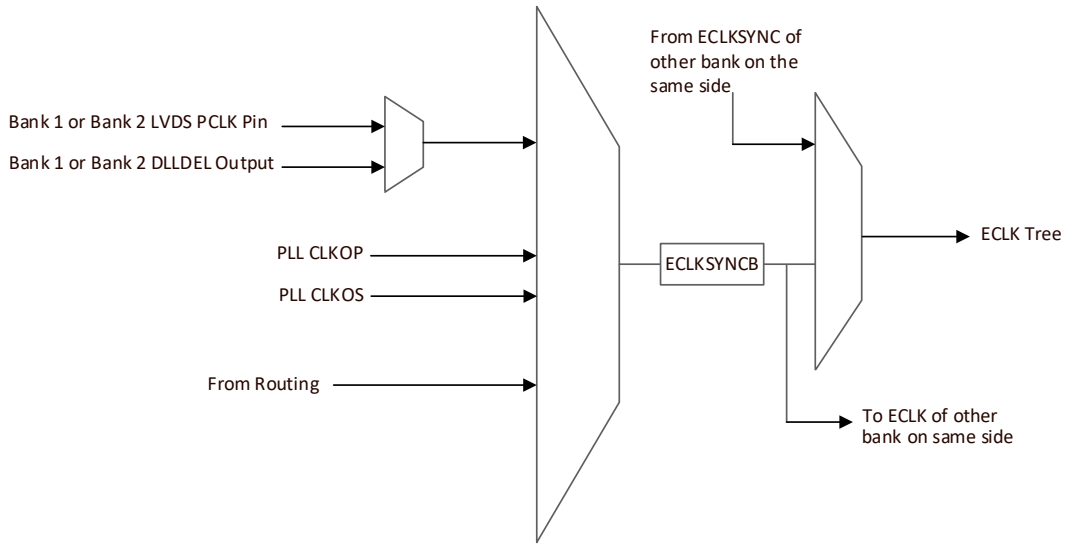

Figure 9.1. Edge Clock Sources per Bank

9.1. Edge Clock Dividers (CLKDIVG)

There are four edge clock dividers available in the CrossLinkPlus device, two per Bank, one per ECLK. The clock divider provides a single divided output with available divide values of 2, 3.5, 4, 7, or 8. The inputs to the clock dividers are the edge clocks, PLL outputs, primary clock input pins, and clocks internally from the fabric. The outputs of the clock divider drive the primary clock network and are mainly used for DDR I/O domain crossing.

9.2. CLKDIVG Component Definition

The CLKDIVG 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 define the CLKDIVG component. Verilog and VHDL instantiations are included.

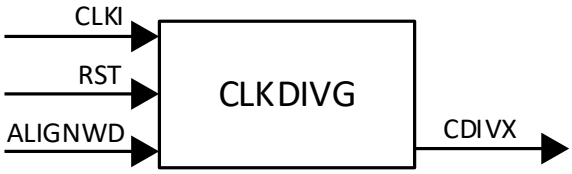

Figure 9.2. CLKDIVG Component Symbol

Table 9.1. CLKDIVG Component Port Definition

Port Name	I/O	Description
CLKI	I	Clock Input.
RST	I	Reset input – Active High, Asynchronously forces all outputs low. • RST = 0 Clock outputs are active • RST = 1 Clock outputs are OFF
ALIGNWD	I	Signal is used for word alignment. When enabled it slips the output one cycle relative to the input clock.
CDIVX	O	Divide by 2, 3.5, 4, 7, or 8 Output Port

Table 9.2. CLKDIVG Component Attribute Definition

Defparam Name	Description	Value	Default
GSR	Enable or Disable Global Reset Signal to Component	ENABLED, DISABLED	DISABLED
DIV	CLK Divider Value	2, 3.5, 4, 7, or 8	2.0

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

9.3. CLKDIVG Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
component CLKDIVG
Generic (DIV      : string;
         GSR      : string);
Port    (RST      : in STD_LOGIC;
         CLKI     : in STD_LOGIC;
         ALIGNWD  : in STD_LOGIC;
         CDIVX    : out STD_LOGIC);
end component;
attribute DIV : string;
attribute DIV of I1 : label is "2.0";---"2.0","3.5","4.0","7.0","8.0"

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

CLKDIVG Instantiation

```
I1: CLKDIVG
generic map (DIV     => "2.0",
             GSR     => "DISABLED")
port map    (RST     => RST,
             CLKI    => CLKI,
             ALIGNWD => ALIGNWD,
             CDIVX   => CDIVX);
```

9.4. CLKDIVG Usage in Verilog HDL

Module

```
module CLKDIVG (RST, CLKI, ALIGNWD, CDIVX);

parameter DIV = "2.0";      // "2.0", "3.5","4.0","7.0","8.0"
parameter GSR = "DISABLED"; // "ENABLED", "DISABLED"

input  RST, CLKI, ALIGNWD;
output CDIVX;
endmodule
```

CLKDIVG Instantiation

```
defparam I1.DIV = "2.0";
defparam I1.GSR = "DISABLED";
CLKDIVG I1 (
   .RST     (RST),
   .CLKI    (CLKI),
   .ALIGNWD (ALIGNWD),
   .CDIVX   (CDIVX));
```

10. Edge Clock Synchronization (ECLKSYNCB)

CrossLinkPlus devices have dynamic edge clock synchronization control (ECLKSYNCB) 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 you to save power by dynamically disabling the clock and associated logic when not needed. This component is used in applications such as high-speed DDR and 7:1 LVDS.

10.1. ECLKSYNCB Component Definition

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

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 (4th) rising edge of the input ECLKI clock.

Figure 10.1 and Table 10.1 show the ECLKSYNCB component definition.

Figure 10.1. ECLKSYNCB Component Symbol

Table 10.1. ECLKSYNCB Component Port Definition

Port Name	I/O	Description
ECLKI	I	Clock Input Port
STOP	I	Select Port • STOP = 0 Clock is Active • STOP = 1 Clock is OFF
ECLKO	O	Clock Output Port

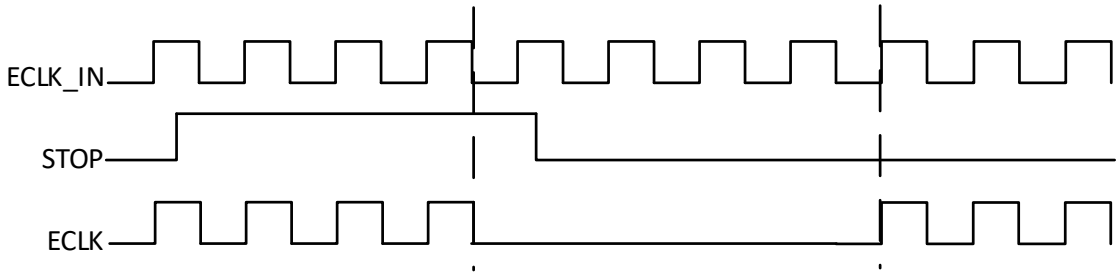

Figure 10.2. ECLKSYNCB Functional Waveform

10.2. ECLKSYNCB Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
COMPONENT ECLKSYNCB
PORT    (ECLKI  :IN STD_LOGIC;
  STOP   :IN  STD_LOGIC;
  ECLKO  :OUT STD_LOGIC);
END COMPONENT;
```

ECLKSYNCB Instantiation

```
I1: ECLKSYNCB
port map (
   ECLKI  => ECLKI,
   STOP   => STOP,
   ECLKO  => ECLKO);
```

10.3. ECLKSYNCB Usage in Verilog HDL

Module

```
module ECLKSYNCB (ECLKI,STOP,ECLKO);
input   ECLKI;
input   STOP;
output  ECLKO;
endmodule
```

ECLKSYNCB Instantiation

```
ECLKSYNCB ECLKSYNCInst0 (
.ECLKI (ECLKI),
.STOP (STOP),
.ECLKO (ECLKO));
```

11. General Routing for Clocks

The CrossLinkPlus 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 (Figure 11.2). If the software cannot place the clock gating logic close enough to a primary clock entry point, an error 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 CrossLinkPlus device fabric (Figure 11.1). In this case, it is recommended to add a preference for this gated clock to use primary routing.

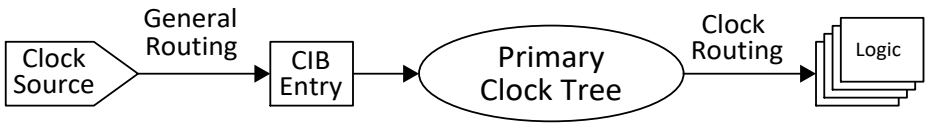

Figure 11.1. Gated Clock to the Primary Clock Routing

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* as well as specify a *PROHIBIT PRIMARY* on the generated clock. See Diamond Help > Constraints Reference Guide > Preferences > UGROUP for more details on UGROUP. If the software cannot place the logic tree within the BBOX, an error message occur.

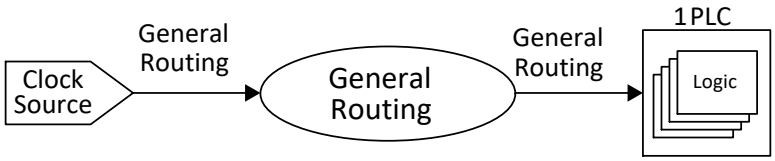

Figure 11.2. Gated Clock to Small Logic Domain

12. General Routing PCLK Pins

Some dedicated pins (GR_PCLK) have short general routing routes onto the primary clock network. There is one pair in each of the LVDS banks of the device. These pins can be used when you run out of PCLK pins. Note that for any DDR interface, it is still required to use dedicated clock pins and clock trees.

13. PLL Reference Clock Switch Primitive (PLLREFCS)

The CrossLinkPlus PLL contains an input mux to dynamically switch between two input reference clocks. This mux is modeled by the PLLREFCS component. This mux may allow glitches and runt pulses through depending on when the clock is switched. It is expected that the input clocks have the same frequency. Table 13.1 defines the I/O ports of the PLLREFCS block.

This component is instantiated in the PLL wrapper when the **Enable Clock Select** option is checked in the **Optional Ports** tab of the Clarity Designer GUI. It can also be directly instantiated and software automatically assigns it to an unused PLL in bypass mode and route the output to the CLKOP port.

Figure 13.1. PLLREFCS Component Symbol

Table 13.1. PLLREFCS Component Port Definition

Port Name	Description
CLK0	CLK0
CLK1	CLK1
SEL	SEL = *0*, CLK0 is selected SEL = *1*, CLK1 is selected
PLLCSOUT	PLLCSOUT

13.1. PLLREFCS Usage in VHDL

Component and Attribute Declaration

```
COMPONENT PLLREFCS
PORT (
  CLK0     : IN STD_LOGIC;
  CLK1     : IN STD_LOGIC;
  SEL      : IN STD_LOGIC;
  PLLCSOUT : OUT STD_LOGIC);
END COMPONENT;
```

PLLREFCS Instantiation

```
PLLREFCSInst0 : PLLREFCS
  PORT MAP (
     CLK0     => CLK_0,
     CLK1     => CLK_1,
     SEL      => SELECT,
     PLLCSOUT => CLK_OUT);
```

13.2. PLLREFCS Usage in Verilog HDL

Module

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

PLLREFCS Instantiation

```
PLLREFCS PLLREFCSInst0 (
   .CLK0 (CLK_0),
   .CLK1 (CLK_1),
   .SEL (SELECT),
   .PLLCSOUT (CLK_OUT));
```

References

For more information, refer to the following documents:

- CrossLinkPlus Family Data Sheet (FPGA-DS-02054)
- CrossLinkPlus High-Speed I/O Interface (FPGA-TN-02102)
- CrossLinkPlus Hardware Checklist (FPGA-TN-02105)
- CrossLinkPlus Programming and Configuration Usage Guide (FPGA-TN-02103)
- CrossLinkPlus sysI/O Usage Guide (FPGA-TN-02108)
- CrossLinkPlus Memory Usage Guide (FPGA-TN-02110)
- Power Management and Calculation for CrossLinkPlus Devices (FPGA-TN-02111)
- CrossLinkPlus I2C Hardened IP Usage Guide (FPGA-TN-02112)
- Advanced CrossLinkPlus I2C Hardened IP Reference Guide (FPGA-TN-02135)

Technical Support Assistance

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

Revision History

Revision 1.0, August 2019

Section	Change Summary
All	Production release.

www.latticesemi.com