CrossLink sysCLOCK PLL/DLL Design and Usage Guide

Preliminary Technical Note

FPGA-TN-02015 Version 1.1

July 2016
Contents

Acronyms in This Document ................................................................................................................. 4
1. Introduction ........................................................................................................................................ 5
2. Clock/Control Distribution Network ............................................................................................... 5
3. CrossLink Top-Level View .............................................................................................................. 5
4. Clocking Architecture Overview .................................................................................................... 6
   4.1. Primary Clock Network ............................................................................................................. 6
   4.2. Edge Clock Network ................................................................................................................ 6
5. Overview of Other Clocking Elements ............................................................................................ 6
   5.1. Edge Clock Dividers (CLKDIVG) .......................................................................................... 6
   5.2. Edge Clock Synchronization Control (ECLKSYNCB) .............................................................. 6
   5.3. Oscillator (OSCI) ...................................................................................................................... 6
6. sysclock PLL ...................................................................................................................................... 7
   6.1. Functional Block Description .................................................................................................. 8
   6.2. PLL Features .......................................................................................................................... 9
   6.3. PLL Inputs and Outputs ......................................................................................................... 10
   6.4. Dynamic Phase Adjustment .................................................................................................. 12
   6.5. Low Power Features .............................................................................................................. 15
   6.6. PLL Usage in Clarity Designer ............................................................................................... 15
7. Primary Clocks ................................................................................................................................... 22
   7.1. Primary Clock Sources ........................................................................................................... 22
   7.2. Primary Clock Routing ........................................................................................................... 22
   7.3. Dedicated Clock Inputs ........................................................................................................... 22
8. Internal Oscillator (OSCI) .................................................................................................................. 23
   8.1. OSCI Component Definition .................................................................................................. 23
   8.2. OSCI Usage in VHDL ............................................................................................................. 24
   8.3. OSCI Usage in Verilog .......................................................................................................... 24
9. Edge Clocks .......................................................................................................................................... 25
   9.1. Edge Clock Dividers (CLKDIVG) ........................................................................................ 25
   9.2. CLKDIVG Component Definition ......................................................................................... 26
   9.3. CLKDIVG Usage in VHDL .................................................................................................... 27
   9.4. CLKDIVG Usage in Verilog .................................................................................................. 27
10. Edge Clock Synchronization (ECLKSYNCB) .................................................................................. 28
    10.1. ECLKSYNCB Component Definition ................................................................................ 28
    10.2. ECLKSYNCB Usage in VHDL ........................................................................................... 29
    10.3. ECLKSYNCB Usage in Verilog ........................................................................................ 29
11. General Routing for Clocks ............................................................................................................. 30
12. General Routing PCLK Pins ............................................................................................................ 30
13. PLL Reference Clock Switch Primitive (PLLREFCS) .................................................................... 31
    13.1. PLLREFCS Usage in VHDL ............................................................................................... 31
    13.2. PLLREFCS Usage in Verilog ............................................................................................. 31

References ............................................................................................................................................... 33

Technical Support Assistance ................................................................................................................... 33
Revision History ....................................................................................................................................... 33

© 2015-2016 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.
Figures
Figure 3.1. CrossLink Clocking Structure ................................................................. 5
Figure 6.1. CrossLink PLL Block Diagram ................................................................. 7
Figure 6.2. PLL Component Instance ....................................................................... 7
Figure 6.3. PLL Dedicated Inputs to the PLLREFCS Component .............................. 9
Figure 6.4. RST Input Timing Diagram ................................................................. 10
Figure 6.5. PLL Phase Shifting Using the PHASESTEP Signal ................................. 13
Figure 6.6. Divider Phase Shift Timing Diagram .................................................... 14
Figure 6.7. Dynamic Clock Enable for PLL Outputs ............................................... 15
Figure 6.8. Clarity Designer Main Window for PLL Module .................................... 16
Figure 6.9. CrossLink PLL Frequency Configuration Tab ....................................... 17
Figure 6.10. CrossLink PLL Phase Configuration Tab ............................................ 19
Figure 6.11. CrossLink PLL Optional Ports Configuration Tab ............................... 20
Figure 8.1. OSCI Component Symbol ................................................................. 23
Figure 9.1. Edge Clock Sources Per Bank .......................................................... 25
Figure 9.2. CLKDIVG Component Symbol ......................................................... 26
Figure 10.1. ECLKSYNCB Component Symbol ..................................................... 28
Figure 10.2. ECLKSYNCB Functional Waveform ............................................... 28
Figure 11.1. Gated Clock to the Primary Clock Routing ....................................... 30
Figure 11.2. Gated Clock to Small Logic Domain ................................................... 30
Figure 13.1. PLLREFCS Component Symbol ....................................................... 31

Tables
Table 1.1. Key Clocking Resources ........................................................................ 5
Table 6.1. EHXPLL Component Port Definition .................................................. 8
Table 6.2. PLL Clock Output Enable Signal List ................................................. 10
Table 6.3. PHASESEL Signal Settings Definition .............................................. 11
Table 6.4. PHASEDIR Signal Settings Definition ............................................... 11
Table 6.5. PLL Clock Outputs and ECLK Connectivity ....................................... 12
Table 6.6. Page 1, PLL Frequency Settings, Clarity Designer GUI ....................... 18
Table 6.7. Tab 2, PLL Phase Settings, Clarity Designer GUI .................................. 19
Table 6.8. Tab 2, PLL Optional Ports, Clarity Designer GUI ................................. 20
Table 8.1. OSCI Component Port Definition ....................................................... 23
Table 8.2. OSCI Component Attribute Definition .............................................. 23
Table 9.1. CLKDIVG Component Port Definition .............................................. 26
Table 9.2. CLKDIVG Component Attribute Definition ....................................... 26
Table 10.1. ECLKSYNCB Component Port Definition ........................................ 28
Table 13.1. PLLREFCS Component Port Definition ........................................... 31
## Acronyms in This Document

A list of acronyms used in this document.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKDIVG</td>
<td>Edge Clock Dividers</td>
</tr>
<tr>
<td>DDR</td>
<td>Double Data Rate</td>
</tr>
<tr>
<td>ECLKSYNCB</td>
<td>Edge Clock Stop</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input/Output</td>
</tr>
<tr>
<td>LVDS</td>
<td>Low-Voltage Differential Signaling</td>
</tr>
<tr>
<td>MIPI</td>
<td>Mobile Industry Processor Interface</td>
</tr>
<tr>
<td>OSCI</td>
<td>Oscillator</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase Locked Loop</td>
</tr>
<tr>
<td>VCO</td>
<td>Voltage-Controlled Oscillator</td>
</tr>
</tbody>
</table>
1. Introduction

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of PLLs</td>
<td>General purpose PLLs.</td>
<td>1</td>
</tr>
<tr>
<td>Number of Edge Clocks</td>
<td>Edge Clocks for high speed applications.</td>
<td>4</td>
</tr>
<tr>
<td>Number of Clock Dividers</td>
<td>Edge Clock Dividers for DDR applications.</td>
<td>4</td>
</tr>
<tr>
<td>Number of DDRDLLs</td>
<td>DDRDLL used for High Speed IO interfaces.</td>
<td>2</td>
</tr>
</tbody>
</table>

It is important to note that the user needs to validate their pinout so that correct pin placement is used. The Lattice Diamond® tools should be used to validate the pinout while designing the printed circuit board.

2. Clock/Control Distribution Network

CrossLink 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, high speed edge clocks.

3. CrossLink Top-Level View

A top level view of the major clocking resources for the CrossLink device is shown in Figure 3.1.
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 the user eight available unique clock domains in CrossLink. 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 CrossLink device. There are two edge clocks per bank located on Bank 1 and 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 drives to the primary clock routing to the fabric. There are four clock dividers on the CrossLink device.

5.2. Edge Clock Synchronization Control (ECLKSYNCB)

Each ECLK has a block to allow dynamic start and stop of the edge clock. The user can start and stop the clock synchronous to an event or external signal. These are important for applications requiring exact clock timing relationships on the inputs, such as DDR 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 CrossLink device. The oscillator clock outputs have access to primary clock routing resources. The CrossLink 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 CrossLink PLL provides features such as clock injection delay removal, frequency synthesis, and phase adjustment. Figure 6.1 shows a block diagram of the CrossLink PLL.

![CrossLink PLL Block Diagram](image)

**Figure 6.1. CrossLink PLL Block Diagram**

![PLL Component Instance](image)

**Figure 6.2. PLL Component Instance**
Table 6.1. EHXPLL Component Port Definition

<table>
<thead>
<tr>
<th>Signal</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKI</td>
<td>I</td>
<td>Input clock to PLL.</td>
</tr>
<tr>
<td>CLKFB</td>
<td>I</td>
<td>Feedback clock</td>
</tr>
<tr>
<td>USRSTDBY</td>
<td>I</td>
<td>User port to put the PLL in sleep mode.</td>
</tr>
<tr>
<td>PHASESEL[1:0]</td>
<td>I</td>
<td>Select the output affected by Dynamic Phase adjustment.</td>
</tr>
<tr>
<td>PHASEDIR</td>
<td>I</td>
<td>Dynamic phase adjustment direction.</td>
</tr>
<tr>
<td>PHASESTEP</td>
<td>I</td>
<td>Dynamic phase adjustment step.</td>
</tr>
<tr>
<td>PHASELOADREG</td>
<td>I</td>
<td>Load dynamic phase adjustment values into PLL.</td>
</tr>
<tr>
<td>RST</td>
<td>I</td>
<td>Resets the whole PLL.</td>
</tr>
<tr>
<td>ENCLKOP</td>
<td>I</td>
<td>Enable PLL output CLKOP.</td>
</tr>
<tr>
<td>ENCLKOS</td>
<td>I</td>
<td>Enable PLL output CLKOS.</td>
</tr>
<tr>
<td>ENCLKOS2</td>
<td>I</td>
<td>Enable PLL output CLKOS2.</td>
</tr>
<tr>
<td>ENCLKOS3</td>
<td>I</td>
<td>Enable PLL output CLKOS3.</td>
</tr>
<tr>
<td>PLLWAKESYNC</td>
<td>I</td>
<td>Enable PLL switching from internal feedback to user feedback path when PLL wake up.</td>
</tr>
<tr>
<td>CLKOP</td>
<td>O</td>
<td>PLL main output clock</td>
</tr>
<tr>
<td>CLKOS</td>
<td>O</td>
<td>PLL output clock</td>
</tr>
<tr>
<td>CLKOS2</td>
<td>O</td>
<td>PLL output clock</td>
</tr>
<tr>
<td>CLKOS3</td>
<td>O</td>
<td>PLL output clock</td>
</tr>
<tr>
<td>LOCK</td>
<td>O</td>
<td>PLL LOCK to CLKI, Asynchronous signal. Active high indicates PLL lock.</td>
</tr>
</tbody>
</table>

6.1. Functional Block Description

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.

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 to user interface when “user clock” option is selected for feedback clock, otherwise this port will be connected by the tool to appropriate signal as selected by the user in the software.

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.

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.
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 on page 12 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 would apply on other clocks.

6.2. PLL Features

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.

PLL Input Clock Mux (PLLREFCS)
CrossLink PLL contains an input mux to dynamically switch between two input reference clocks. The output of the PLLREFCS is routed directly into the PLL. In order to enhance the clock muxing capability of the CrossLink 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](image)

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.

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. Users 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.

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 to PLL to ensure the clock phase alignment or canceling the 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 loss due to this feedback and the whole ECLK tree insertion delay is compensated.
6.3. PLL Inputs and Outputs

CLKI Input
The CLKI signal is the reference clock for the PLL. It must conform to the specifications in the data sheet in order for the PLL to operate correctly. The CLKI signal can come from a dedicated 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.

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 will be divided by the feedback (N) divider. A bypassed PLL output cannot be used as the feedback signal.

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.

```
RST   Trst   Trstrec
CLKI
CLKOP/OS/OS2/OS3
```

**Figure 6.4. RST Input Timing Diagram**

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 6.2. PLL Clock Output Enable Signal List**

<table>
<thead>
<tr>
<th>Clock Enable Signal Name</th>
<th>Corresponding PLL Output</th>
<th>Clarity Designer Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCLKOP</td>
<td>CLKOP</td>
<td>Clock Enable OP</td>
</tr>
<tr>
<td>ENCLKOS</td>
<td>CLKOS</td>
<td>Clock Enable OS</td>
</tr>
<tr>
<td>ENCLKOS2</td>
<td>CLKOS2</td>
<td>Clock Enable OS2</td>
</tr>
<tr>
<td>ENCLKOS3</td>
<td>CLKOS3</td>
<td>Clock Enable OS3</td>
</tr>
</tbody>
</table>

This allows the user to save power by stopping the corresponding output clock when not in use. The clock enable signals are optional and are available only when the user selects the corresponding option in Clarity Designer. If a clock enable signal is not requested, its corresponding output will be active at all times when the PLL is instantiated unless the PLL is placed into standby mode. The user cannot access a clock enable signal in Clarity Designer when using it for external feedback to avoid shutting off the feedback clock input.
Dynamic Phase Shift Inputs
The PLL has five ports to allow for dynamic phase adjustment from FPGA logic. The Dynamic Phase Adjustment section on page 12 provides details on how the user should 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 on page 12. The PHASESEL signal must be stable for 1 ns before the PHASESTEP signal is pulsed. The PHASESEL signal is optional and will be available if the user selects the “Dynamic Phase Ports” option in Clarity Designer.

Table 6.3. PHASESEL Signal Settings Definition

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

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 will be available if the user selects the Dynamic Phase ports option in Clarity Designer.

Table 6.4. PHASEDIR Signal Settings Definition

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

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 will be available if the user has selected the Dynamic Phase ports option 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 will be available if the user has selected the Dynamic Phase ports option in Clarity Designer.

PLL Clock Outputs
The PLL has four outputs, listed in Table 6.5. 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. The outputs can come from their output divider or the reference clock input (PLL bypass).
Table 6.5. PLL Clock Outputs and ECLK Connectivity

<table>
<thead>
<tr>
<th>Clock Output Name</th>
<th>Edge Clock Connectivity</th>
<th>Selectable Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKOP</td>
<td>Bank 0, Bank 1 ECLKs</td>
<td>Always Enabled</td>
</tr>
<tr>
<td>CLKOS</td>
<td>Bank 0, Bank 1 ECLKs</td>
<td>Selectable via Clarity Designer</td>
</tr>
<tr>
<td>CLKOS2</td>
<td>No ECLK Connection</td>
<td>Selectable via Clarity Designer</td>
</tr>
<tr>
<td>CLKOS3</td>
<td>No ECLK Connection</td>
<td>Selectable via Clarity Designer</td>
</tr>
</tbody>
</table>

**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 “PLL Lock is Sticky”. In sticky lock mode, when 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 is deasserted (logic ‘0’). 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 will be available if the user selects the Provide PLL Lock signal option in Clarity Designer.

### 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 6.3 above 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 will either be advanced or delayed depending upon the value of the PHASEDIR port or signal. Table 6.4 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.

**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:

\[
(CLKO_{n} \_FPHASE/(8*CLKO_{n} \_DIV))\times360
\]

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 4 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.
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 will get \[\frac{1}{8*2}\]*360 = 22.5 degree phase shift (delayed).

**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:

\[
\frac{\text{CLKO}_n._\text{CPHASE} - \text{CLKO}_n._\text{DIV}}{\text{CLKO}_n._\text{DIV} + 1} \times 360^0
\]

Where \(n\) is the clock output specified by PHASESEL (CLKOP/OS2/OS3). Values for \(\text{CLKO}_n._\text{CPHASE}\) and \(\text{CLKO}_n._\text{DIV}\) are located in the HDL source file. Note that if these values are both “1”, no shift will be made.

---

© 2015-2016 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at [www.latticesemi.com/legal](http://www.latticesemi.com/legal). All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.
*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 CrossLink PLL contains several features that allow the user 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 the user 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’s output is not enabled, its dynamic clock enable port is unavailable.

![Dynamic Clock Enable for PLL Outputs](image)

**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. 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.

When the PLL is selected 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 Customize to open the PLL configuration window.
Configuration Tab
The Configuration tab lists all user accessible attributes with default values set. Upon completion, clicking Configure generates the PLL module for use in the design.

PLL Frequency and Phase Configuration
Enter the input and output clock frequencies and 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. The user 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, the user enters the desired phase shift and 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.
Figure 6.9. CrossLink PLL Frequency Configuration Tab
## Table 6.6. PLL Frequency Settings, Clarity Designer GUI

<table>
<thead>
<tr>
<th>User Parameters</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
<th>Corresponding HDL Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKI</td>
<td>Frequency Input</td>
<td>10 MHz — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKI</td>
</tr>
<tr>
<td></td>
<td>Refclk Divider — Read Only</td>
<td></td>
<td></td>
<td>CLKI_DIV</td>
</tr>
<tr>
<td></td>
<td>Enable High Bandwidth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLL Reference Clock from Pin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLKFB</td>
<td>Feedback mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feedback Divider (read only)</td>
<td>1 — 128</td>
<td>1</td>
<td>CLKFB_DIV</td>
</tr>
<tr>
<td>CLKOP</td>
<td>Enable</td>
<td>OFF</td>
<td></td>
<td>CLKOP_ENABLE</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON</td>
<td></td>
<td>OUTDIVIDER_MUXA</td>
</tr>
<tr>
<td></td>
<td>Output Divider (read only)</td>
<td></td>
<td></td>
<td>CLKOP_DIV</td>
</tr>
<tr>
<td></td>
<td>Desired Frequency*</td>
<td>4.6875 MHz — 600 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOP</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Actual Frequency (read only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLKOS</td>
<td>Enable</td>
<td>ON/OFF</td>
<td>ON/OFF</td>
<td>CLKOS_Enable</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>ON/OFF</td>
<td>OUTDIVIDER_MUXB</td>
</tr>
<tr>
<td></td>
<td>Clock Divider (read only)</td>
<td></td>
<td></td>
<td>CLKOS_DIV</td>
</tr>
<tr>
<td></td>
<td>Desired Frequency*</td>
<td>3.125 MHz — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOS</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Actual Frequency (read only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLKOS2</td>
<td>Enable</td>
<td>ON/OFF</td>
<td>ON/OFF</td>
<td>CLKOS2_Enable</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>ON/OFF</td>
<td>OUTDIVIDER_MUXC</td>
</tr>
<tr>
<td></td>
<td>Clock Divider (read only)</td>
<td></td>
<td></td>
<td>CLKOS2_DIV</td>
</tr>
<tr>
<td></td>
<td>Desired Frequency*</td>
<td>3.125 MHz — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOS2</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Actual Frequency (read only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLKOS3</td>
<td>Enable</td>
<td>ON/OFF</td>
<td>ON/OFF</td>
<td>CLKOS3_Enable</td>
</tr>
<tr>
<td></td>
<td>Bypass</td>
<td>ON/OFF</td>
<td>ON/OFF</td>
<td>OUTDIVIDER_MUXD</td>
</tr>
<tr>
<td></td>
<td>Clock Divider (read only)</td>
<td></td>
<td></td>
<td>CLKOS3_DIV</td>
</tr>
<tr>
<td></td>
<td>Desired Frequency*</td>
<td>3.125 MHz — 400 MHz</td>
<td>100 MHz</td>
<td>FREQUENCY_PIN_CLKOS3</td>
</tr>
<tr>
<td></td>
<td>Tolerance (%)</td>
<td>0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Actual Frequency (read only)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note: If this clock is selected as feedback, the minimum achievable output frequency is 10 MHz.
Figure 6.10. CrossLink PLL Phase Configuration Tab

Table 6.7. Tab 2, PLL Phase Settings, Clarity Designer GUI

<table>
<thead>
<tr>
<th>User Parameters</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
<th>Corresponding HDL Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKOP</td>
<td>Desired Phase*</td>
<td>Based on Frequency</td>
<td>100 MHz</td>
<td>CLKOP_CPHASE, CLKOP_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Actual Phase (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>CLKOS</td>
<td>Desired Phase*</td>
<td>Based on Frequency</td>
<td>100 MHz</td>
<td>CLKOS_CPHASE, CLKOS_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Actual Phase (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>CLKOS2</td>
<td>Desired Phase*</td>
<td>Based on Frequency</td>
<td>100 MHz</td>
<td>CLKOS2_CPHASE, CLKOS2_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Actual Phase (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>CLKOS3</td>
<td>Desired Phase*</td>
<td>Based on Frequency</td>
<td>100 MHz</td>
<td>CLKOS3_CPHASE, CLKOS3_FPHASE</td>
</tr>
<tr>
<td></td>
<td>Actual Phase (read only)</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

*Note: Phase is now a calculated value based on frequency parameters, which provides finer phase resolution.
Figure 6.11. CrossLink PLL Optional Ports Configuration Tab

### Table 6.8. Tab 2, PLL Optional Ports, Clarity Designer GUI

<table>
<thead>
<tr>
<th>User Parameters</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
<th>Corresponding HDL Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Clock Select</td>
<td>Enables the input clock mux (PLLREFCS component).</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>Dynamic Phase Ports</td>
<td>Provides Dynamic Phase Shift ports.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>DPHASE_SOURCE</td>
</tr>
<tr>
<td>Clock Enable OP</td>
<td>Provides ENCLKOP; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>Clock Enable OS</td>
<td>Provides ENCLKOS; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>Clock Enable OS2</td>
<td>Provides ENCLKOS2; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>Clock Enable OS3</td>
<td>Provides ENCLKOS3; clock enable port for dynamic clock output shutoff.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>Provide PLL Reset</td>
<td>Provides RST signal.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>PLLRST_ENA</td>
</tr>
<tr>
<td>Provide PLL Lock Signal</td>
<td>Provides the LOCK signal.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>—</td>
</tr>
<tr>
<td>PLL Lock is Sticky</td>
<td>When LOCK goes high it does not deassert unless the PLL is reset.</td>
<td>ON/OFF</td>
<td>OFF</td>
<td>PLL_LOCK_MODE</td>
</tr>
</tbody>
</table>
For the PLL, Clarity Designer 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 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 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)
- OSC 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 multiplexer 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 on page 5.

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 CrossLink 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 the input clock should use one of the dedicated PCLK pins.
8. Internal Oscillator (OSCI)

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

- It permits a design to be fully self-clocked, as long as the accuracy of the OSCI element’s silicon-based oscillator characteristics are adequate.
- The CrossLink OSCI allows user to generate a high frequency clock as well as a low frequency clock at the same time.
- Low frequency clock output, that is LFCLKOUT, is always enabled with 10 kHz output frequency.
- High frequency clock output, that is 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.

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.

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFOUTEN</td>
<td>I</td>
<td>High frequency clock output enable</td>
</tr>
<tr>
<td>HFCLKOUT</td>
<td>O</td>
<td>High frequency clock output</td>
</tr>
<tr>
<td>LFCLKOUT</td>
<td>O</td>
<td>Low Frequency clock output</td>
</tr>
</tbody>
</table>

Table 8.2. OSCI Component Attribute Definition

<table>
<thead>
<tr>
<th>Defparam Name</th>
<th>Description</th>
<th>Value</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFCLKDIV</td>
<td>Configure HF oscillator output divider</td>
<td>1, 2, 4, 8</td>
<td>1</td>
</tr>
</tbody>
</table>
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

Component and Attribute Declaration
module OSCI (HFOUTEN, HFCLKOUT,
         LFCLKOUT);
parameter HFCLKDIV = 1;
input HFOUTEN;
input 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 CrossLink device has Edge Clock (ECLK) at the bottom 2 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 IO banks, the balanced ECLK tree to ensure the maximum clock skew of the ECLK tree which drives the PIC modules, and 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.

9.1. Edge Clock Dividers (CLKDIVG)

There are four edge clock dividers available in the CrossLink 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 and 8. The inputs to the clock dividers are the edge clocks, PLL outputs, primary clock input pins and internally from 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.

![CLKDIVG Component Symbol](image)

**Figure 9.2. CLKDIVG Component Symbol**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKI</td>
<td>I</td>
<td>Clock Input.</td>
</tr>
</tbody>
</table>
| 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 9.1. CLKDIVG Component Port Definition**

<table>
<thead>
<tr>
<th>Defparam Name</th>
<th>Description</th>
<th>Value</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSR</td>
<td>Enable or Disable Global Reset Signal to Component</td>
<td>ENABLED, DISABLED</td>
<td>DISABLED</td>
</tr>
<tr>
<td>DIV</td>
<td>CLK Divider Value</td>
<td>2, 3.5, 4, 7 or 8</td>
<td>2.0</td>
</tr>
</tbody>
</table>

**Table 9.2. CLKDIVG Component Attribute Definition**

The ALIGNWD input is intended for 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

Component and Attribute Declaration
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)

CrossLink 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 the designer to synchronize the edge clock to an event or external signal if desired. It also allows the design to dynamically disable a clock and its associated logic in the design when it is not needed and thus save power.

10.1. ECLKSYNCB Component Definition

The ECLKSYNCB 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 in order to synchronize the signals derived from ECLK and used in high speed DDR mode applications and 7:1 LVDS.

Control signal STOP is synchronized with ECLK when asserted. When control signal STOP is asserted, the clock output will be 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](image)

![Figure 10.2. ECLKSYNCB Functional Waveform](image)

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECLKI</td>
<td>I</td>
<td>Clock Input Port</td>
</tr>
<tr>
<td>STOP</td>
<td>I</td>
<td>Select Port</td>
</tr>
<tr>
<td>STOP = 0</td>
<td></td>
<td>Clock is Active</td>
</tr>
<tr>
<td>STOP = 1</td>
<td></td>
<td>Clock is OFF</td>
</tr>
<tr>
<td>ECLKO</td>
<td>O</td>
<td>Clock Output Port</td>
</tr>
</tbody>
</table>

Table 10.1. ECLKSYNCB Component Port Definition
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

Component and Attribute Declaration
module ECLKSYNCB (ECLKI,STOP,ECLKO);
   input  ECLKI;
   input  STOP;
   output ECLKO;
endmodule

ECLKSYNCB Instantiation
ECLKSYNCB ECLKSYNCKinst0 (ECLKI (ECLKI),
                          .STOP (STOP),
                          .ECLKO (ECLKO));
11. General Routing for Clocks

The CrossLink 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 will limit 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 then an error will occur:

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

There are multiple entry points to the primary clock routing throughout the CrossLink device fabric. 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](image)

For a very small clock domain, the user can limit the distance of a general routing based (gated) clock to one PLC in distance to the logic it clocks. The user must group this logic (UGROUP) with a BBOX = “1, 1” (see Diamond 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, then an error message will occur.

![Figure 11.2. Gated Clock to Small Logic Domain](image)

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 user runs 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 CrossLink 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 Clarity Designer GUI. It can also be directly instantiated and software will automatically assign it to an unused PLL in bypass mode and route the output to the CLKOP port.

Table 13.1. PLLREFCS Component Port Definition

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK0</td>
<td>CLK0</td>
</tr>
<tr>
<td>CLK1</td>
<td>CLK1</td>
</tr>
<tr>
<td>SEL</td>
<td>SEL = '0’, CLK0 is selected</td>
</tr>
<tr>
<td></td>
<td>SEL = '1’, CLK1 is selected</td>
</tr>
<tr>
<td>PLLCSOUT</td>
<td>PLLCSOUT</td>
</tr>
</tbody>
</table>

13.1. PLLREFCS Usage in VHDL

Component Declaration

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

PLLREFCS Instantiation

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

13.2. PLLREFCS Usage in Verilog

Component and Attribute Declaration

```verilog
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:

- FPGA-DS-02007, CrossLink Family Data Sheet
- FPGA-TN-02012, CrossLink High-Speed I/O Interface
- FPGA-TN-02013, CrossLink Hardware Checklist
- FPGA-TN-02014, CrossLink Programming and Configuration Usage Guide
- FPGA-TN-02016, CrossLink sysI/O Usage Guide
- FPGA-TN-02017, CrossLink Memory Usage Guide
- FPGA-TN-02018, Power Management and Calculation for CrossLink Devices
- FPGA-TN-02019, CrossLink I2C Hardened IP Usage Guide

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

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2016</td>
<td>1.1</td>
<td>1. Updated CrossLink document numbers. The previous document number of this</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Technical Note was TN1304</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Updated Figure 6.2. PLL Component Instance and Figure 6.6. Divider Phase Shift Timing Diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Updated Figure 6.9. CrossLink PLL Frequency Configuration Tab and corresponding Table 6.6. Page 1, PLL Frequency Settings, Clarity Designer GUI, added “PLL Reference Clock from Pin” parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Updated OSCI Usage in VHDL section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. Updated CLKDIVG Usage in VHDL section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. Updated ECLKSYNCC to ECLKSYNCCB in Edge Clock Synchronization (ECLKSYNCCB) section</td>
</tr>
<tr>
<td>May 2016</td>
<td>1.0</td>
<td>First preliminary release.</td>
</tr>
</tbody>
</table>