ECP5 and ECP5-5G sysCLOCK PLL/DLL Design and Usage Guide

November 2015 Technical Note TN1263

Introduction

This usage guide describes the clock resources available in the ECP5™ and ECP5-5G™ device architecture. Details are provided for primary clocks, edge clocks, PLLs, the internal oscillator, and clocking elements such as clock dividers, clock multiplexers, and clock stop blocks available in the ECP5 and ECP5-5G device.

The number of PLLs, Edge clocks, and Clock dividers for each device is listed in Table 1.

Table 1. Number of PLLs, Edge Clocks, and Clock Dividers[1]

Parameter	Description	LFE5-85	LFE5-45	LFE5-25
Number of PLLs	General purpose PLLs.	4	4	2
Number of Edge Clocks	Edge Clocks for high speed applications.	8	8	8
Number of Clock Dividers	Edge Clock Dividers for DDR applications.	4	4	4
Number of PCS Clock Dividers[1]	Clock dividers for domain crossing applications.	2	2	1
Number of DDRDLLs	DDRDLL used to DDR memory and High Speed IO interfaces	4	4	2

1. LFE5U devices do not have PCS Clock Dividers.

It is very 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.

Clock/Control Distribution Network

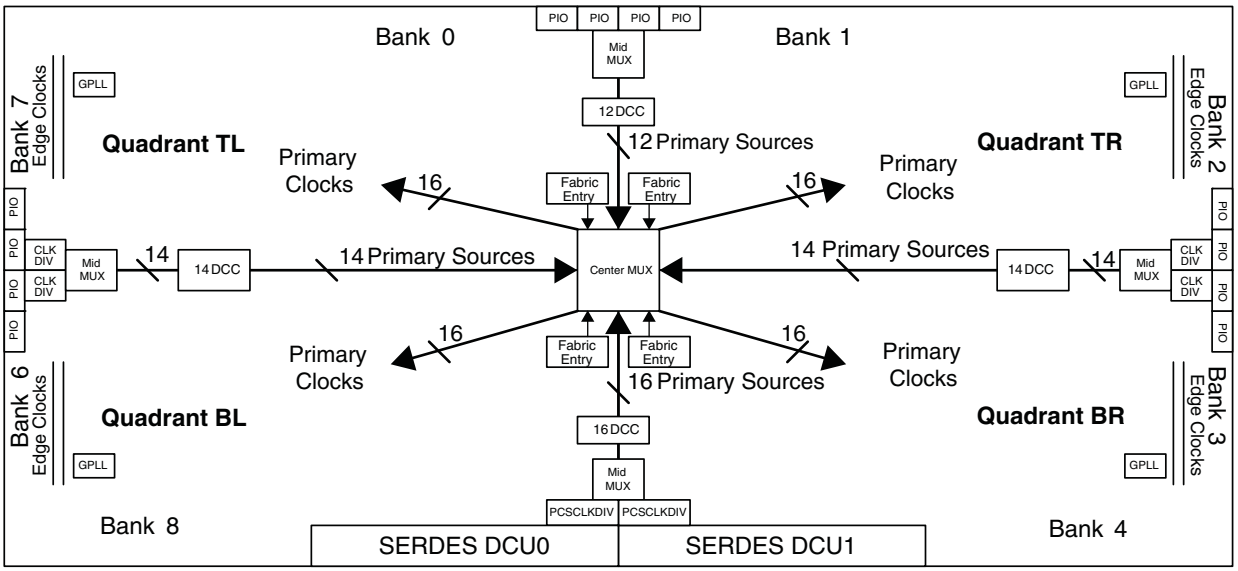
ECP5 and ECP5-5G devices provide global clock distribution in the form of 16 global primary clocks. These Primary clocks can be divided into 16 clocks per each of the four quadrants; however there is a maximum of 60 unique clock input sources. The ECP5 and ECP5-5G primary clocking structure is enhanced as it features more Edge clock resources, more low-skew Primary clock resources and removes the Secondary clock resources.

© 2015 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.

ECP5 and ECP5-5G Top-Level View

A top level view of the major clocking resources for the ECP5 and ECP5-5G devices is shown in Figure 1.

Figure 1. ECP5 and ECP5-5G Clocking Structure (LFE5-85)



Note: LFE4U devices do not have PCS Clock Dividers

Clocking Architecture Overview

Below is a brief overview of the clocking structure, elements, and PLL. Greater detail is provided starting with the Appendix A. Primary Clock Sources and Distribution section.

Primary Clock Network

Up to 60 Primary Clock Sources (PLLs, External Inputs, SERDES, etc.) can be selected and routed to the Primary Clock Network. This gives the user an available 60 unique clock domains in the ECP5 and ECP5-5G.

The Primary Clock Network provides low-skew, high fanout clock distribution to all synchronous elements in the FPGA fabric. The Primary Clock Network is divided into four clocking quadrants: Top Left (TL), Bottom Left (BL), Top Right (TR), and Bottom Right (BR). Each of these quadrants has 16 clocks that can be distributed to the fabric in the quadrant. Initially, the Lattice Diamond software automatically routes each clock to all four quadrants; up to a maximum of 16 clocks since each clock is routed to all four quadrants. The user can change how the clocks are routed by specifying a preference in the Lattice Diamond software to locate the clock to specific quadrants.

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 ECP5 and ECP5-5G. There are two edge clocks per bank located on the left and right sides.

Overview of Other Clocking Elements

Edge Clock Dividers (CLKDIVF)

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 ECP5 and ECP5-5G device.

PCS Clock Dividers (PCSCLKDIV)

A new clock divider is provided to create phase-matched divided-down clocks for bus-widening/narrowing circuits. A port is provided to dynamically change the divide value of the input clock. The PCSCLKDIV will mainly be used to slow the clock rate from high speed SERDES applications by providing the clocks for domain crossing circuits. There is one PCSCLKDIV per SERDES on an LFE5UM device.

Dynamic Clock Select (DCSC)

The ECP5 and ECP5-5G dynamic clock select provides run-time selectable glitchless or non-glitchless operation between two independent clock sources to the primary clock network. This clock select allows the selection of clock sources without leaving the dedicated clock resources in the device. There are two dynamic clock select blocks on the ECP5 and ECP5-5G device.

Edge Clock Bridge with Clock Select (ECLKBRIDGECS)

The ECLKBRIDGECS allows non-glitchless ECLK selection between two ECLKs. The ECLKBRIDGECS will allow user bridge ECLK from one side to the other. There are two of these elements and they are used for ECLK clock bridging and ECLK selection.

Edge Clock Stop (ECLKSYNCB)

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

Oscillator (OSCG)

An internal programmable rate oscillator is provided. The oscillator can be used for master configuration modes when the FPGA sources the configuration clock, Soft Error Detect (SED), and as a user logic clock source that is available after FPGA configuration. There is one OSCG on the ECP5 and ECP5-5G device. The oscillator clock output is routed directly to primary clocking.

The oscillator output is not a high-accuracy clock, having a +/- 15% variation in its output frequency. It is mainly used for circuits that do not require a high degree of clock accuracy. Examples of usage would be asynchronous logic blocks such as a timer or reset generator, or other logic that require a constantly running clock.

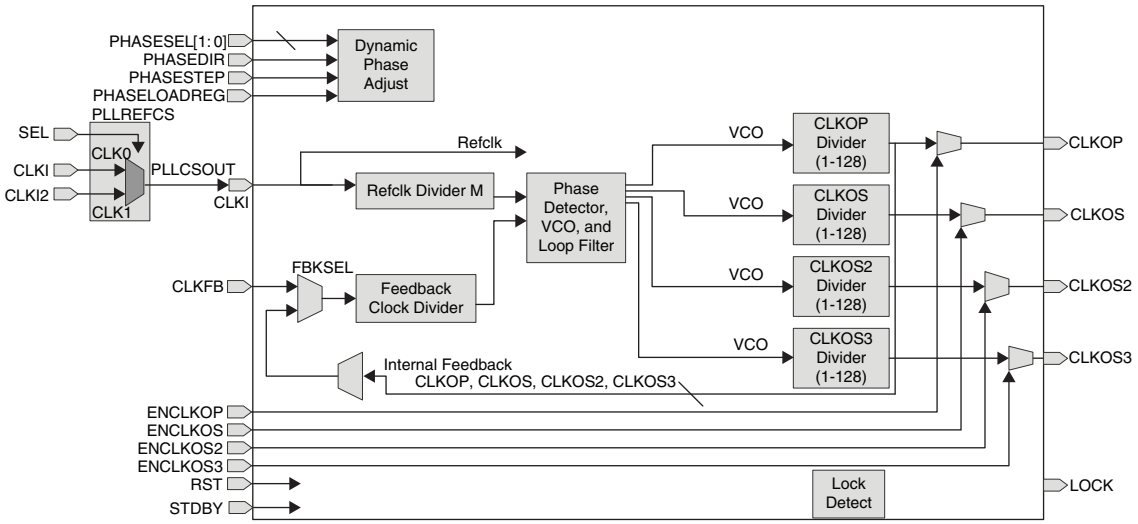
sysCLOCK PLL Overview

The sysCLOCK PLLs can be used in a variety of clock management applications such as clock injection removal, clock phase adjustment, clock timing adjustment, and frequency synthesis (multiplication and division of a clock). The ECP5 and ECP5-5G Clarity Designer PLL GUI shows important timing parameters such as the VCO rate and the PLL loop bandwidth.

PLL Input sources are:

- Dedicated PLL Input Pins
- Primary Clock Routing
- Edge Clock Routing
- FPGA Fabric

Figure 2. ECP5 and ECP5-5G PLL Block Diagram



There are four PLLs on the bigger density devices LFE5-85 and LFE5-45 and two PLLs on the smaller density LFE5-25 device. There is one PLL on each corner of the device on the bigger density devices and the smaller density devices have one PLL only on the Lower Left and Lower Right corner. Each PLL has four outputs. All four PLL outputs can go to the Primary Clock network. Only the CLKOP and CLKOS outputs can go to the ECLK network.

PLL Features

Dedicated PLL Inputs

The PLLs have dedicated PLL input pins that are not Primary Clock input pins. Each of the Top Left and Right corner PLLs have two pairs of dedicated PLL input pins, one from the Left/Right side bank and the other from the Top bank. Either one of these can be used as input to the PLL.

The bottom two PLLs have one pair of dedicated input pin on the Left/Right side banks.

Figure 3. PLL Input Pins for LFE5UM/LFE5UM5G-85 and LFE5UM/LFE5UM5G-45

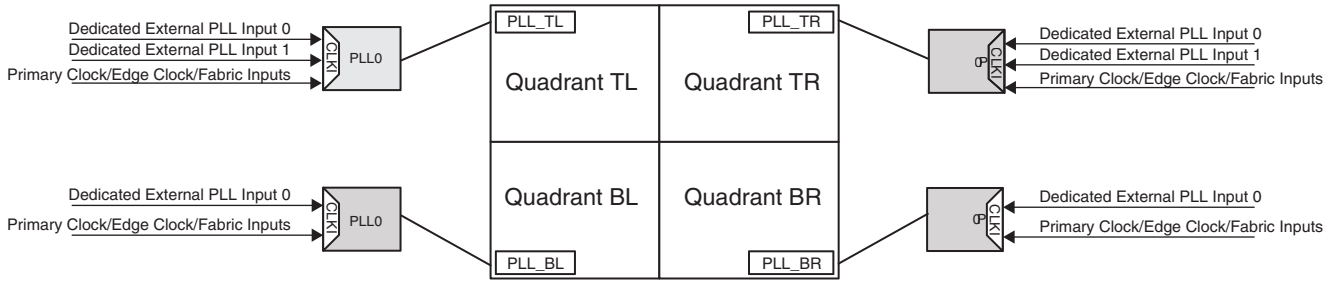
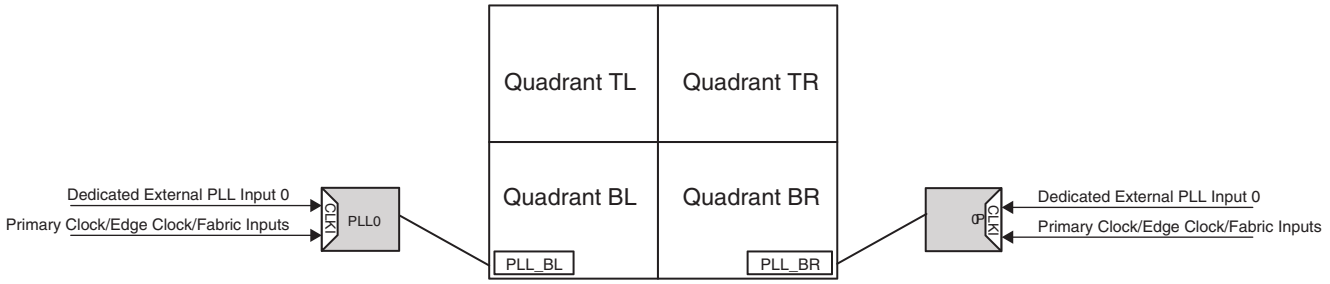


Figure 4. PLL Input Structure for LFE5-25



Input PLL Clock Selection (PLLREFCS)

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

Clock Injection Delay Removal

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

Clock Phase Adjustment

The clock phase adjustment feature of the PLL provides the ability to set a specific phase offset between the outputs of the PLL. New to the ECP5 and ECP5-5G device, phase adjustments can be calculated in much finer increments since the frequency is used to calculate the available phase increments. This feature is detailed further in the Dynamic Phase Adjustment section.

Frequency Synthesis

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

Additional Features

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

- A Standby mode to reduce power.
- Smaller phase shift increments are supported, based upon frequency.

Primary Clocks

Primary Clock Sources

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

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

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

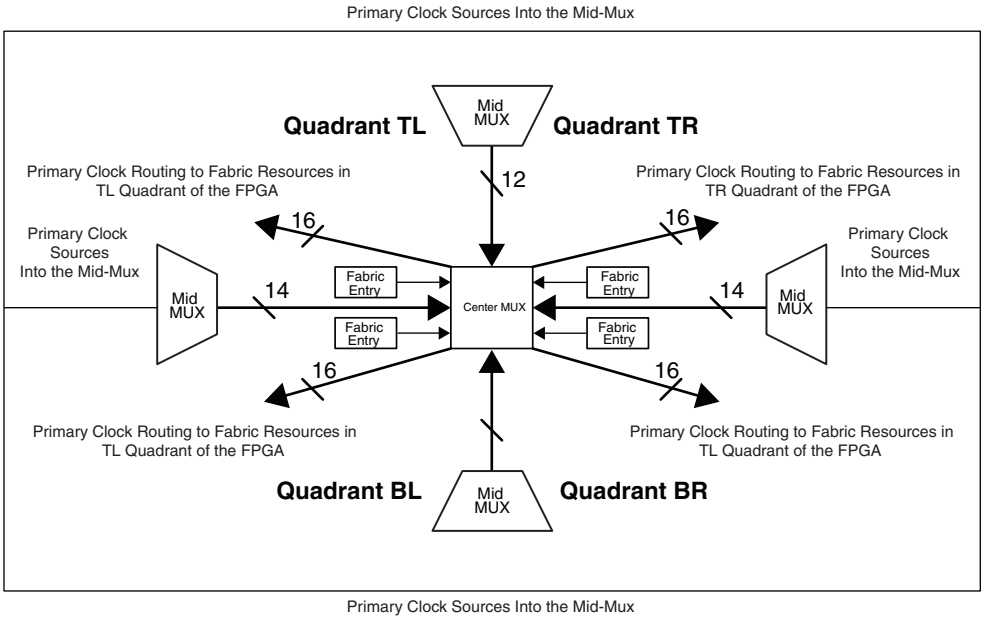
All potential primary clock sources are multiplexed prior to going to the primary clock routing by a mid-mux. There are 56 mid-mux connections and four FPGA fabric connections, 60 total, routed to a multiplexor in the center of the chip called the centermux. From the centermux primary clocks are selected and distributed to the FPGA fabric. The maximum number of unique clock sources is 16 bottom mid-mux sources + 12 top mid-mux sources + 14 left mid-mux sources + 14 right mid-mux sources + 4 direct FPGA fabric entry points (from general routing) = 60. The basic clocking structure is shown in Figure 1 and elaborated in Appendix A. Primary Clock Sources and Distribution.

Primary Clock Routing

The primary clock routing network is made up of low skew clock routing resources with connectivity to every synchronous element of the device. Primary clock sources are selected at the mid-mux, then selected in the centermux and distributed on the primary clock routing to clock the synchronous elements in the FPGA fabric.

The primary clock routing network is divided into four sections, called quadrants. Figure 5 is a simplified view of Figure 1.

Figure 5. Primary Clock Routing Architecture



The centermux can source up to 16 independent primary clocks per quadrant which can clock the logic located in that quadrant. The centermux can also route each clock source to all quadrants. The Diamond software will automatically route a primary clock to all four quadrants in the FPGA.

Dedicated Clock Inputs

The ECP5 and ECP5-5G device has dedicated pins, called PCLK pins, to bring an external clock source into the FPGA and allow them to be used as FPGA primary clocks. These inputs route directly to the Primary clock network, or to Edge clock routing resources. A dedicated PCLK clock pin must always be used to route an external clock source to FPGA logic and I/O.

If an external input clock is being sourced to a PLL, then in most cases the input clock should use a dedicated PLL input pin. SERDES reference clocks also have dedicated SERDES reference clock pins. The ECP5 and ECP5-5G device allows a PLL reference clock or a SERDES reference clock to come from an external Primary Clock (PCLK) pin and route through the Primary clock network to drive the reference clock to the SERDES or the input of a PLL.

PCS Clock Dividers (PCSCLKDIV)

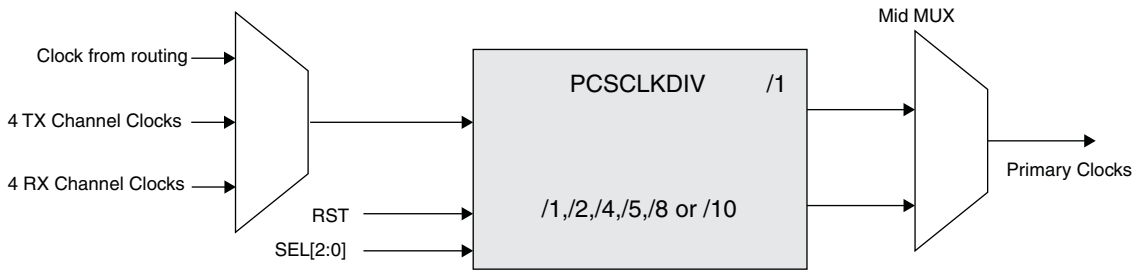
The ECP5 and ECP5-5G device has a new clock divider called the PCSCLKDIV. The PCSCLKDIV is built with our PCS / IP Core interaction in mind. Its main purpose is to allow our IP cores to do multi-rate clocking for cores that can change data rates. The specific features are:

- Capability to generate different divider clocks
- Input mux to change the divider value on-the-fly.
- Non-glitchless when the divider's select mux input is changed.
- Data width changes in the fabric for widening / narrowing circuits. This element must allow clock domain crossing in the fabric registers.

There is one PCSCLKDIV per SERDES Channel on the bottom of the device where the SERDES blocks are located. The clock outputs of the PCSCLKDIV (CDIV1, CDIVX) are delay and phase matched to each other and has a run-time selectable divider value. The PCSCLKDIV clock input sources are:

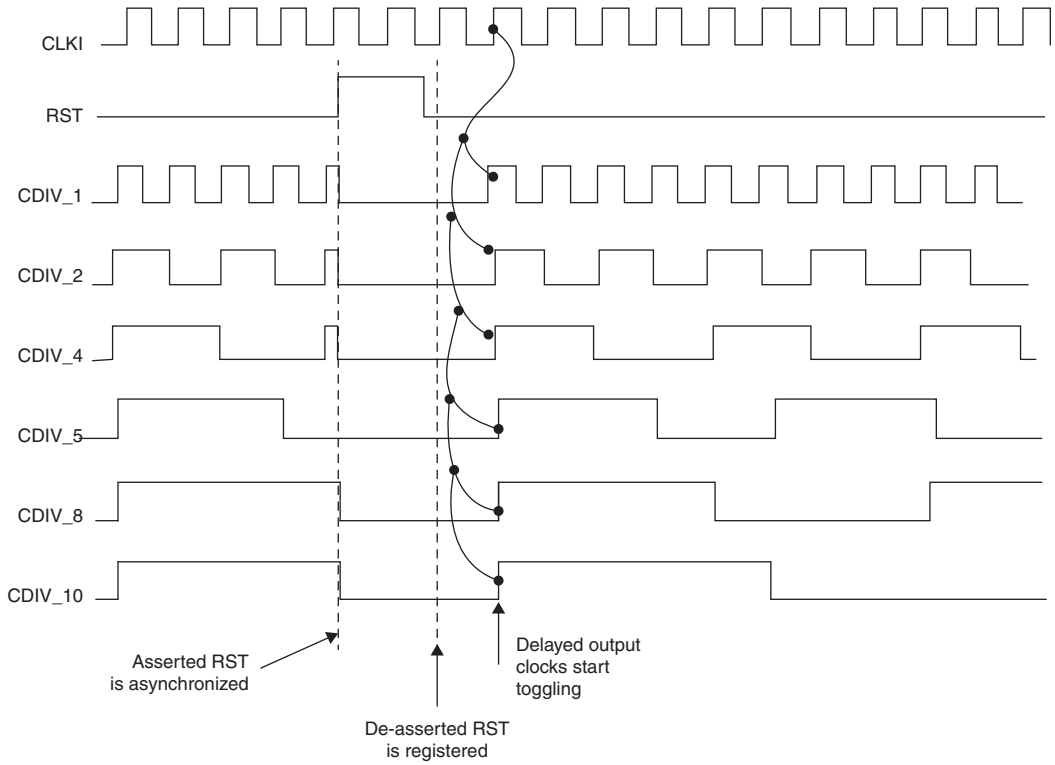
- Up to four TX Channel Clocks
- Up to four RX Channel Clocks
- Clock from Routing

Figure 6. PCSCLKDIV Connection Diagram



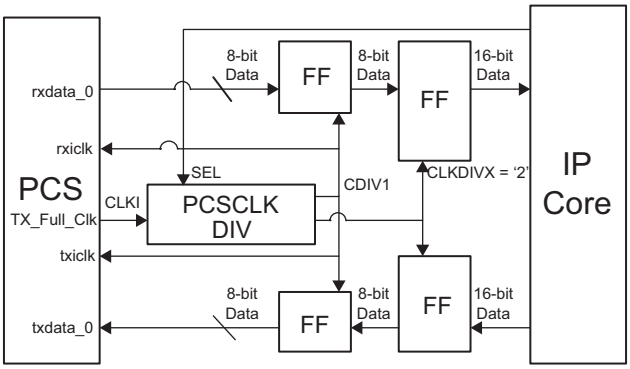
The output PCS channel clocks route directly to the input of the PCSCLKDIV without requiring the use of a primary clock. The CDIV1 and CDIVX outputs route directly to the primary clock routing. Its function is to do bus widening / narrowing circuits in the FPGA fabric to reduce the fabric frequency. The PCSCLKDIV is not suitable for DDR I/O interfaces (ECLK to SCLK domain crossing).

Figure 7. Logical Functional Timing of PCSCLKDIV Block



An example design is shown in Figure 8. It is a flip-flop based clock domain crossing circuit between the CDIV1 and the CDIVX = 2 clock outputs of the PCSCLKDIV. In this example an IP core is using the PCSCLKDIV to change the operating frequency and bus width for a PCS (SERDES) application.

Figure 8. PCSCLKDIV Example Usage



PCSCLKDIV Component Definition

The PCSCLKDIV component can be instantiated in the source code of a design as defined in this section. Figure 9, Table 2 and Table 3 define the PCSCLKDIV component. Verilog and VHDL instantiations are included.

Figure 9. PCSCLKDIV Component Symbol



Table 2. PCSCLKDIV 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 output outputs are active — RST = 1 Clock output outputs are OFF
SEL(2:0)	I	Signals are used for dynamically changing divide value. — SEL = 000, Output = '0' — SEL = 001, ÷1 — SEL = 010, ÷2 — SEL = 011, ÷4 — SEL = 100, ÷5 — SEL = 101, ÷8 — SEL = 110, ÷10 — SEL = 111, Output = '0'
CDIV1	O	Output is the same frequency as the input clock. Delay matched to CDIVX.
CDIVX	O	Divide by 1, 2, 4, 5, 8, 10 Output Port

Table 3. PCSCLKDIV Component Attribute Definition

Name	Description	Value	Default
GSR	Enable or Disable Global Reset Signal to Component	ENABLED, DISABLED	DISABLED

PCSCLKDIV Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
component PCSCLKDIV
Generic (GSR    : string);
Port    (RST    : in  STD_LOGIC;
         CLKI   : in  STD_LOGIC;
         SEL    : in  STD_LOGIC_VECTOR(2 downto 0);
         CDIV1  : in  STD_LOGIC;
         CDIVX  : out STD_LOGIC);
end component;
```

PCSCLKDIV Instantiation

```
attribute GSR : string;
attribute GSR of I1 : label is “DISABLED”;
I1: PCSCLKDIV
generic map (
  GSR     => “DISABLED”)
port map (
   RST    => RST
  ,CLKI   => CLKI
  ,SEL    => SEL
  ,CDIV1  => CDIV1
  ,CDIVX  => CDIVX);
```

PCSCLKDIV Usage in Verilog

Component and Attribute Declaration

```
module PCSCLKDIV (RST, CLKI, SEL, CDIV1, CDIVX);
parameter GSR = “DISABLED”;// “ENABLED”, “DISABLED”
input        RST, CLKI;
input  [2:0] SEL;
output       CDIVX;
endmodule
```

PCSCLKDIV Instantiation

```
defparam I1.GSR = “DISABLED”;
PCSCLKDIV I1 (
   .RST    (RST)
  ,.CLKI   (CLKI)
  ,.SEL    (SEL)
  ,.CDIV1  (CDIV1)
  ,.CDIVX  (CDIVX));
```

Dynamic Clock Select (DCSC)

The ECP5 and ECP5-5G device has two dynamic clock select (DCS) blocks that can drive to any or all the quadrants. The inputs to the DCS block come from all the output of MIDMUXs and local routing that is located at the center of the PLC array core. The output of the DCS is connected to one of the inputs of Primary Clock Center MUX.

Each DCS can select between two independent input clock sources. There are two modes of clock switching, a glitchless mode of operation, or a non glitchless mode where the DCS operates as a regular mux. Figure 10 and Figure 11 show the glitchless / non-glitchless modes of operation. It should be noted that the clock source used as feedback into the GPLL should not be switched when using the DCS as it will lead to loss of lock for the GPLL.

As indicated there are two modes of clock switching. In non-glitchless mode (input MODESEL='1') the DCS acts like a regular mux, allowing glitches and runt pulses on the output, depending on when the clock is switched. The SEL and MODESEL inputs are driven by signals from the general routing fabric and are used to support the clock switching feature in the DCS.

In glitchless mode (input MODESEL='0') the DCS avoids glitches or runt pulses on the output clock, regardless of when the enable signal is toggled. In order to switch between clocks glitchlessly, both input clocks must be oscillating; otherwise the output clock will be '0'.

For glitchless operation, the "DCSMODE" attribute sets the behavior of the DCS output. The additional attribute values and their functions are shown in Table 5.

DCS Timing Diagrams

Figure 10 shows timing diagrams to show the operation of the DCS in Glitchless mode in conjunction with the DCS-MODE attribute.

Figure 10. Timing Diagrams by "DCSMODE" Attribute Setting, Glitchless Operation (MODESEL='0')

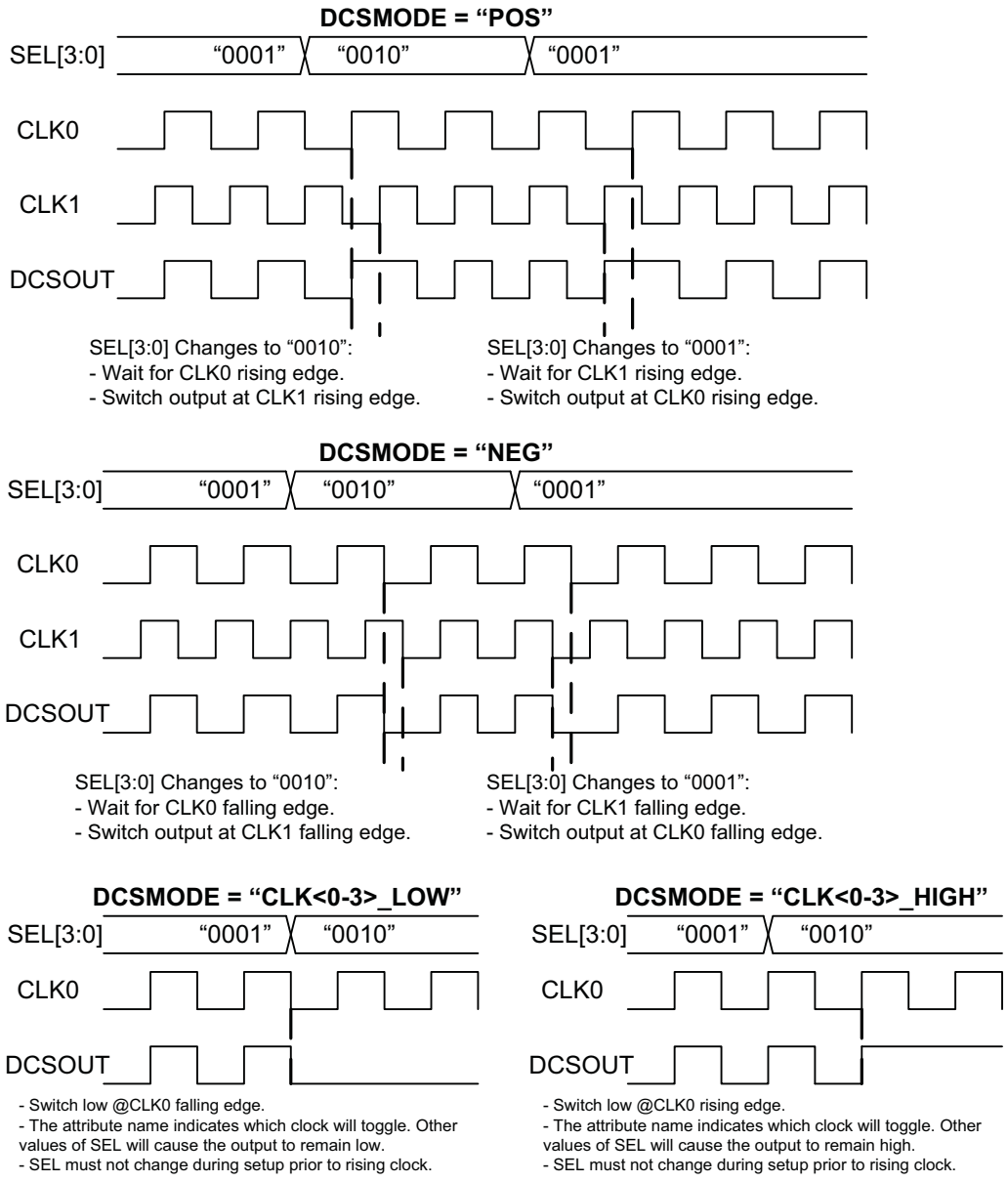
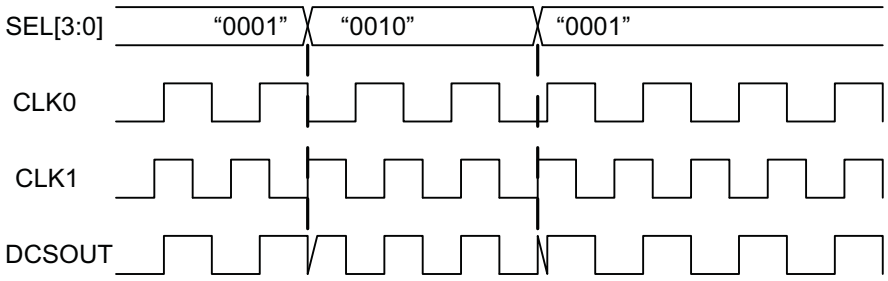


Figure 11. DCS Operation, Non-Glitchless Operation (MODESEL='1')



DCSC Component Definition

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

Figure 12. DCSC Component Symbol

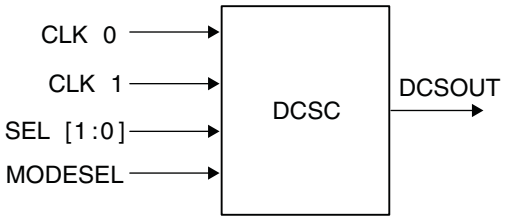


Table 4. DCSC Component Port Definition

Port Name	I/O	Description
CLK0	I	Clock Input port 0 – Default.
CLK1	I	Clock Input port 1
SEL[1:0]	I	Input Clock Select bit (See Table 6 below)
MODESEL	I	Selects Glitchless ('0') or Non-Glitchless ('1') behavior.
DCSOUT	O	Clock Output Port

DCSMODE Attribute

Table 5 provides the behavior of the DCS output based on the setting of the DCSMODE attribute when the pin MODESEL ='0'. The MODESEL pin is dynamic and can toggle during operation. Table 5 is only valid when MODESEL ='0'.

Table 5. DCSC – DCSMODE Attribute Table when MODESEL='0'

Attribute Name	Description	SEL[1:0]			Attribute Value
		2'b01	2'b10	2'b00/2'b11	
DCSMODE Attributes	Falling Edge Triggered	Clk0	Clk1	0	NEG
	Rising Edge Triggered	Clk0	Clk1	1	POS
	Disabled Output is Low, Clk0	Clk0	0	0	CLK0_LOW
	Disabled Output is High, Clk0	Clk0	1	1	CLK0_HIGH
	Disabled Output is Low, Clk0	0	Clk1	0	CLK1_LOW
	Disabled Output is High, Clk0	1	Clk1	1	CLK1_HIGH
	Clk0 Buffered	Clk0	Clk0	Clk0	CLK0
	Clk1 Buffered	Clk1	Clk1	Clk1	CLK1
	Tie Low	0	0	0	LOW
	Tie High	1	1	1	HIGH
MODESEL= "1"	Non-Glitchless	Clk0	Clk1	0	-

DCSC Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
COMPONENT DCSC
GENERIC(DCSMODE  : string := "POS");
PORT   (CLK0     :IN  STD_LOGIC;
  CLK1     :IN  STD_LOGIC;
  SEL      :IN  STD_LOGIC_VECTOR(1 downto 0);
  MODESEL  :IN  STD_LOGIC;
  DCSOUT   :OUT STD_LOGIC);
END COMPONENT;
```

DCSC Instantiation

```
attribute DCSMODE : string;
attribute DCSMODE of DCSinst0 : label is "POS";
I1: DCSC
generic map(
   DCSMODE  => "POS")
port map   (
   CLK0     => CLK0
  ,CLK1     => CLK1
  ,SEL      => SEL
  ,MODESEL  => MODESEL
  ,DCSOUT   => DCSOUT);
```

DCSC Usage in Verilog

Component and Attribute Declaration

```
module DCSC(CLK0,CLK1,CLK2,CLK3,SEL,MODESEL,DCSOUT);
input         CLK0;
input         CLK1;
input  [1:0]  SEL;
input         MODESEL;
output        DCSOUT;
endmodule
```

DCSC Instantiation

```
defparam DCSInst0.DCSMODE    = "POS";
DCSC DCSInst0 (
 .CLK0      (CLK0)
,.CLK1      (CLK1)
,.SEL       (SEL)
,.MODESEL   (MODESEL)
,.DCSOUT    (DCSOUT));
```

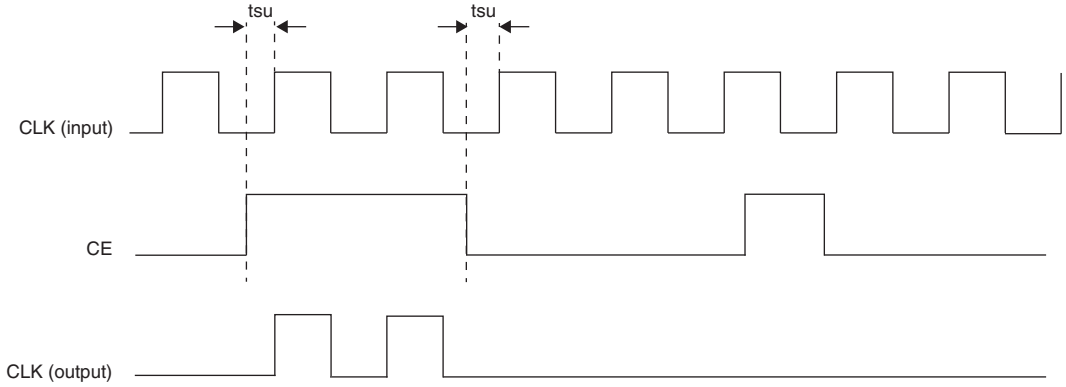
Dynamic Clock Control (DCCA)

The ECP5 and ECP5-5G device has a Dynamic Clock Control feature which allows internal logic dynamically to enable or disable quadrant primary clock network. The disable function will not create glitch and increase the clock latency to the primary clock network. Also, this dynamic clock control function can be disabled by a configuration memory fuse to always enable the primary clock network.

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

The ECP5 and ECP5-5G device clock architecture allows both DCC and DCS to function at the same. It should be noted that the clock source used as feedback into the GPLL should always keep enable signal "high" in the DCC, otherwise it will lead to loss of lock for the GPLL when toggling the enable signal.

Figure 13. Glitchless DCC Functional Waveform



DCCA Component Definition

The DCCA component can be instantiated in the source code of a design as defined in this section. Figure 14 and Table 6 show the DCCA definitions.

Figure 14. DCCA Component Symbol

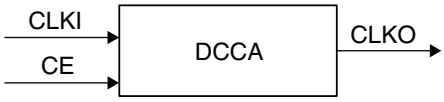


Table 6. DCCA Component Port Definition

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

DCCA Usage in VHDL

Component Instantiation

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

DCCA Instantiation

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

DCCA Instantiation

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

Internal Oscillator (OSCG)

The OSCG element performs multiple functions on the ECP5 and ECP5-5G device. It is used for configuration, SED, as well as optionally in user mode. In user mode, the OSCG element has the following features:

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

OSCG Component Definition

The OSCG component can be instantiated in the source code of a design as defined in this section. Figure 15 and Table 7 below show the OSCG definitions.

Figure 15. OSCG Component Symbol

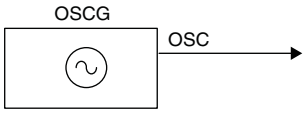


Table 7. OSCG Component Port Definition

Port Name	I/O	Description
OSC	O	Clock Output Port

Table 8. OSCG Component Attribute Definition

Name	Description	Value	Default
DIV	Selects Divider Value	2 (~155.0 MHz) – 128 (~2.4 MHz)	2.4 MHz

OSCG Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
component OSCG
Port    (OSC      : out STD_LOGIC);
end component;
```

OSCG Instantiation

```
I1: OSCG
port map (
   OSC      => OSC);
```

OSCG Usage in Verilog

Component and Attribute Declaration

```
module OSCG (OSC);
output  OSC;
endmodule
```

OSCG Instantiation

```
OSCG I1 (.OSC(OSC))
```

Edge Clocks

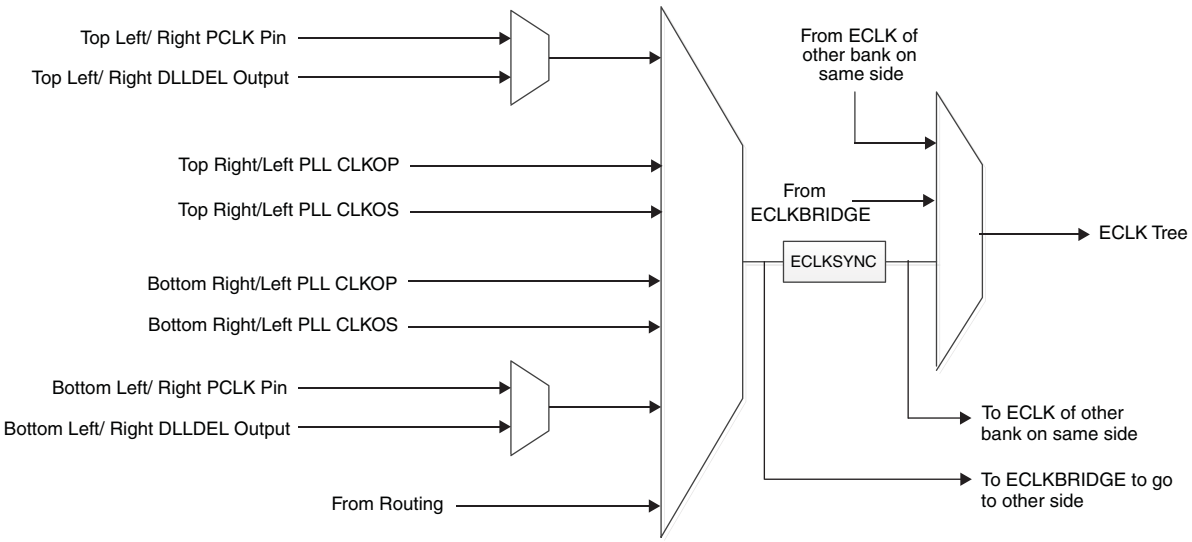
Each ECP5 and ECP5-5G device I/O bank has four ECLK resources. These clocks, which have low injection time and skew, are used to clock I/O registers. Edge clock resources are designed for high speed I/O interfaces with high fan-out capability. Refer to Appendix B. EDGE CLOCK Sources and Connectivity for detailed information on the ECLK locations and connectivity.

The sources of edge clocks are:

- Dedicated Clock (PCLK) pins
- DLLDEL output
- PLL outputs (CLKOP & CLKOS)
- ECLK Bridge
- Internal nodes

The ECP5 and ECP5-5G device has Edge Clock (ECLK) at the Left side and right side of the device. There are two ECLK network per bank IO. ECLK Input MUX collects all clock sources available shown in figure below. There are two ECLK Input MUXs, one on the left side and one on the right side. Each of these MUX will generate total of four ECLK Clock sources. Two of them drive the upper IO bank and two of them drive the lower IO bank. Two out of four also drive the ECLK Bridge Switch Block to form an ECLK Bridge high speed clock before drive the ECLK Tree Network.

Figure 16. Edge Clock Sources Per Bank



Edge Clock Dividers (CLKDIVF)

There are four edge clock dividers available in the ECP5 and ECP5-5G device, two per side of the device. The clock divider provides a single divided output with available divide values of 2 or 3.5. The inputs to the clock dividers are the edge clocks, PLL outputs and Primary Clock Input pins. The outputs of the clock divider drive the primary clock network and are mainly used for DDR I/O domain crossing.

CLKDIVF Component Definition

The CLKDIVF component can be instantiated in the source code of a design as defined in this section. Figure 17, Table 9, and Table 10 define the CLKDIVF component. Verilog and VHDL instantiations are included.

Figure 17. CLKDIVF Component Symbol



Table 9. CLKDIVF 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 or 3.5 Output Port

Table 10. CLKDIVF Component Attribute Definition

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

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

CLKDIVF Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
component CLKDIVF
Generic (DIV      : string;
         GSR      : string);
Port    (RST      : in  STD_LOGIC;
         CLKI     : in  STD_LOGIC;
         ALIGNWD  : in  STD_LOGIC;
         CDIVX    : out STD_LOGIC);
end component;
```

CLKDIVF Instantiation

```
attribute DIV : string;
attribute DIV of I1 : label is “2.0”;
attribute GSR : string;
attribute GSR of I1 : label is “DISABLED”;

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

CLKDIVF Usage in Verilog

Component and Attribute Declaration

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

parameter DIV = “2.0”;        // “2.0”, “3.5”
parameter GSR = “DISABLED”;   // “ENABLED”, “DISABLED

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

CLKDIVF Instantiation

```
defparam I1.DIV = “2.0”;
defparam I1.GSR = “DISABLED”;
CLKDIVF I1 (
   .RST      (RST)
  ,.CLKI     (CLKI)
  ,.ALIGNWD  (ALIGNWD)
  ,.CDIVX    (CDIVX));
```

Edge Clock Bridge (ECLKBRIDGECS)

ECLK Clock Bridge provides to bridge the ECLK for banks on the same side or the ECLK of the left side and the right side. The ECLK Bridge enhances the communication of high speed clocks of the two edges with minimum skew to ECLK tree.

There are two ECLK bridge components in the ECP5 and ECP5-5G device. There are two ECLK muxes on the left and two ECLK muxes on the right and they allow a user to bridge edge clocks to the left and right sides of the chip with minimal skew.

In the edge clock bridge there is a non-glitchless clock select mux that allows a design to switch between two different clock sources for each edge clock. This clock select mux is instantiated using the ECLKBRIDGECS primitive. Not all edge clocks can drive the ECLKBRIDGECS, one of the two ECLKs on the left and right side can be bridged. When connected to the ECLKBRIDGECS it would use up the ECLK1 of both banks on the left side and ECLK0 of both banks on the right side of the device.

ECLKBRIDGECS Component Definition

The ECLKBRIDGECS component must be instantiated in the source code of a design in order to bridge Edge clocks. Figure 18 and Table 11 define the ECLKBRIDGECS.

Note that an ECLKSYNCB module has to be always used after the ECLKBRIDGECS instance to be able to connect to the ECLK tree.

Figure 18. ECLKBRIDGECS Component Symbol



Table 11. ECLKBRIDGECS Component Port Definition

Port Name	I/O	Description
CLK0	I	Clock Input Port 0 – Default
CLK1	I	Clock Input Port 1
SEL	I	Select Port — SEL = 0 for CLK0 — SEL = 1 for CLK1
ECSOUT	O	Clock Output Port

ECLKBRIDGECS Usage in VHDL

Component Instantiation

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

Component and Attribute Declaration

```
COMPONENT ECLKBRIDGECS
PORT    (CLK0     :IN  STD_LOGIC;
  CLK1     :IN  STD_LOGIC;
  SEL      :IN  STD_LOGIC;
  ECSOUT   :OUT STD_LOGIC);
END COMPONENT;
```

ECLKBRIDGECS Instantiation

```
I1: ECLKBRIDGECS
port map    (
   CLK0     => CLK0
  ,CLK1     => CLK1
  ,SEL      => SEL
  ,ECSOUT   => ECSOUT);
```

ECLKBRIDGECS Usage in Verilog

Component and Attribute Declaration

```
module ECLKBRIDGECS (CLK0,CLK1,SEL,ECSOUT);
input   CLK0;
input   CLK1;
input   SEL;
output  ECSOUT;
endmodule
```

DCSC Instantiation

```
ECLKBRIDGECS ECSInst0 (
 .CLK0     (CLK0)
,.CLK1     (CLK1)
,.SEL      (SEL)
,.ECSOUT   (ECSOUT));
```

Edge Clock Synchronization (ECLKSYNCB)

ECP5 and ECP5-5G 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 is it not needed and thus save power. Applications such as DDR2, DDR3 and 7:1 LVDS for displays will use this component for clock synchronization.

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 as DDR memory, generic DDR and 7:1 LVDS.

Control signal STOP is synchronized with ECLK when asserted. When control signal STOP is asserted, the clock output 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 19 and Figure 12 show the ECLKSYNCB component definition.

Figure 19. ECLKSYNCB Component Symbol

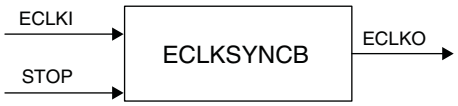
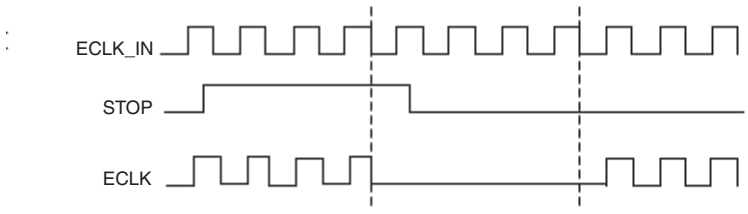


Table 12. ECLKSYNCB Component Port Definition

Port Name	I/O	Description
ECLKI	I	Clock Input port.
STOP	I	Control signal to stop Edge Clock — STOP = 0 Clock is Active — STOP =1 Clock is Off
ECLKO	O	Clock Output Port

Figure 20. ECLKSYNC Functional Waveform



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

ECLKSYNCB Instantiation

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

General Routing for Clocks

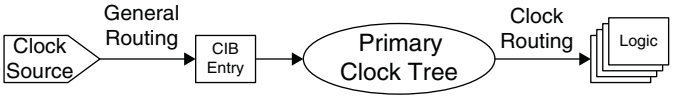
The ECP5 and ECP5-5G device architecture supports the ability to use data routing, or general routing, for a clock. This capability is intended to be used for small areas of the design to allow additional flexibility in linking dedicated clocking resources and building very small clock trees. General 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.

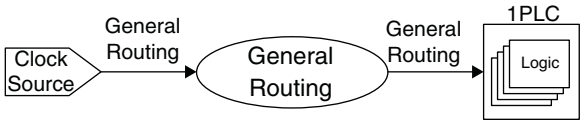
There are multiple entry points to the Primary clock routing throughout the ECP5 and ECP5-5G device fabric. In this case it is recommended to add a preference for this gated clock to use primary routing.

Figure 21. Gated Clock to the Primary Clock Routing



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 22. Gated Clock to Small Logic Domain



General routing PCLK pins

Some Dedicated pins (GR_PCLK) can access some CIBs along the edge of the device and can drive the primary clock routing from this CIB. These will go through some general routing but have direct access to the primary clock from this dedicated CIB. There are four at the left side and right side, four at the top side 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.

sysCLOCK PLL

The ECP5 and ECP5-5G PLL provides features such as clock injection delay removal, frequency synthesis, and phase adjustment. Figure 23 shows a block diagram of the ECP5 and ECP5-5G PLL.

Figure 23. ECP5 and ECP5-5G PLL Block Diagram

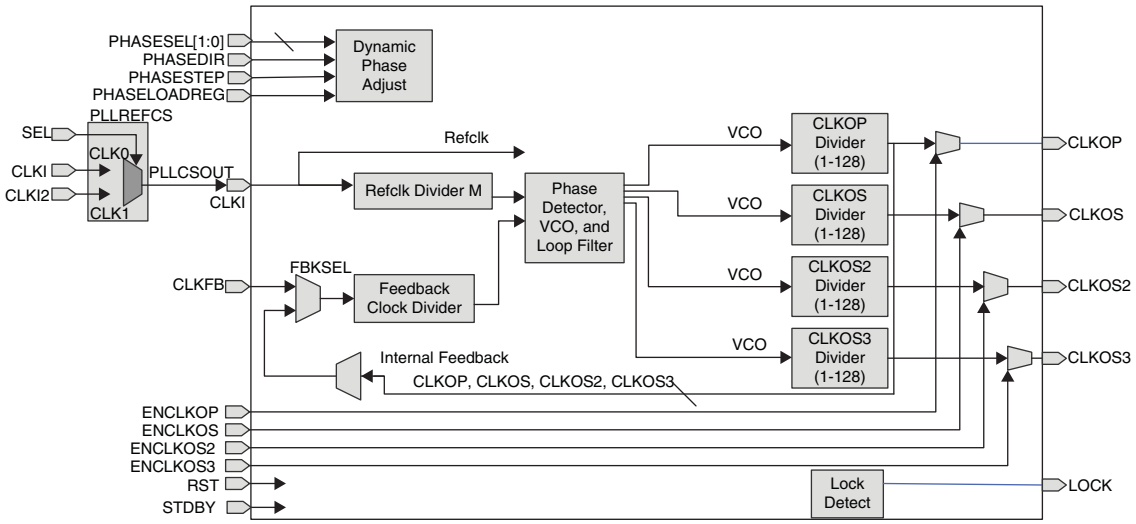


Figure 24. PLL Component Instance

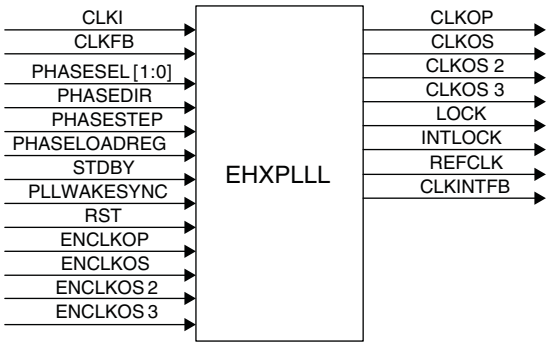


Table 13. PLL Component Port Definition

Signal	I/O	Description
CLKI	I	Input Clock to PLL.
CLKFB	I	Feedback Clock
CLKINTFB	O	Internal Feedback Clock
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.
STDBY	I	Standby signal to power down the 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 wake 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.
INTLOCK	O	Internal Lock Signal
REFCLK	O	Output of Reference clock mux

Functional Description

Refclk (CLKI) Divider

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

Feedback Loop (CLKFB) Divider

The CLKFB divider is used to divide the feedback signal, effectively multiplying the output clock. The VCO block increases the output frequency until the divided feedback frequency equals the input frequency. The output of the feedback divider must be within the phase detector frequency range specified in the device data sheet. This port is only available to user interface when “user clock” option is selected for feedback clock, otherwise this port 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. New to the ECP5 and ECP5-5G device, phase adjustments are now calculated values in the software tools based on VCO clock frequency. This provides a finer phase shift depending on the required frequency. The clock output selected as the feedback cannot use the static phase adjustment feature since it will cause the PLL to unlock. For example if the FB_MODE is INT_OP or CLKOP, then there should be no phase shift on CLKOP. Similar restriction would apply 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 the feedback should not use the dynamic phase adjustment feature. Please see the Dynamic Phase Adjustment section in this document for usage details. The clock output selected as the feedback cannot use the dynamic phase adjustment feature since it will cause the PLL to unlock. For example if the FB_MODE is INT_OP or CLKOP, then there should be no phase shift on CLKOP. Similar restriction would apply on other clocks.

PLL Features

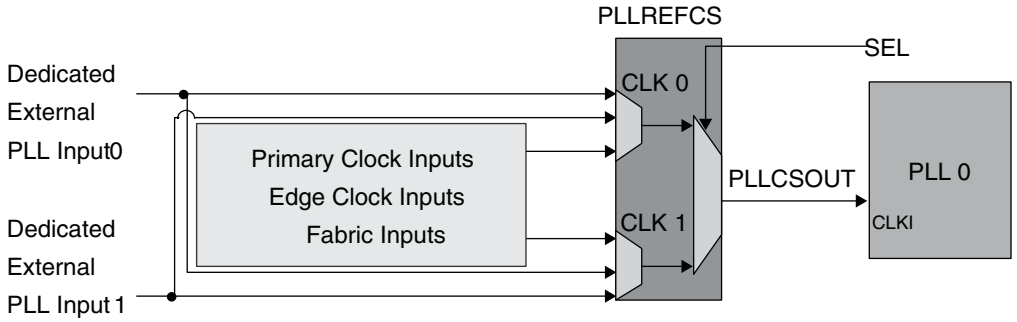
Dedicated PLL Inputs

Every PLL has a dedicated low skew input that will route directly to its reference clock input. These are the recommended inputs for a PLL. It is possible to route a PLL input from the Primary clock routing, but it incurs more clock input injection delay, which is not natively compensated for using feedback, than a dedicated PLL input. There is one PLLs in each corner of the FPGA on bigger densities. Each of the PLL on the top left and right corners have two pairs dedicated PLL inputs that user can choose from. Both of these inputs can route to the PLL in the top side corner. The PLLs on the bottom corners each have one pair of dedicated PLL input pin.

PLL Input Clock Mux (PLLREFCS)

Each ECP5 and ECP5-5G PLL contains an input mux to dynamically switch between two input reference clocks. The output of the PLLREFCS is routed directly into the PLL. There is one PLLREFCS component in each top left and right corner of the FPGA. In order to enhance the clock muxing capability of the ECP5 and ECP5-5G device, the dedicated clock inputs for the PLLs in each corner are routed to the PLLREFCS component in a corner along with the other potential PLL sources such as edge clocks and primary clocks. This structure is seen in Figure 25.

Figure 25. PLL Dedicated Inputs to the PLLREFCS Component for Top Left and Right PLL



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

Standby Mode

The ECP5 and ECP5-5G device PLL contains a Standby Mode that allows the PLL to be placed into a standby state to save power when not needed in the design. Standby mode is very similar to holding the PLL in reset since the VCO will be turned off and will need to regain lock when exiting standby. In both cases, reset and standby mode, the PLL will retain its programming.

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

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

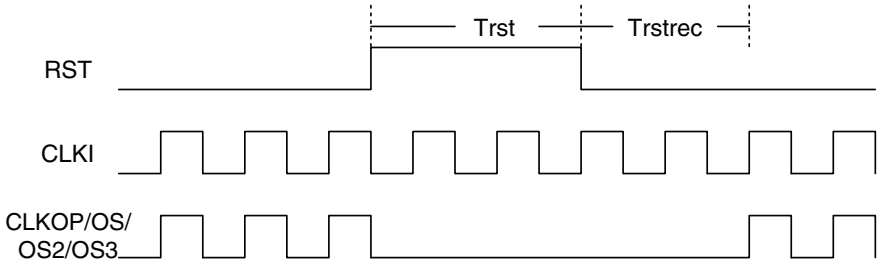
The CLKFB signal is the feedback signal to the PLL. The feedback signal is used by the Phase Frequency Detector inside the PLL to determine if the output clock needs adjustment to maintain the correct frequency and phase. The CLKFB signal can come from a primary clock net (feedback mode = CLKO[P/S/S2/S3]) 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 to create an input to the VCO of the PLL. 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 will cause the outputs to be logic ‘0’. In bypass mode the output will not be reset.

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

Figure 26. 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 14. 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 the user to save power by stopping the corresponding output clock when not in use. The clock enable signals are optional and will only be available if the user has selected 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 in order to avoid shutting off the feedback clock input.

STDBY Input

The STDBY signal is used to put the PLL into a low power standby mode when it is not required. The STDBY signal is optional and will only be available if the user has selected the Standby port option in Clarity Designer. The STDBY signal is active high. When asserted the PLL outputs are pulled to "0" and the PLL will be reset. 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.

Dynamic Phase Shift Inputs

The ECP5 and ECP5-5G PLL has five ports to allow for dynamic phase adjustment from FPGA logic. The Dynamic Phase Adjustment section will elaborate on how the user should drive these ports.

PHASESEL Input

The PHASESEL[1:0] inputs are used to specify which PLL output port will be affected by the dynamic phase adjustment ports. The settings available are shown in the Dynamic Phase Adjustment section. The PHASESEL signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed. The PHASESEL signal is optional and will be available if the user has selected the "Dynamic Phase Ports" option in Clarity Designer.

Table 15. 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 which direction the dynamic phase shift will occur, advanced (leading) or delayed (lagging). When PHASEDIR = 0 then the phase shift will be delayed. When PHASEDIR = 1 then the phase shift will be advanced. The PHASEDIR signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed. The PHASEDIR signal is optional and will be available if the user has selected the Dynamic Phase ports option in Clarity Designer.

Table 16. 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 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 5 ns prior to PHASESTEP falling 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 and in the direction specified by the PHASESEL and PHASEDIR inputs. A phase shift is started on the falling edge of the PHASELOADREG signal and there is a minimum pulse width of 10 ns from assertion to deassertion. The PHASESEL and PHASEDIR are required to have a setup time of 5 ns prior to PHASELOADREG falling edge.The PHASELOADREG signal is optional and 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 17. All four outputs can be routed to the Primary clock routing of the FPGA. All four outputs can be phase shifted statically or dynamically if external feedback on the clock is not used. They can also statically or dynamically adjust their output duty cycle. The outputs can come from their output divider or the reference clock input (PLL bypass). In bypass mode the output divider can be bypassed or used to divide the reference clock.

Table 17. PLL Clock Outputs and ECLK Connectivity

Clock Output Name	Edge Clock Connectivity	Selectable Output
CLKOP	Right, Left ECLKs	Always Enabled
CLKOS	Right, Left ECLKs	Selectable via Clarity Designer
CLKOS2	No ECLK Connection	Selectable via Clarity Designer
CLKOS3	No ECLK Connection	Selectable via 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 will achieve lock within 16 ms. Once lock is achieved, the PLL LOCK signal will be 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 sticky lock mode by selecting “PLL Lock is Sticky”. In sticky lock mode, once the LOCK signal is asserted (logic ‘1’) it will stay 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 will lose lock and the LOCK output will de-assert (logic ‘0’). It is recommended to assert PLL RST to re-synchronize the PLL to the reference clock when the PLL loses lock. The LOCK signal is available to the FPGA routing to implement the generation of the RST signal if requested by the designer. The LOCK signal is optional and will be available if the user has selected the Provide PLL Lock signal option in Clarity Designer.

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 & CLKOS3 have the dynamic phase adjustment feature but only one output clock can be adjusted at a time. Table above shows the output clock selection settings available for the PHASESEL[1:0] signal. The PHASESEL signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed.

The selected output clock phase will either be advanced or delayed depending upon the value of the PHASEDIR port or signal. Table 16 shows the PHASEDIR settings available. The PHASEDIR signal must be stable for 5 ns before the PHASESTEP or PHASELOADREG signals are pulsed.

VCO Phase Shift

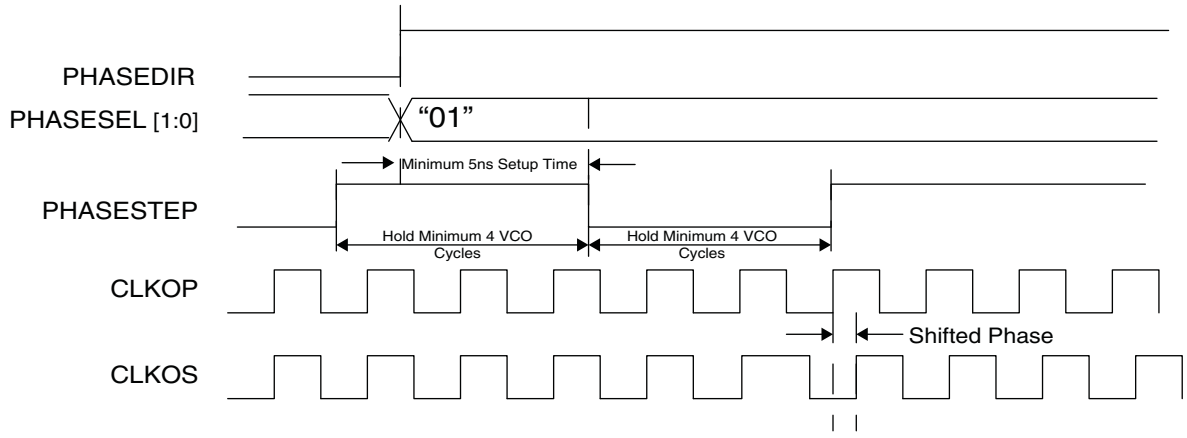
Once the PHASESEL and PHASEDIR have been set, a VCO phase adjustment is made by toggling the PHASESTEP signal from the current setting. Each pulse of the PHASESTEP signal will generate a phase step based on this equation:

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

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

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

Figure 27. 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 5 ns before the falling edge of PHASESTEP and the minimum pulse width of PHASESTEP should be four VCO clock cycles. It should also stay low for four VCO Clock Cycles.

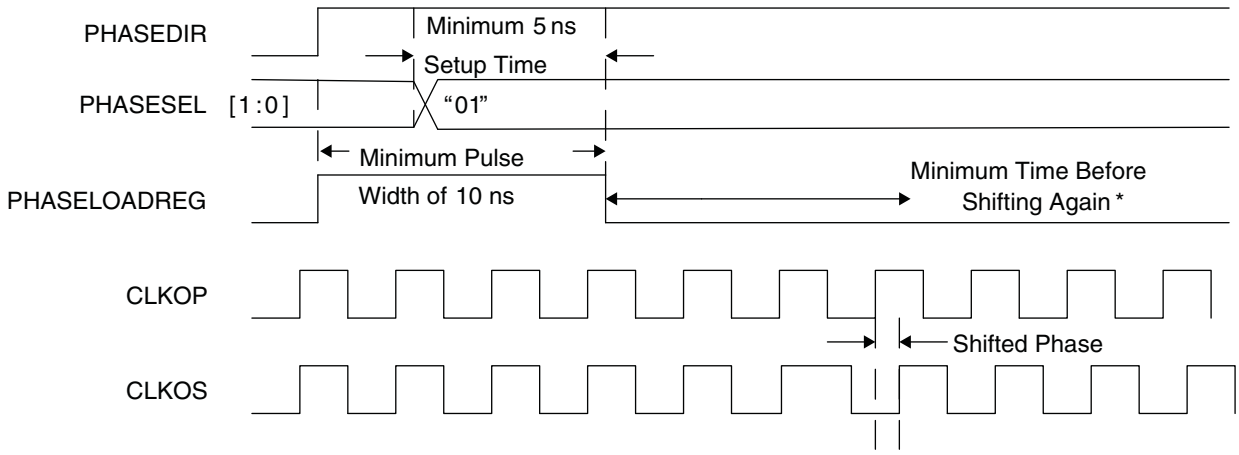
For each toggling of PHASESTEP, you will get [1/(8*2)]*360 = 22.5 degree phase shift (delayed).

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

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

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

Figure 28. Divider Phase Shift Timing Diagram



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

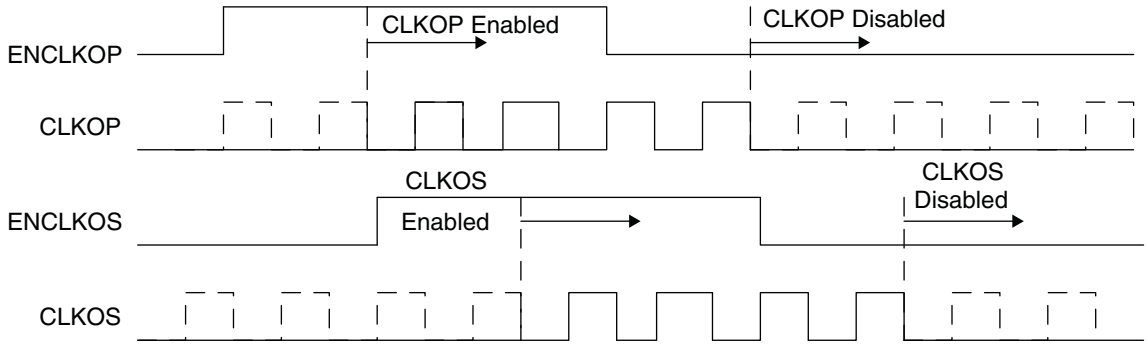
Low Power Features

The ECP5 and ECP5-5G 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 not used in the design. A disabled output clock will be 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.

Figure 29. Dynamic Clock Enable for PLL Outputs



Standby Mode

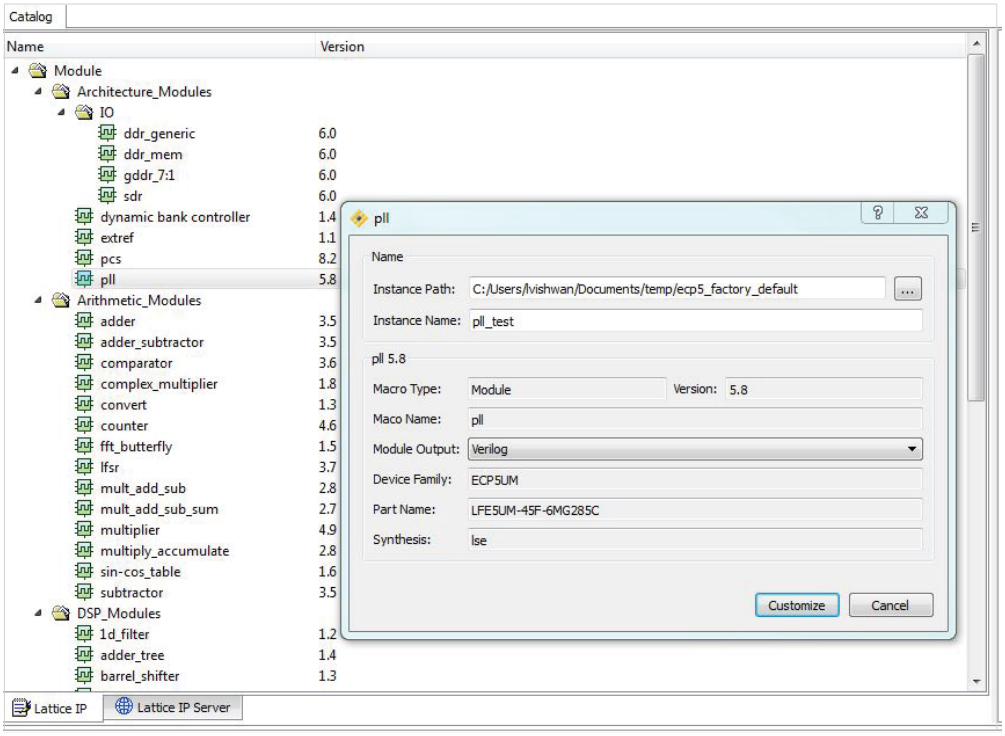
The PLL can also be put into standby mode. This is similar to reset in that the PLL is still powered, however the VCO is not running and the clock outputs driven low. The PLL will enter Standby mode when the STDBY signal is driven high and the outputs will be driven low. Users need to stay in the STDBY mode for at least 1 ms to make sure the PLL analog circuits are fully reset and to have a stable analog startup. The PLL can be restarted when it is needed again and the output clocks will be reactivated. It will take Tlock_time = 16 us to achieve PLL lock again. To support this mode the "Standby Port" option is in the Clarity Designer GUI and will cause the STDBY port to be brought out to the top level of the PLL module.

PLL Usage in Clarity Designer

It is expected that Clarity Designer will be used to create and configure a PLL. PLL can be found in the **Catalog** tab of Clarity Designer under Module - Architecture Modules. The graphical user interface is used to select parameters for the PLL. The result is an HDL block to be used in the simulation and synthesis flow.

The main window when the PLL is selected is shown in Figure 30. When opening Clarity Designer inside a Diamond project, the only entry required is the file name as the other entries are set to the project settings. If Clarity Designer is opened as a stand-alone tool then it is necessary to supply the additional parameters shown on this screen. After entering the module name of choice, clicking on Customize will open the PLL configuration window as shown in Figure 30.

Figure 30. Clarity Designer Main Window for PLL Module



Configuration Tab

The configuration window lists all user accessible attributes with default values set. Upon completion, clicking Generate generates the source.

PLL Frequency and Phase Configuration

Enter the input and output clock frequencies and the software will calculate the divider settings. After the input and output frequencies are entered clicking the Calculate button will display the divider values and the closest achievable frequency will be displayed in the “Actual Frequency” text box. If an entered value is out of range it will be displayed in red and an error message will be displayed. The user can also select a tolerance value from the “Tolerance %” drop-down box. When the Calculate button is pressed the calculation will be considered accurate if the result is within the entered tolerance percentage range.

New to the ECP5 and ECP5-5G PLL GUI, the user enters the desired phase shift and the software will calculate the closest achievable shift. After the desired phase is entered, clicking the Calculate button will display the closest achievable phase shift in the “Actual Phase” text box. If an entered value is out of range it will be displayed in red and an error message will be displayed.

Figure 31. ECP5 and ECP5-5G PLL Frequency Configuration Tab

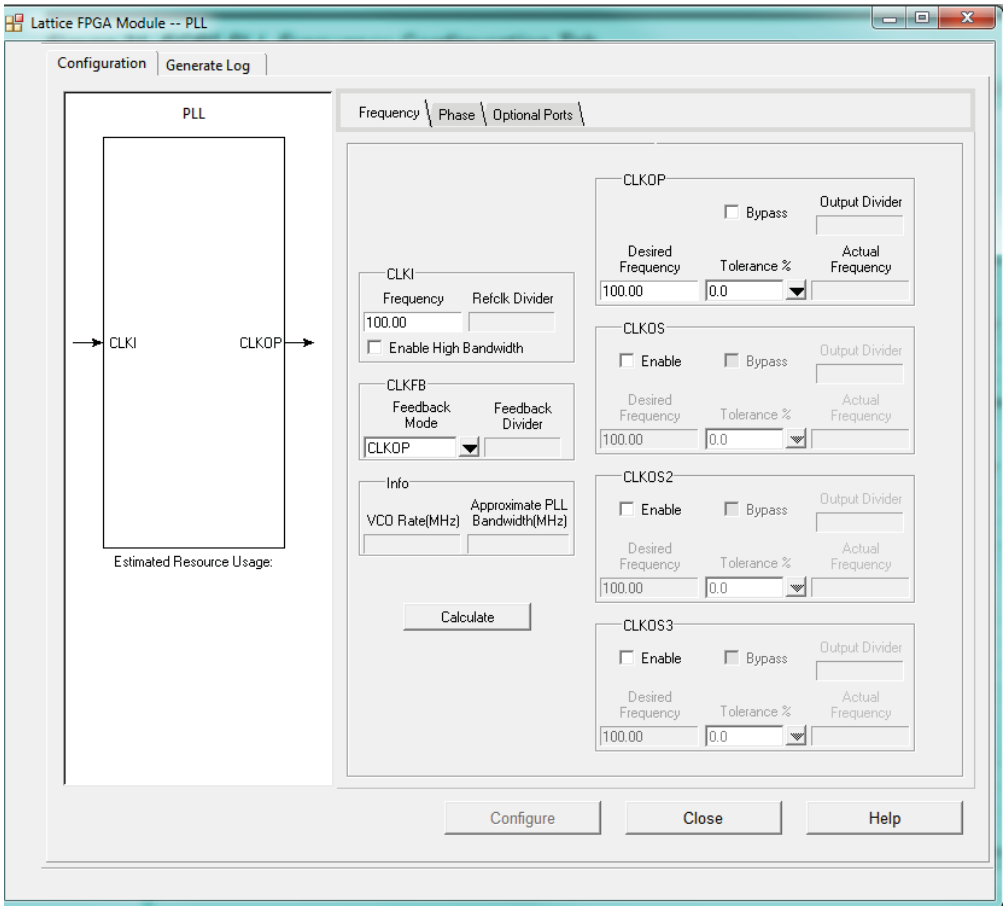


Table 18. Page 1, PLL Frequency Settings, Clarity Designer GUI

User Parameters	Description	Range	Default	Corresponding HDL Attribute
CLKI	Frequency Input	10 – 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	
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

User Parameters	Description	Range	Default	Corresponding HDL Attribute
CLKOP	Enable	ON / OFF	OFF	CLKOP_ENABLE
	Bypass	ON / OFF	OFF	OUTDIVIDER_MUXA
	Output Divider (read only)	—	—	CLKOP_DIV
	Desired Frequency *1	3.125 – 400 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	OFF	CLKOS_Enable
	Bypass	ON / OFF	OFF	OUTDIVIDER_MUXB
	Clock Divider (read only)	—	—	CLKOS_DIV
	Desired Frequency *1	3.125 – 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	OFF	CLKOS2_Enable
	Bypass	ON / OFF	OFF	OUTDIVIDER_MUXC
	Clock Divider (read only)	—	—	CLKOS2_DIV
	Desired Frequency *1	3.125 – 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	OFF	CLKOS3_Enable
	Bypass	ON / OFF	OFF	OUTDIVIDER_MUXD
	Clock Divider (read only)	—	—	CLKOS3_DIV
	Desired Frequency *1	3.125 – 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)	—	—	

1. If this clock is selected as feedback, the minimum output frequency that is achievable is 10 MHz.

Figure 32. ECP5 and ECP5-5G PLL Phase Configuration Tab

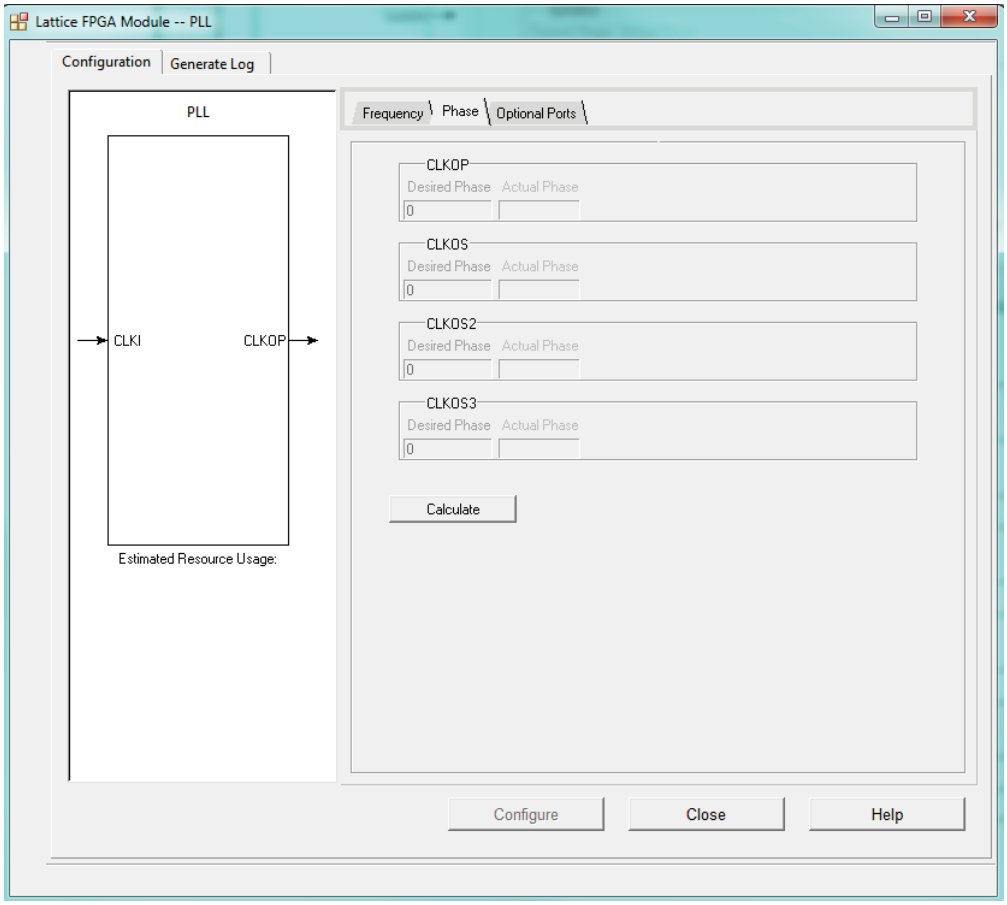


Table 19. Tab 2, PLL Phase Settings, Clarity Designer GUI

User Parameters	Description	Range	Default	Corresponding HDL Attribute
CLKOP	Desired Phase*1	(Based on Frequency)	100 MHz	CLKOP_CPHASE, CLKOP_FPHASE
	Actual Phase (read only)	—	—	
CLKOS	Desired Phase*1	(Based on Frequency)	100 MHz	CLKOS_CPHASE, CLKOS_FPHASE
	Actual Phase (read only)	—	—	
CLKOS2	Desired Phase*1	(Based on Frequency)	100 MHz	CLKOS2_CPHASE, CLKOS2_FPHASE
	Actual Phase (read only)	—	—	
CLKOS3	Desired Phase*1	(Based on Frequency)	100 MHz	CLKOS3_CPHASE, CLKOS3_FPHASE
	Actual Phase (read only)	—	—	

1. Phase is now a calculated value based on frequency parameters, which gives finer phase resolution.

Figure 33. ECP5 and ECP5-5G PLL Optional Ports Configuration Tab

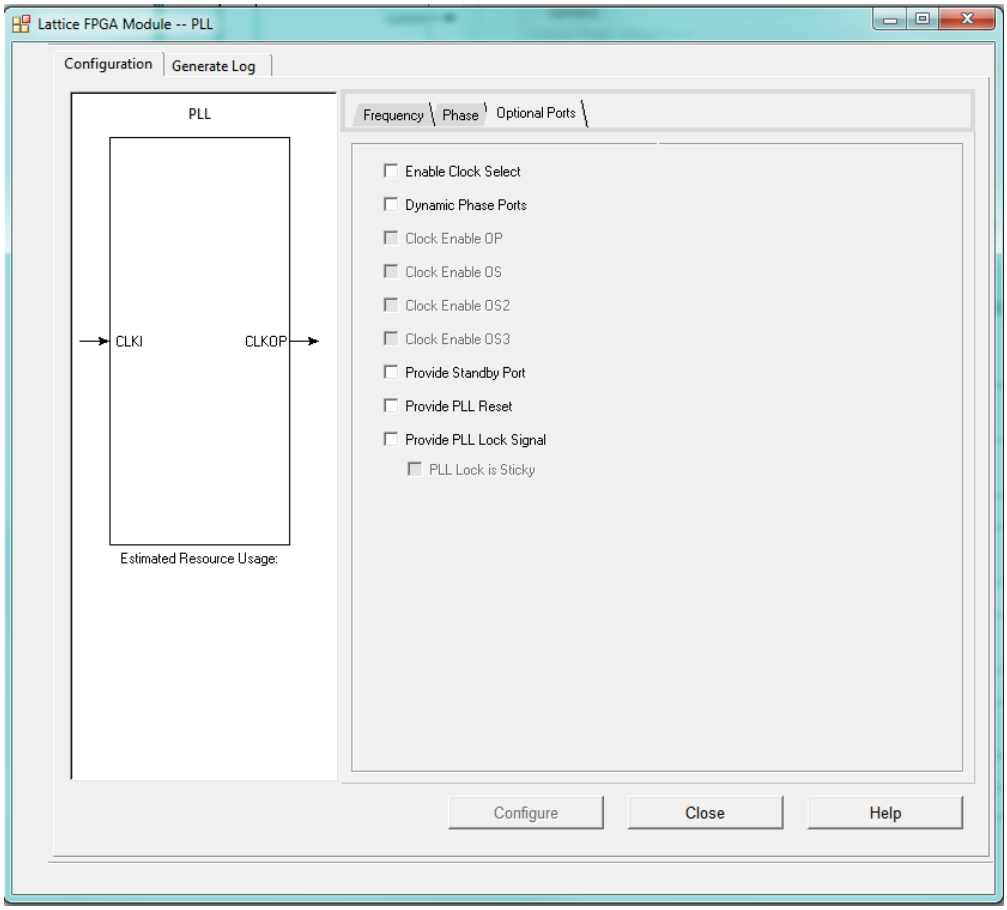


Table 20. Tab 3, PLL Optional Ports, Clarity Designer GUI

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 STDBY port to put the PLL into standby mode.	ON / OFF	OFF	STDBY_ENABLE
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	Once LOCK goes high it won't de-assert 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. 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 will be included in the preference file to automatically constrain the clocks produced by the PLL. For a step by step guide to using Clarity Designer, refer to the Clarity Designer User Manual.

PLL Reference Clock Switch Primitive (PLLREFCS)

The ECP5 and ECP5-5G 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 22 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.

Figure 34. PLLREFCS Component Symbol



Table 21. PLLREFCS Component Port Definition

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

PLLREFCS Usage in VHDL

Component 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);
PLLREFCS Usage in Verilog
Component and Attribute Declaration
```

```
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));
```

Technical Support Assistance

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

Revision History

Date	Version	Change Summary
November 2015	1.1	Added support for ECP5-5G.
		Changed document title to ECP5 and ECP5-5G sysCLOCK PLL/DLL Design and Usage Guide.
		Updated Oscillator (OSCG) section. Removed paragraph on STDBY port.
		Updated PLL Usage in Clarity Designer section. Replaced Figure 30, Clarity Designer Main Window for PLL Module.
		Updated Technical Support Assistance section.
March 2014	01.0	Initial Release

Appendix A. Primary Clock Sources and Distribution

The following figures show the inputs into the Primary Clock Network through the mid-mux into the centermux for each device. There are DCC components at the input of the centermux to allow the user to stop the clock in order to save power. All mid-mux inputs with the nomenclature "<quadrant>_pclkcib<#>" are fabric entry points to the PCLK network. All mid-mux inputs with the nomenclature "<location>_DQS_CLK" are DQS clocks from the I/O.

Figure 35. ECP5 and ECP5-5G Primary Clock Sources and Distribution, LFE5UM/LFE5UM5G-85 Devices

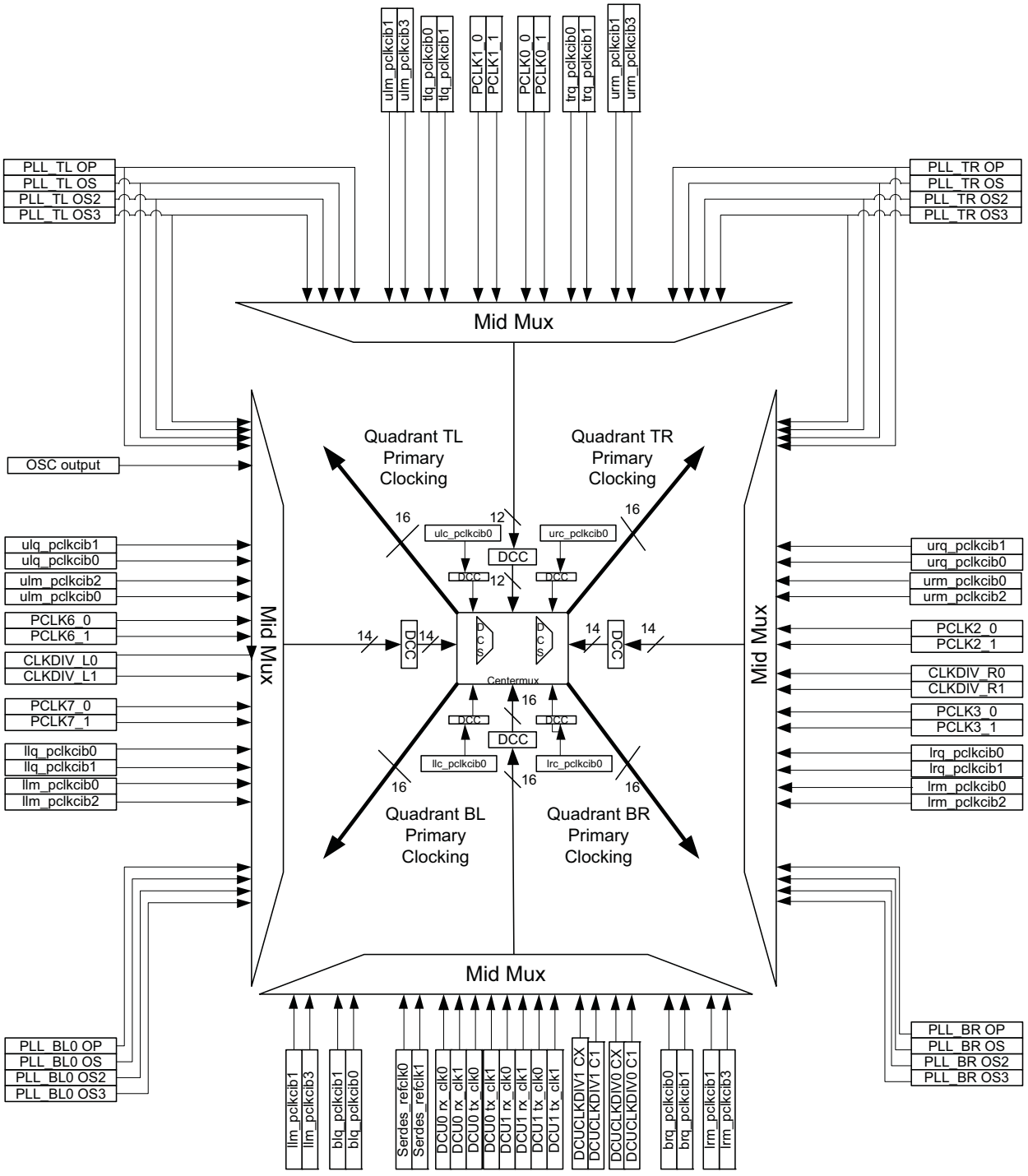


Figure 36. ECP5 and ECP5-5G Primary Clock Sources and Distribution, LFE5UM/LFE5UM5G-45 Devices

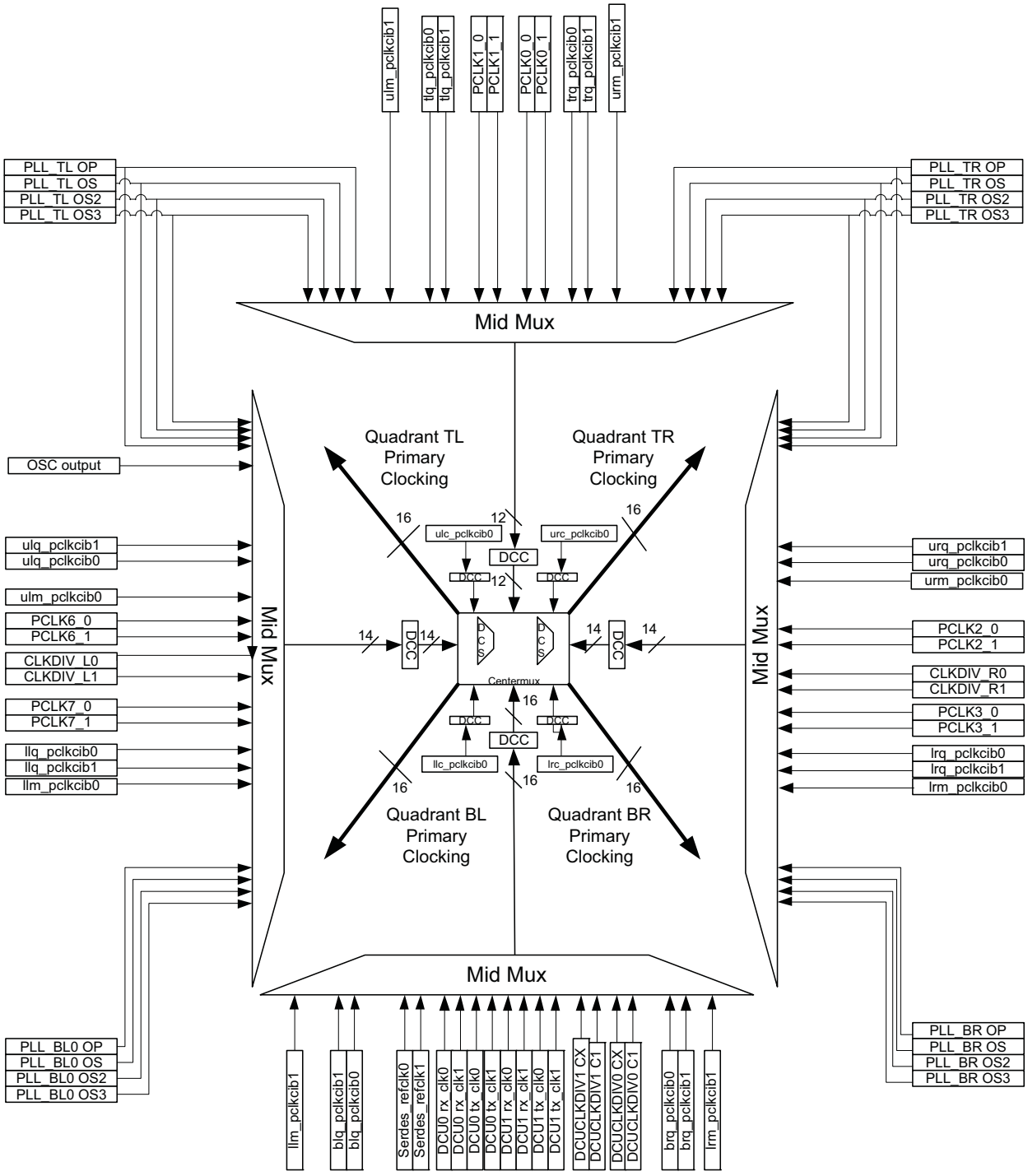
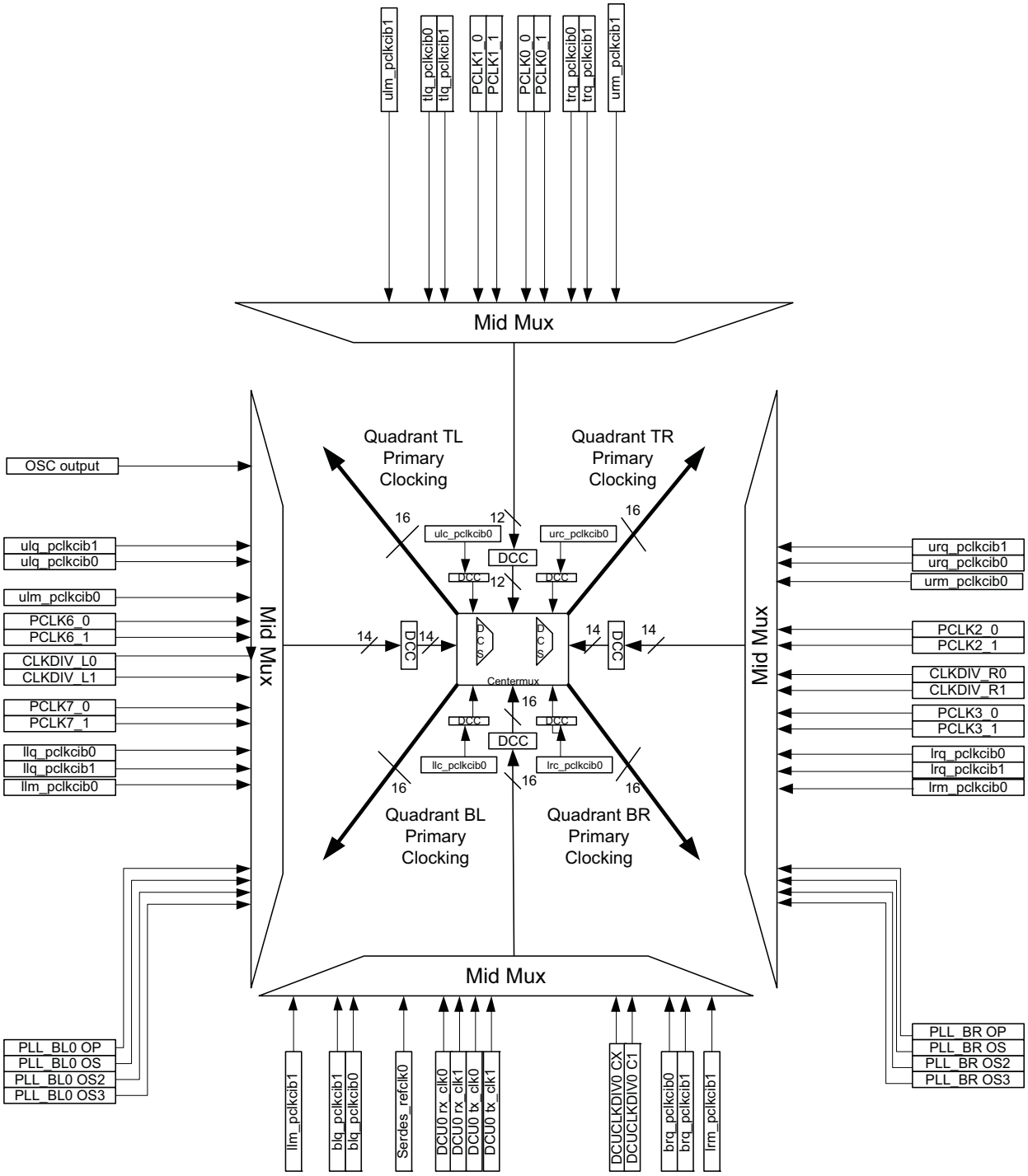


Figure 37. ECP5 and ECP5-5G Primary Clock Sources and Distribution, LFE5UM/LFE5UM5G-25 Devices



Appendix B. Pinout Rules for Clocking in ECP5 and ECP5-5G Devices

In the ECP5 and ECP5-5G device, as with all other architectures, there are general rules and guidelines for board designers which are required to be followed. These rules will give the best possible timing and allow for a successful design.

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

Primary Clock Input Pin – PCLKT<Bank>_<0/1>

Dedicated PLL Input Pin – <LOC>_GPLL0T_IN

Table 22. Clock Input Selection Table

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