LatticeXP2 sysCLOCK PLL/DLL Design and User Guide

Technical Note

FPGA-TN-02092-1.4

September 2021

Disclaimers

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

Contents

Figures

Tables

1. Introduction

This technical note describes the clock resources available in the LatticeXP2™ device architecture. Details are provided for primary clocks, secondary clocks, edge clocks, and general routing, as well as clock elements such as PLLs, DLLs, clock dividers and more.

The number of PLLs and DDR-DLLs for each device is listed in Table 1.1.

Table 1.1. Number of PLLs and DDR-DLLs

Parameter	Description	XP2-5	XP2-8	XP2-17	XP2-30	XP2-40
Number of GPLLs	General purpose PLL	2	2	4	4	4
Number of DDR-DLLs	DLL for DDR applications	2	2	2	2	2

2. Clock/Control Distribution Network

LatticeXP2 devices provide global clock distribution in the form of eight quadrant-based primary clocks and flexible secondary clocks. Two edge clocks are also provided on every edge of the device. Other clock sources include clock input pins, internal nodes, PLLs, and clock dividers.

3. LatticeXP2 Top-Level View

Figure 3.1 provides a view of the primary clocking structure of the LatticeXP2-40 device.

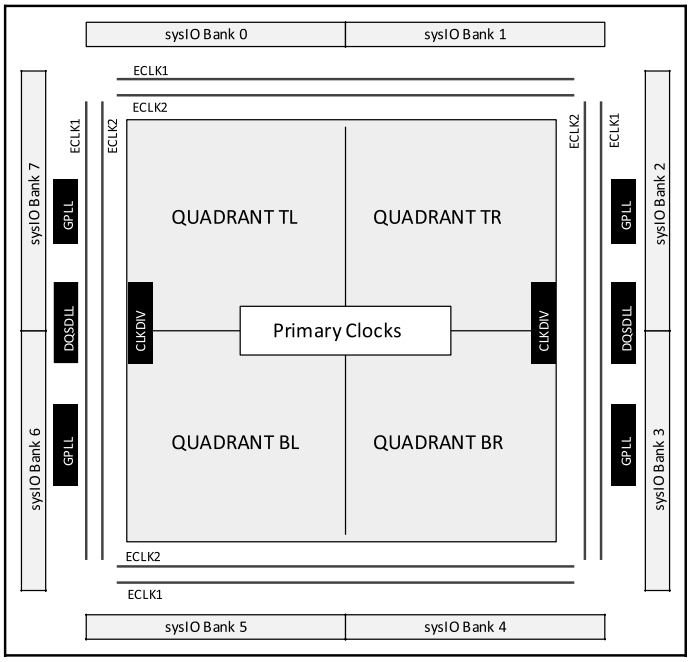

Figure 3.1. LatticeXP2 Clocking Structure (LatticeXP2-35)

4. Primary Clocks

Each quadrant receives up to eight primary clocks. Two of these clocks provide the Dynamic Clock Selection (DCS) feature. The six primary clocks without DCS can be specified in the Pre-map Preference Editor as *Primary Pure* and the two DCS clocks as *Primary-DCS*.

The sources of primary clocks are:

- PLL outputs
- CLKDIV outputs
- Dedicated clock pins
- Internal nodes

5. Secondary Clocks

The LatticeXP2 secondary clocks are a flexible region-based clocking resource. Each region can have four independent clock inputs. Since the secondary clock is a regional resource, it can cross the primary clock quadrant boundaries.

There are eight secondary clock muxes per quadrant. Each mux has inputs from four different sources. Three of these are from internal nodes. The fourth input comes from a primary clock pin. The input sources are not necessarily located in the same quadrant as the mux. This structure enables global use of secondary clocks.

The sources of secondary clocks are:

- Dedicated clock pins
- Clock Divider (CLKDIV) outputs
- Internal nodes

Table 5.1 lists the number of secondary clock regions in LatticeXP2 devices.

Table 5.1. Number of Secondary Clock Regions

Parameter	XP2-5	XP2-8	XP2-17	XP2-30	XP2-40
Number of regions	6	6	6	6	8

6. Edge Clocks

The LatticeXP2 device has two Edge Clocks (ECLK) per side. 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 fanout capability. Refer to Appendix B for detailed information on ECLK locations and connectivity.

The sources of edge clocks are:

- Left and Right Edge Clocks
  - Dedicated clock pins
  - PLL outputs
  - PLL input pins
  - Internal nodes
- Top and Bottom Edge Clocks
  - Dedicated clock pins
  - Internal nodes

Edge clocks can directly drive the secondary clock resources and general routing resources. Refer to Figure B. for detailed information on edge clock routing.

Figure 6.1 shows the structure of the secondary clocks and edge clocks for a typical device.

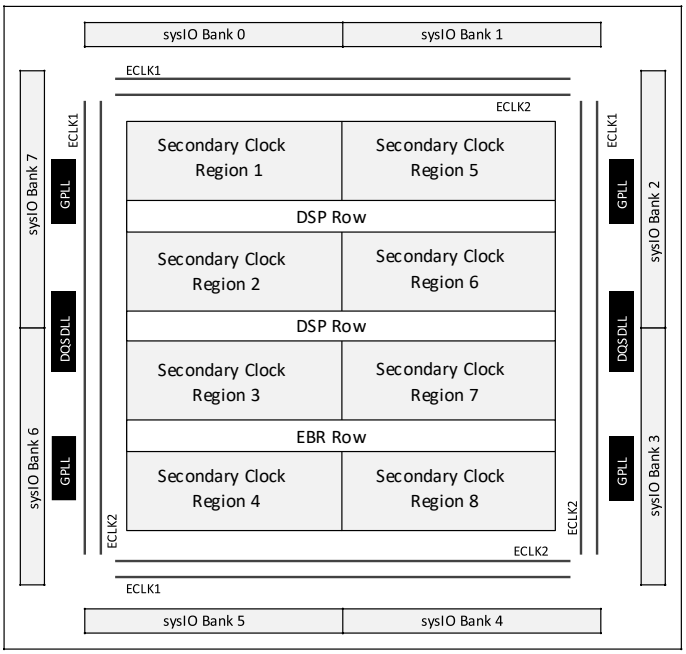

Figure 6.1. LatticeXP2 Secondary Clocks and Edge Clocks (ECP3-35)

7. Primary Clock Note

The CLKOP must be used as the feedback source to optimize PLL performance.

Most designers use PLLs for clock tree injection removal mode and the CLKOP should be assigned to a primary clock. This is done automatically by the software unless otherwise specified by the user.

CLKOP can route only to CLK0 to CLK5, while CLKOS/CLKOK can route to all primary clocks (CLK0 TO CLK7). When CLK6 or CLK7 is used as a primary clock and there is only one clock input to the DCS, the DCS is assigned as a buffer mode by the software. Please see the DCS section of this document for detailed information.

8. Specifying Clocks in the Design Tools

If desired, designers can specify the clock resources, primary, secondary or edge to be used to distribute a given clock source. Figure 9.1 illustrates how this can be done in the Pre-map Preference editor. Alternatively, the preference file can be used, as discussed in Appendix C.

8.1. Primary-Pure and Primary-DCS

Primary Clock Net can be assigned to either Primary-Pure (CLK0 to CLK5) or Primary-DCS (CLK6 and CLK7).

9. Global Primary Clock and Quadrant Primary Clock

9.1. Global Primary Clock

If a primary clock is not assigned as a quadrant clock, the software assumes it is a global clock. There are six Global Primary/Pure clocks and two Global Primary/DCS clocks available.

9.2. Quadrant Primary Clock

Any primary clock may be assigned to a quadrant clock. The clock may be assigned to a single quadrant or to two adjacent quadrants (not diagonally adjacent).

When a quadrant clock net is used, the user must ensure that the registers each clock drives can be assigned in that quadrant without any routing issues.

In the Quadrant Primary Clocking scheme, the maximum number of primary clocks is 32, as long as all the primary clock sources are available.

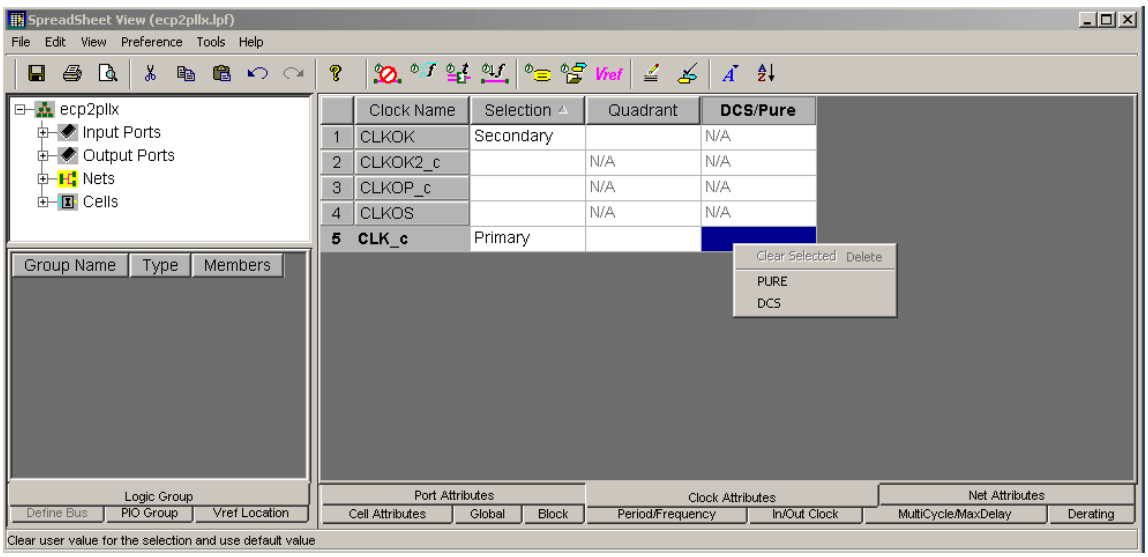

Figure 9.1. Clock Attributes in the Pre-map Preference Editor

Refer to Appendix A. Primary Clock Sources and Distribution for detailed clock network diagrams.

10. sysCLOCK™ PLL

The LatticeXP2 PLL provides features such as clock injection delay removal, frequency synthesis, phase/duty cycle adjustment, and dynamic delay adjustment. Figure 10.1 shows the block diagram of the LatticeXP2 PLL. Generally, the best way to add a PLL to a design is by using IPexpress™; the user simply provides information to the tool via a GUI, and the tool performs all calculations and design rule checks. It then generates a package that can be added to your design in the HDL language of your choice.

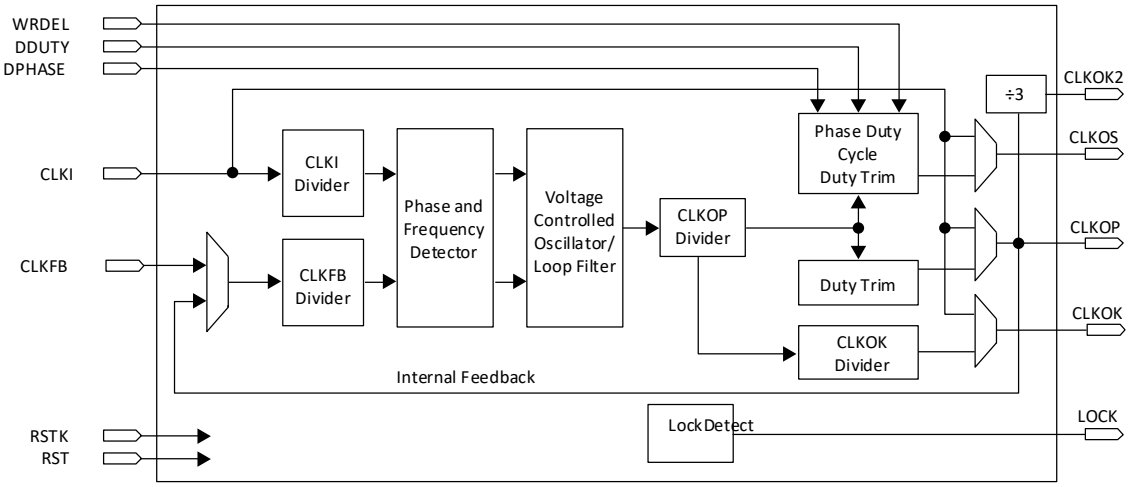

Figure 10.1. LatticeXP2 PLL Block Diagram

11. Functional Description

11.1. PLL Divider and Delay Blocks

11.1.1. Input Clock (CLKI) Divider

The CLKI divider is used to control the input clock frequency into the PLL block. The divider setting directly corresponds to the divisor of the output clock. The input and output of the input divider must be within the input and output frequency ranges specified in the device data sheet.

11.1.2. Feedback Loop (CLKFB) Divider

The CLKFB divider is used to divide the feedback signal. Effectively, this multiplies the output clock, because the divided feedback must speed up to match the input frequency into the PLL block. The PLL block increases the output frequency until the divided feedback frequency equals the input frequency. The input and output of the feedback divider must be within the input and output frequency ranges specified in the device data sheet.

11.1.3. Output Clock (CLKOP) Divider

The CLKOP divider serves the dual purposes of squaring the duty cycle of the VCO output and scaling up the VCO frequency into the 435MHz to 870MHz range to minimize jitter. The CLKOP divider values are the same whether or not the CLKOS is used.

11.1.4. CLKOK Divider

The CLKOK divider acts as a source for the global clock nets. It divides the CLKOP signal of the PLL by the value of the divider to produce lower frequency clock.

11.1.5. CLKOK2 Divider

The CLKOK2 is CLKOP divided by 3 for generating 140 MHz from 420 MHz to support SPI4.2.

11.1.6. Phase Adjustment and Duty Cycle Select (Static Mode)

Users can program CLKOS with Phase and Duty Cycle options. Phase adjustment can be done in 22.5° steps. The duty cycle resolution is 1/16th of a period except 1/16th and 15/16th duty cycle options are not supported to avoid minimum pulse violation.

11.1.7. Dynamic Phase Adjustment (DPHASE) and Dynamic Duty Cycle (DDUTY) Select

The Phase Adjustment and Duty Cycle Select can be controlled in dynamic mode. When this mode is selected, both the Phase Adjustment and Duty Cycle Select must be in dynamic mode. If only one of the features is to be used in dynamic mode, users can set the other control inputs with the fixed logic levels of their choice.

11.1.8. Duty Trim Adjustment

With the LatticeXP2 device family, the duty cycle can be fine-tuned with the Duty Trim Adjustment.

11.1.9. Fine Delay Adjust

This optional feature is controlled by the input port, WRDEL. See information on the WRDEL input in the next section of this document.

12. PLL Inputs and Outputs

12.1. 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 can be derived from a dedicated dual-purpose pin or from routing.

12.2. RST Input

The PLL reset occurs under two conditions. First, at power-up, the internal power-up reset signal from the configuration block resets the PLL. Second, the user-controlled PLL reset signal RST, provided as part of the PLL module, can be driven by an internally generated reset function or an external pin. This RST signal resets all internal PLL counters, flip-flops (including the M, N, V, K and CLKOK2 Dividers) and the charge pump. When RST goes inactive, the PLL will start the lock-in process, and will take t_{LOCK} time to complete the PLL lock. Figure 12.1 shows the timing diagram of the RST input.

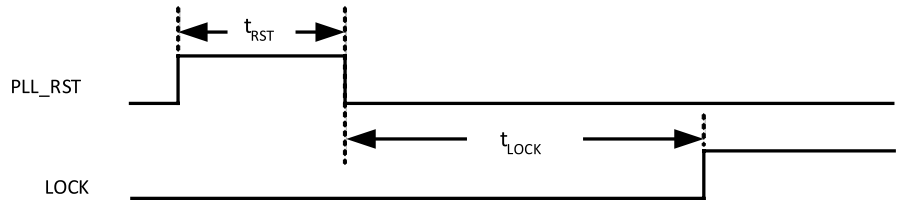

Figure 12.1. RST Input Timing Diagram

12.3. RSTK Input

RSTK is the reset input for the K-Divider. This K-Divider reset is used to synchronize the K-Divider output clock to the input clock. LatticeXP2 has an optional gearbox in the I/O cell for both outputs and inputs. The K-Divider reset is useful for the gearbox implementation. RSTK is active high.

12.4. CLKFB Input

The feedback signal to the PLL, which is fed through the feedback divider, can be derived from the Primary Clock net (CLKOP), a preferred pin, directly from the CLKOP divider or from general routing. External feedback allows the designer to compensate for board-level clock alignment.

12.5. CLKOP Output

The sysCLOCK PLL main clock output, CLKOP, is a signal available for selection as a primary clock. This clock signal is available at the CLK_OUT pin.

12.6. CLKOS Output with Phase and Duty Cycle Select

The sysCLOCK PLL auxiliary clock output, CLKOS, is a signal available for selection as a primary clock. The CLKOS is used when phase shift and/or duty cycle adjustment is desired. The programmable phase shift allows for different phase in increments of 22.5°. The duty select feature provides duty select in 1/16th of the clock period.

This feature is also supported in Dynamic Control Mode.

12.7. CLKOK Output with Lower Frequency

The CLKOK is used when a lower frequency is desired. It is a signal available for selection as a primary clock.

12.8. CLKOK2 Output

This extra clock is provided for SPI4.2 application. The 420 MHz CLKOP clock is divided by 3, producing 140 MHz.

The clock can also be used for any applications where CLKOP-divided-by-3 is required.

12.9. 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 the specified lock time. Once lock is achieved, the PLL lock signal will be asserted. If, during operation, the input clock or feedback signals to the PLL become invalid, the PLL will lose lock. However, when the input clock completely stops, the LOCK output will remain in its last state, since it is internally registered by this clock. It is recommended to assert PLL RST to re-synchronize the PLL to the reference clock. The LOCK signal is available to the FPGA routing to implement generation of RST. ModelSim® simulation models take two to four reference clock cycles from RST release to LOCK high.

12.10. Dynamic Phase and Dynamic Duty Cycle Adjustment

The DPHASE[3:0] port is used with the Dynamic Phase Adjustment feature to allow the user to connect a control signal to the PLL. The DDUTY[3:0] port is used with the Dynamic Duty Adjustment feature to allow the user to connect a control signal to the PLL. The DPHASE and DDUTY ports are listed in Table 12.1.

The Dynamic Phase and Dynamic Duty Cycle Adjustment features will be discussed in more detail in later sections of this document.

Table 12.1. Dynamic Phase and Duty Cycle Adjust Ports

Port Name	I/O	Description
DPHASE[3:0]	I	Dynamic Phase Adjust inputs
DDUTY[3:0]	I	Dynamic Duty Cycle Adjust inputs

12.11. WRDEL (Write Delay)

The fine delay option supports SPI4.2. The PLL has a coarse phase adjust feature in which a cycle is divided into 16 equal steps (22.5°). For SPI4.2 running at 840 Mbps, the clock frequency is 420 MHz. At this frequency, the period is roughly 150 ps. This is slightly too coarse for dynamic phase adjust requirements. It may be more effective to use increments half as large. Combined with coarse phase adjust, a 70 ps (nominal) step provides effectively 32 steps of phase adjustment. This fine delay only applies to CLKOS (just like the coarse phase adjust). This is convenient since it allows one GPLL to be used for both read and write (where read uses CLKOS and write uses CLKOP).

13. PLL Attributes

The PLL utilizes several attributes that allow the configuration of the PLL through source constraints and a preference file. The following section details these attributes and their usage.

13.1. FIN

The input frequency can be any value within the specified frequency range based on the divider settings.

13.2. CLKI_DIV, CLKFB_DIV, CLKOP_DIV, CLKOK_DIV

These dividers determine the output frequencies of each output clock. The user is not allowed to input an invalid combination. Valid combinations are determined by the input frequency, the dividers, and the PLL specifications.

Note: Unlike PLLs in the LatticeECP™, LatticeEC™, LatticeXP™ and MachXO™ devices, the CLKOP divider values are the same whether or not CLKOS is used.

The CLKOP_DIV value is calculated to maximize the fVCO within the specified range based on FIN and CLKOP_FREQ in conjunction with the CLKI_DIV and CLKFB_DIV values. These value settings are designed such that the output clock duty cycle is as close to 50% as possible. Table 13.1 shows the possible divider ranges.

Table 13.1. Divider Range

Attribute	Name	Value	Default
CLKI Divider Setting	CLKI_DIV	1 to 43	1
CLKFB Divider Setting	CLKFB_DIV	1 to 43	1
CLKOP Divider Setting	CLKOP_DIV	2, 4, 8, 16, 32, 48, 64, 80	8
CLKOK Divider Setting	CLKOK_DIV	2, 4, 6,.., 126, 128	2

13.3. FREQUENCY_PIN_CLKI, FREQUENCY_PIN_CLKOP, FREQUENCY_PIN_CLKOK

These input and output clock frequencies determine the divider values.

13.4. CLKOP Frequency Tolerance

When the desired output frequency is not achievable, users may enter the frequency tolerance of the clock out-put.

13.4.1. PHASEADJ (Phase Shift Adjust)

The PHASEADJ attribute is used to select Phase Shift for CLKOS output. The phase adjustment is programmable in 22.5° increments.

13.4.2. DUTY (Duty Cycle)

The DUTY attribute is used to select the Duty Cycle for CLKOS output. The Duty Cycle is programmable at 1/16th of the period increment. Steps 2 to 14 are supported. 1/16th and 15/16th duty cycles are not supported to avoid the minimum pulse width violation.

13.4.3. FB_MODE

There are three sources of feedback signals that can drive the CLKFB Divider: Internal, CLKOP (Clock Tree) and user clock. CLKOP (Clock Tree) feedback is used by default. Internal feedback takes the CLKOP output at CLKOP Divider output before the Clock Tree to minimize the feedback path delay. The user clock feedback is driven from the dedicated pin, clock pin or user-specified internal logic.

DUTY_TRIM Adjustment (Dynamic mode only) Users can fine tune the duty cycle of CLKOP and/or CLKOS with the DUTY_TRIM feature when Dynamic

13.4.4. PHASE/DUTY Adjustment is selected.

- TRIM Polarity Select: Users can select either rising edge or falling edge of clock to trim.
- TRIM Delay of CLKOP can be set to 0 to 7 steps of unit trim delay.
- TRIM Delay of CLKOS can be set to 0 to 3 steps of unit trim delay.

13.4.5. CLKOS/CLKOK/CLKOK2 Select

Users select these output clocks only when they are used in the design.

13.4.6. CLKOP/CLKOS/CLKOK BYPASS

These bypasses are enabled if set. CLKI is directly routed to its corresponding output clock.

13.4.7. RST/RSTK Select

Users may select these reset signals only when they are used in the design.

14. LatticeXP2 PLL Primitive Definition

One PLL primitive is used for LatticeXP2 PLL implementation. Figure 14.1 shows the LatticeXP2 PLL primitive library symbols.

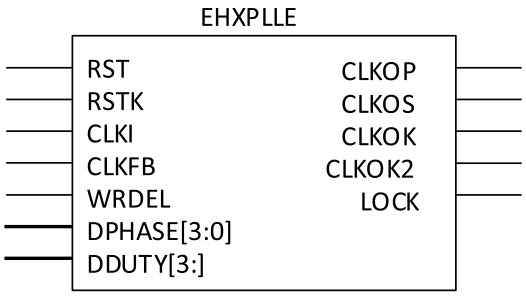

Figure 14.1. LatticeXP2 PLL Primitive Symbol

14.1. EPLLD Design Migration from LatticeECP2 to LatticeXP2

The EPLLD generated for LatticeECP2 can be used with minor changes. If the configuration does not include Dynamic Phase and Duty Options, the migration is fully supported. If Dynamic Phase and Duty Options are included, the user must tie the DPAMODE port to ground.

14.2. Dynamic Phase/Duty Mode

This mode sets both Dynamic Phase Adjustment and Dynamic Duty Select at the same time. There are two modes, *Dynamic Phase and 50% Duty* and *Dynamic Phase and Dynamic Duty*.

14.2.1. Dynamic Phase and 50% Duty

This mode allows users to set up Dynamic Phase inputs only. The 50% Duty Cycle is handled internally by the ispLEVER® software. The DDUTY[3:0] ports are user-transparent in this mode.

14.2.2. Dynamic Phase and Dynamic Duty

This mode allows designers to use both DDPHASE[3:0] and DDUTY[3:0] ports to input dynamic values.

To use Dynamic Phase Adjustment with a fixed duty cycle other than a 50%, simply set the DDUTY[3:0] inputs to the desired duty cycle value. Figure 14.2 illustrates an example circuit.

For example: Assume a design uses dynamic phase adjustment and a fixed duty cycle select and the desired duty cycle in 3/16th of a period. The setup should be as shown in Figure 14.2.

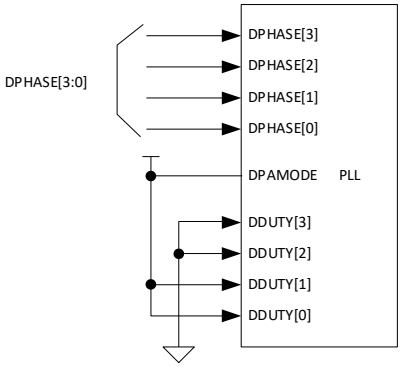

Figure 14.2. Example of Dynamic Phase Adjustment with a Fixed Duty Cycle of 3/16th of a Period

15. Dynamic Phase Adjustment/Duty Cycle Select

Phase Adjustment settings are described in Table 15.1.

Table 15.1. Phase Adjustment Settings

DPHASE[3:0]	Phase (°)
0000	0
0001	22.5
0010	45
0011	67.5
0100	90
0101	112.5
0110	135
0111	157.5
1000	180
1001	202.5
1010	225
1011	247.5
1100	270
1101	292.5
1110	315
1111	337.5

Duty Cycle Select settings are described in Table 15.2.

Table 15.2. Duty Cycle Select Settings

DDUTY[3:0]	Duty Cycle (1/16th of a Period)	Comment
0000	0	Not Supported
0001	1	Not Supported
0010	2	
0011	3	
0100	4	
0101	5	
0110	6	
0111	7	
1000	8	
1001	9	
1010	10	
1011	11	
1100	12	
1101	13	
1110	14	
1111	15	Not Supported

Note: PHASE/DUTY_CTNL is selected in the GUI ‘PLL Phase & Duty Options’ box and if it is set to ‘Dynamic Mode’, then both DPHASE[3:0] and DDUTY[3:0] inputs must be provided. If one of these inputs is a fixed value, the inputs must be tied to the desired fixed logic levels.

16. PLL Usage in IPexpress

IPexpress is used to create and configure a PLL. The graphical user interface is used to select parameters for the PLL. The result is an HDL model to be used in the simulation and synthesis flow.

Figure 16.1 shows the main window when PLL is selected. The only entry required in this window is the module name. Other entries are set to the project settings. Users may change these entries, if desired. After entering the module name of choice, clicking on Customize will open the Configuration Tab window as shown in Figure 16.2.

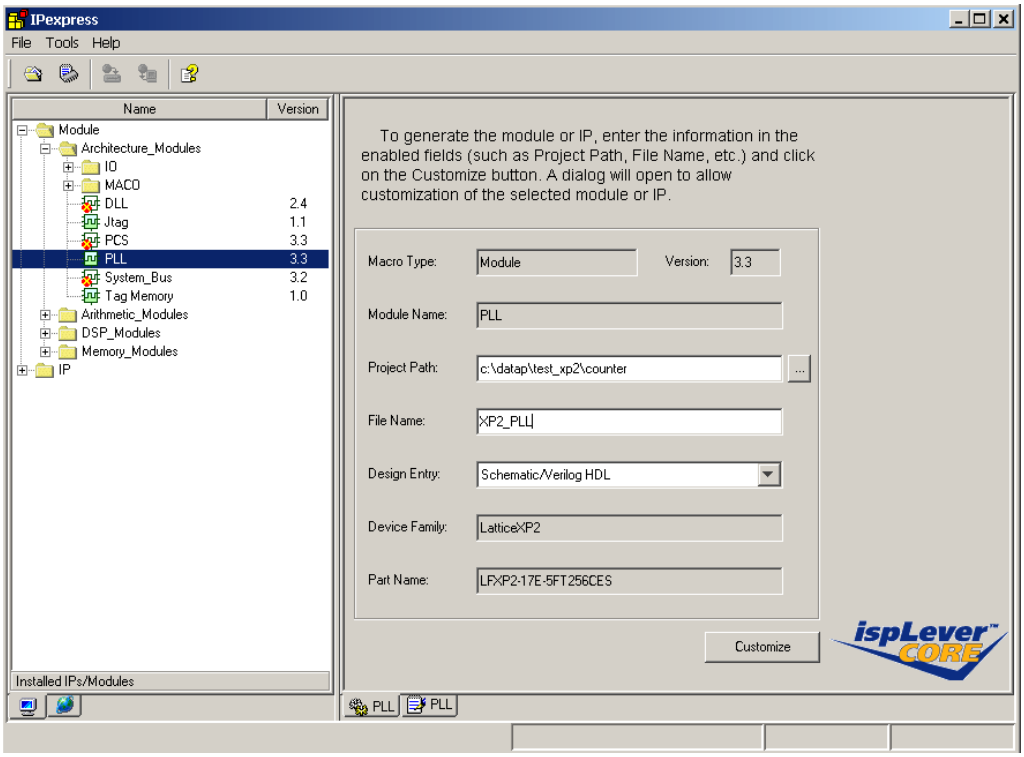

Figure 16.1. IPexpress Main Window

16.1. Configuration Tab

The Configuration Tab lists all user accessible attributes with default values set. Upon completion, clicking Generate will generate source and constraint files. Users may choose to use the *.LPC file (for ispLEVER, or *.IPX file for Diamond) to load parameters.

Configuration Modes

There are two modes that can be used to configure the PLL in the Configuration Tab, Frequency Mode and Divider Mode.

Frequency Mode: In this mode, the user enters input and output clock frequencies and the software calculates the divider settings. If the output frequency entered is not achievable the nearest frequency will be displayed in the 'Actual' text box. After input and output frequencies are entered, clicking the **Calculate** button will display the divider values.

Divider Mode: In this mode, the user sets the divider settings with input frequency. Users must choose the CLKOP Divider value to maximize the fVCO and achieve optimum PLL performance. After input frequency and divider settings are set, clicking the Calculate button will display the frequencies. Figure 16.2 shows the Configuration Tab.

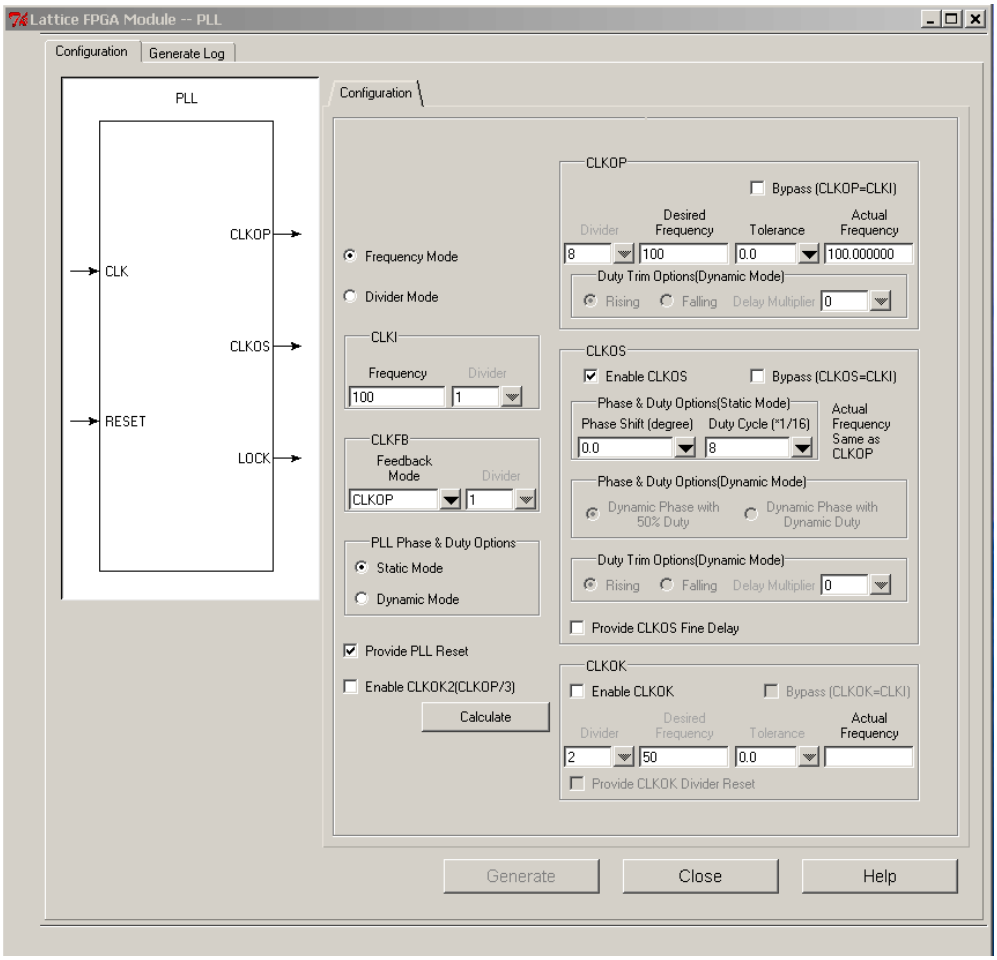

Figure 16.2. LatticeXP2 PLL Configuration Tab

Table 16.1 describes the user parameters in the IPexpress user interface and their usage.

Table 16.1. User Parameters in the IPexpress GUI

User Parameters		Description	Range	Default
Frequency Mode		User desired CLKI and CLKOP frequency	ON/OFF	ON
Divider Mode		User desired CLKI frequency and dividers settings	ON/OFF	OFF
CLKI	Frequency	Input Clock frequency	10 MHz to 435 MHz	100 MHz
	Divider	Input Clock Divider Setting (Divider Mode)	1 to 43	1
CLKFB	Feedback Mode	Feedback Mode	Internal, CLKOP, User Clock	CLKOP
	Divider	Feedback Clock Divider Setting (Divider Mode)	1 to 43	1
PLL Phase & Duty Options	None	No Phase & Duty Options	ON/OFF	ON
	Static Mode	CLKOS Phase/Duty in Static Mode	ON/OFF	OFF
	Dynamic Mode	CLKOS Dynamic Mode Phase/Duty Setting	ON/OFF	OFF
		CLKOS Duty Trimming	ON/OFF	OFF
		CLKOP Duty Trimming	ON/OFF	OFF

User Parameters		Description	Range	Default
CLKOP	Bypass	Bypass PLL: CLKOP = CLKI	ON/OFF	OFF
	Desired Frequency	User enters desired CLKOP frequency	10 MHz to 435 MHz	100 MHz
	Divider	CLKOP Divider Setting (Divider Mode)	2, 4, 8, 16, 32, 48, 64, 80	8
	Tolerance	CLKOP tolerance users can tolerate	0.0, 0.1, 0.2, 0.5, 0.1, 0.2, 0.5, 1.0	0.0
	Actual Frequency	Actual frequency achievable. Read only	—	—
	Rising	Rising Edge Trim	ON/OFF	OFF
	Falling	Falling Edge Trim	ON/OFF	OFF
	Delay Multiplier	Number of delay steps	0 to 7	0
CLKOS	Enable	Enable CLKOS output clock	ON/OFF	OFF
	Bypass	Bypass PLL: CLKOS = CLKI	ON/OFF	OFF
	Phase Shift	CLKOS Static Phase Shift	0°, 22.5°, 45°..337.5°	0°
	Rising	Rising Edge Trim	ON/OFF	OFF
	Delay Multiplier	Number of Delay steps	0 to 7	0
CLKOK	Enable	Enable CLKOS output clock	ON/OFF	OFF
	Bypass	Bypass PLL: CLKOK = CLKI	ON/OFF	OFF
	Frequency	User enters desired CLKOK frequency	78.125 kHz to 217.5 MHz	50 MHz
	Divider	CLKOK Divider Setting	2 to 128	2
	Tolerance	CLKOK tolerance users can tolerate	0.0, 0.1, 0.2, 0.5, 0.1, 0.2, 0.5, 1.0	0.00
	Actual Frequency	Actual frequency achievable. Read only	—	—
CLKOK2	Enable	Enable CLKOK2 output clock	ON/OFF	OFF
Provide PLL Reset		Provide PLL Reset Port (RESET)	ON/OFF	OFF
Provide CLKOK Divide Reset		Provide CLKOK Reset Port (RSTK)	ON/OFF	OFF
Provide CLKOS Fine Delay Port		Provide CLKOS Fine Delay Port (WRDEL)	ON/OFF	OFF
Import LPC to ispLEVER Project		Import .lpc file to ispLEVER project	ON/OFF	OFF

17. PLL Modes of Operation

PLLs have many uses within a logic design. The two most popular are Clock Injection Removal and Clock Phase Adjustment. These two modes of operation are described below.

17.1. PLL Clock Injection Removal

In this mode the PLL is used to reduce clock injection delay. Clock injection delay is the delay from the input pin of the device to a destination element such as a flip-flop. The phase detector of the PLL aligns the CLKI with CLKFB. If the CLKFB signal comes from the clock tree (CLKOP), then the PLL delay and the clock tree delay is removed. Figure 17.1. Illustrates an example block diagram and waveform.

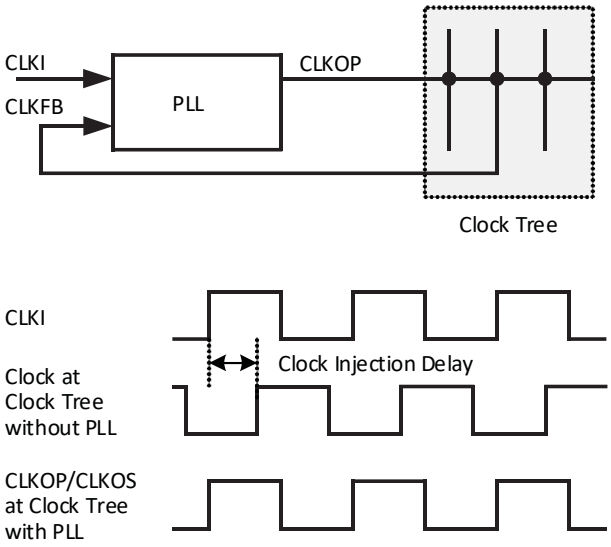

Figure 17.1. Clock Injection Delay Removal Application

17.2. PLL Clock Phase Adjustment

Refer to Figure 17.2. In this mode the PLL is used to create fixed phase relationships in 22.5° increments. Creating fixed phase relationships is useful for forward clock interfaces where a specific relationship between clock and data is required.

The fixed phase relationship can be used between CLKI and CLKOS or between CLKOP and CLKOS.

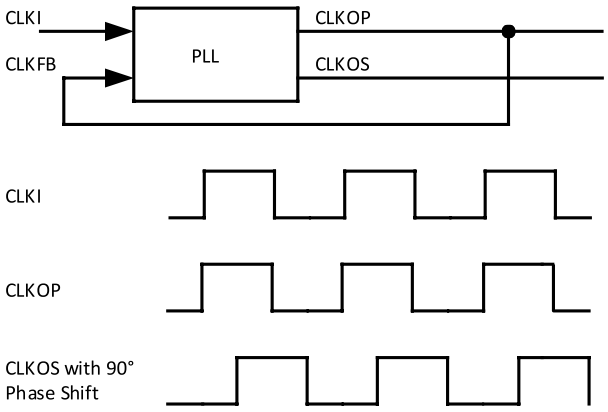

Figure 17.2. CLKOS Phase Adjustment from CLKOP

18. IPexpress Output

There are two IPexpress outputs that are important for use in the design. The first is the <module_name>.[v|vhd] file. This is the user-named module that was generated by the tool to be used in both synthesis and simulation flows. The second is a template file, <module_name>_tmpl.[v|vhd]. This file contains a sample instantiation of the module. This file is provided for the user to copy/paste the instance and is not intended to be used in the synthesis or simulation flows directly.

For the PLL, IPexpress 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 regenerating the package in IPexpress 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 from the PLL.

19. Use of the Pre-Map Preference Editor

Clock preferences can be set in the Pre-Map Preference Editor. Figure 19.1 shows an example screen shot. The Pre-Map Preference Editor is a part of the ispLEVER Design Planner.

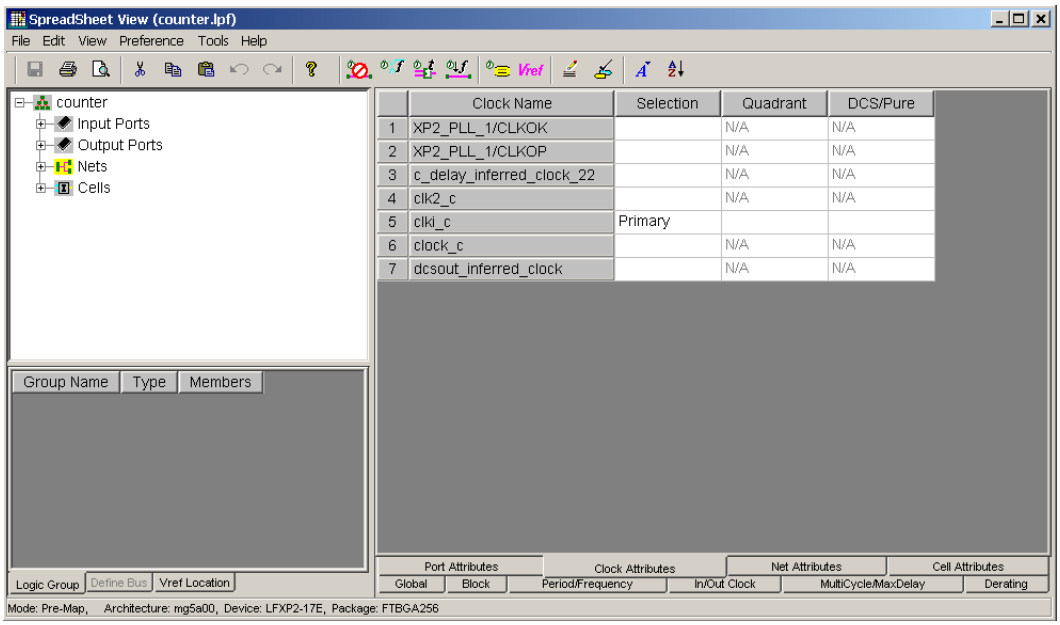

Figure 19.1. Pre-Map Preference Editor Example

20. Clock Dividers (CLKDIV)

The clock divider divides the high-speed clock by 1, 2, 4 or 8. All the outputs have matched input to output delay. CLKDIV can take as its input the edge clocks and the CLKOP of the PLL. The divided outputs drive the primary clock and are also available for general routing or secondary clocks. The clock dividers are used for providing the low speed FPGA clocks for shift registers (x2, x4, x8) and DDR/SPI4 I/O logic interfaces.

20.1. CLKDIV Primitive Definition

Users can instantiate CLKDIV in the source code as defined in this section. Figure 20.1, Table 20.1, and Table 20.2 describe the CLKDIVB definitions.

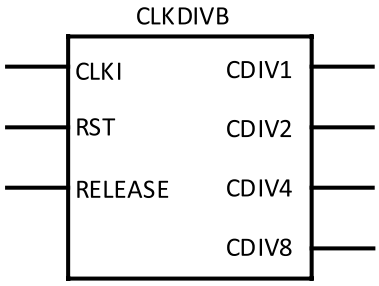

Figure 20.1. CLKDIV Primitive Symbol

Table 20.1. CLKDIVB Port Definition

Name	Description
CLKI	Clock Input
RST	Reset Input, asynchronously forces all outputs low.
RELEASE	Releases outputs synchronously to input clock.
CDIV1	Divided BY 1 Output
CDIV2	Divided BY 2 Output
CDIV4	Divided BY 4 Output
CDIV8	Divided BY 8 Output

Table 20.2. CLKDIVB Attribute Definition

Name	Description	Value	Default
GSR	GSR Enable	ENABLED/DISABLED	DISABLED

20.2. CLKDIV Declaration in VHDL Source Code

```
COMPONENT CLKDIVB
-- synthesis translate_off
      GENERIC (
            GSR   : in String);
-- synthesis translate_on
      PORT (
            CLKI,RST, RELEASE             :IN    std_logic;
            CDIV1, CDIV2, CDIV4, CDIV8    :OUT   std_logic);
END COMPONENT;

      attribute GSR : string;
      attribute GSR of CLKDIVinst0 : label is "DISABLED";

begin

CLKDIVinst0:      CLKDIVB
-- synthesis translate_off
      GENERIC MAP(
      GSR   => "disabled"
      );
-- synthesis translate_on
      PORT MAP(
            CLKI                    => CLKIsig,
      RST   => RSTsig,
      RELEASE     => RELEASEsig,
            CDIV1       => CDIV1sig,
      CDIV2       => CDIV2sig,
      CDIV4       => CDIV4sig,
      CDIV8       => CDIV8sig
      );
```

20.3. CLKDIV Usage with Verilog - Example

```
module clkdiv_top(RST,CLKI,RELEASE,CDIV1,CDIV2,CDIV4,CDIV8);

input CLKI,RST,RELEASE;
output CDIV1,CDIV2,CDIV4,CDIV8;

CLKDIVB CLKDIBinst0 (.RST(RST),.CLKI(CLKI),.RELEASE(RELEASE),
                .CDIV1(CDIV1),.CDIV2(CDIV2),.CDIV4(CDIV4),.CDIV8(CDIV8));

defparam CLKDIBint0.GSR = "DISABLED";

endmodule
```

20.4. CLKDIV Example Circuits

The clock divider (CLKDIV) can divide a clock by 2 or 4 and drives a primary clock network. Clock dividers are useful for providing the low speed FPGA clocks for I/O shift registers (x2, x4) and DDR (x2, x4) I/O logic inter-faces. Divide by 8 is provided for slow speed/low power operation.

To guarantee a synchronous transfer in the I/O logic the CLKDIV input clock must come from an edge clock and the output drive from a primary clock. In this case, they are phase matched.

It is especially useful to synchronously reset the I/O logic when Mux/DeMux gearing is used in order to synchronize the entire data bus as shown in Figure 20.2. Using the low skew characteristics of the edge clock routing a reset can be provided to all bits of the data bus to synchronize the Mux/DeMux gearing.

The second circuit shows that a DLL can replace CLKDIV for x2 and x4 applications.

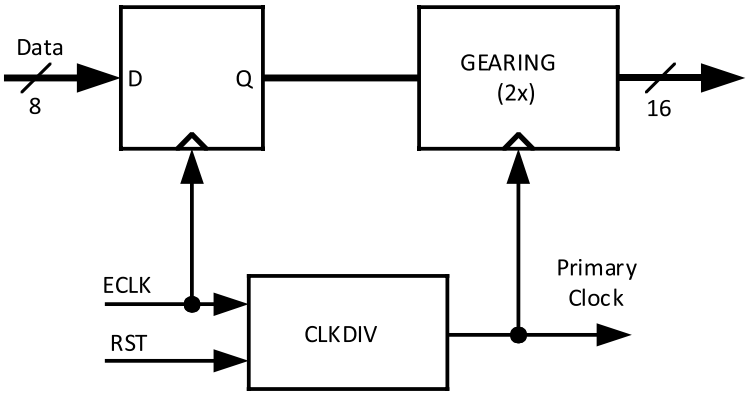

Figure 20.2. CLKDIV Application Example

20.5. Reset Behavior

Figure 20.3 illustrates the asynchronous RST behavior.

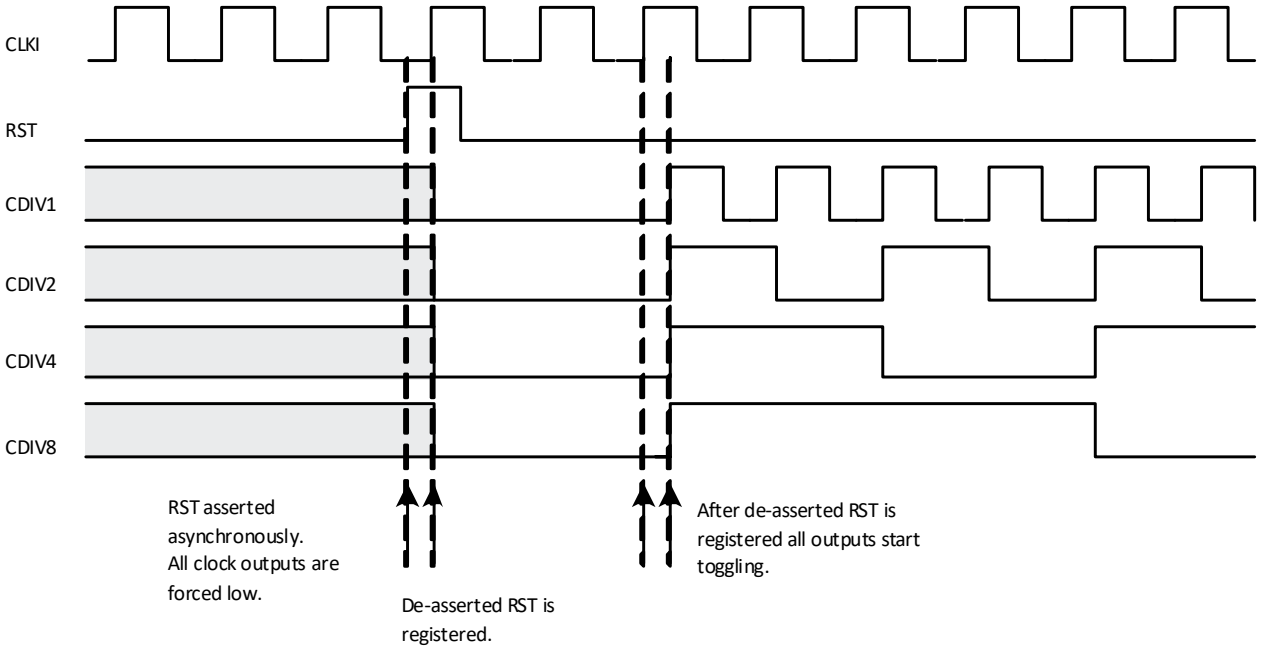

Figure 20.3. CLKDIV Reset Behavior

20.6. Release Behavior

The port, *Release* is used to synchronize the all outputs after RST is de-asserted. Figure 20.4 Illustrates the release behavior.

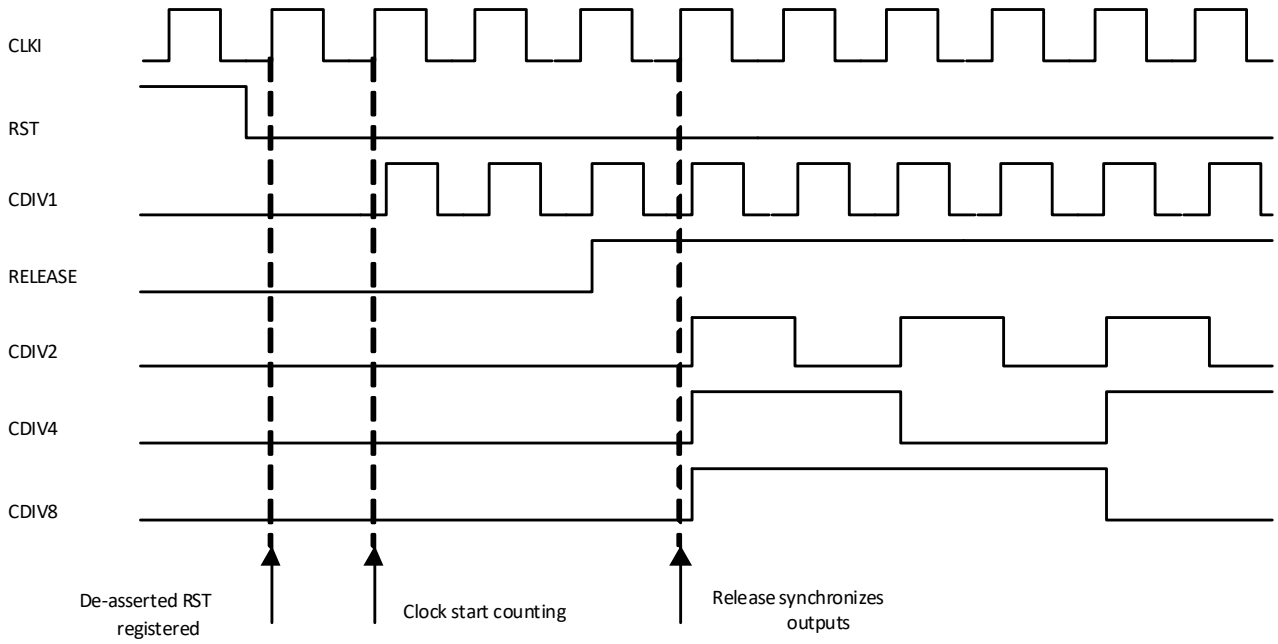

Figure 20.4. CLKDIV Release Behavior

20.7. CLKDIV Inputs-to-Outputs Delay Matching

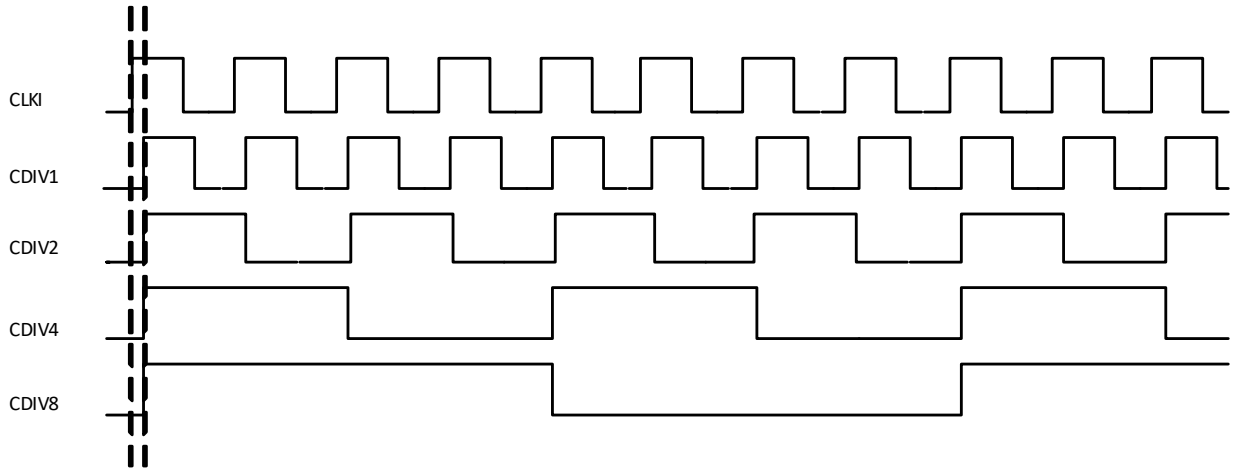

Figure 20.5. CLKDIV Inputs-to-Outputs Delay Matching

21. DCS (Dynamic Clock Select)

DCS is a global clock buffer incorporating a smart multiplexer function that takes two independent input clock sources and avoids glitches or runt pulses on the output clock, regardless of where the enable signal is toggled. There are two DCSs for each quadrant.

The outputs of the DCS then reach primary clock distribution via the feedlines. Figure 21.1 shows the block diagram of the DCS.

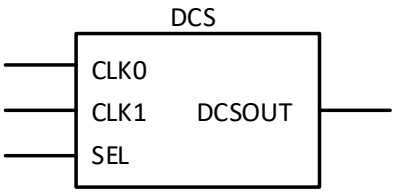

Figure 21.1. DCS Primitive Symbol

21.1. DCS Primitive Definition

Table 21.1 defines the I/O ports of the DCS block. There are eight modes to select from. Table 21.2 describes how each mode is configured.

Table 21.1. DCS I/O Definition

I/O	Name	Description
Input	SEL	Input Clock Select
	CLK0	Clock input 0
	CLK1	Clock Input 1
Output	DCSOUT	Clock Output

Table 21.2. DCS Modes of Operation

Attribute Name	Description	Output		Value
		SEL=0	SEL=1	
DCS MODE	Rising edge triggered, latched state is high	CLK0	CLK1	POS
	Falling edge triggered, latched state is low	CLK0	CLK1	NEG
	Sel is active high, Disabled output is low	0	CLK1	HIGH_LOW
	Sel is active high, Disabled output is high	1	CLK1	HIGH_HIGH
	Sel is active low, Disabled output is low	CLK0	0	LOW_LOW
	Sel is active low, Disabled output is high	CLK0	1	LOW_HIGH
	Buffer for CLK0	CLK0	CLK0	CLK0
	Buffer for CLK1	CLK1	CLK1	CLK1

21.2. DCS Timing Diagrams

Each mode performs a unique operation. The clock output timing is determined by input clocks and the edge of the SEL signal. Figure 21.2 describes the timing of each mode.

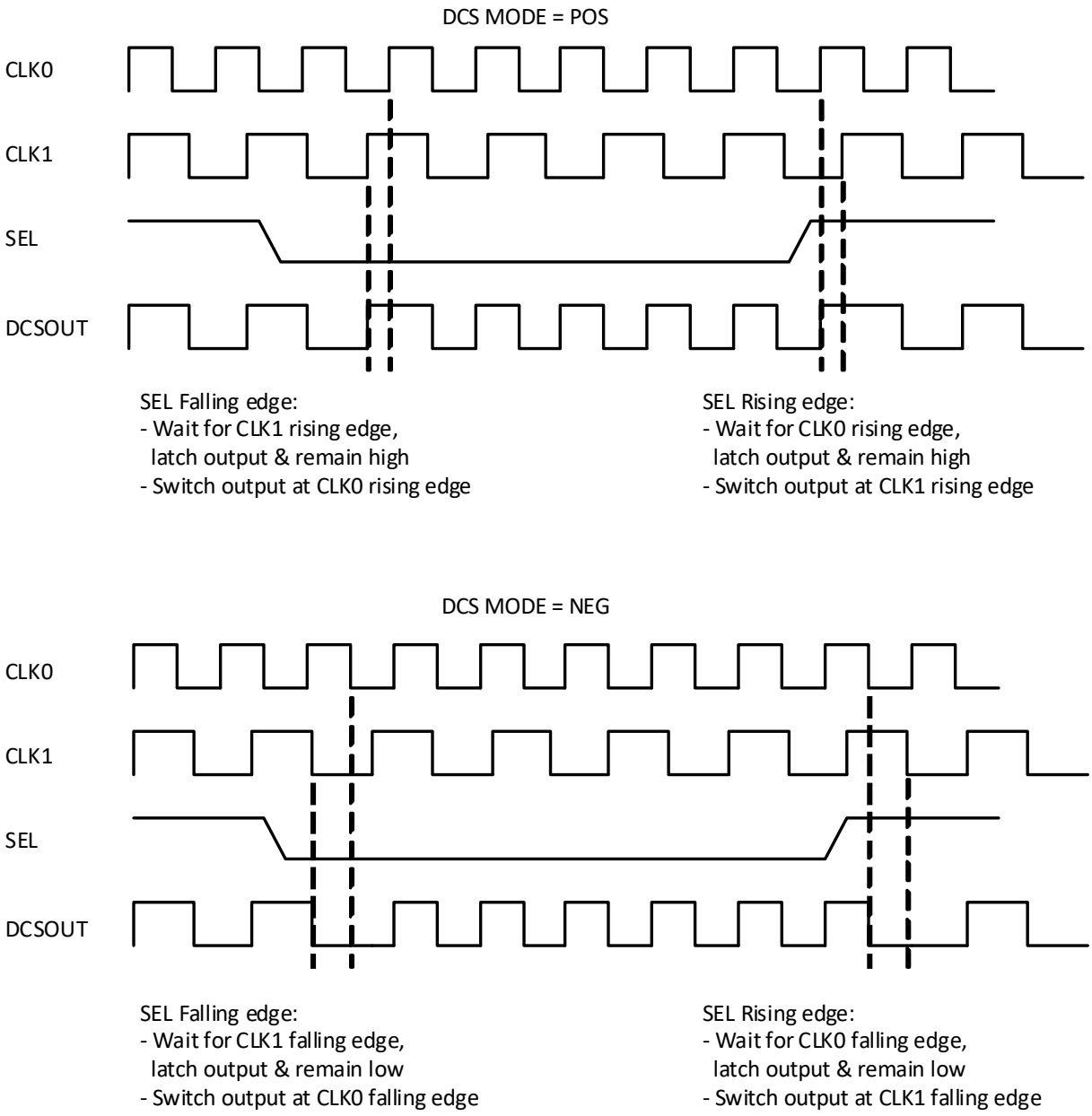

Figure 21.2. Timing Diagrams by DCS MODE

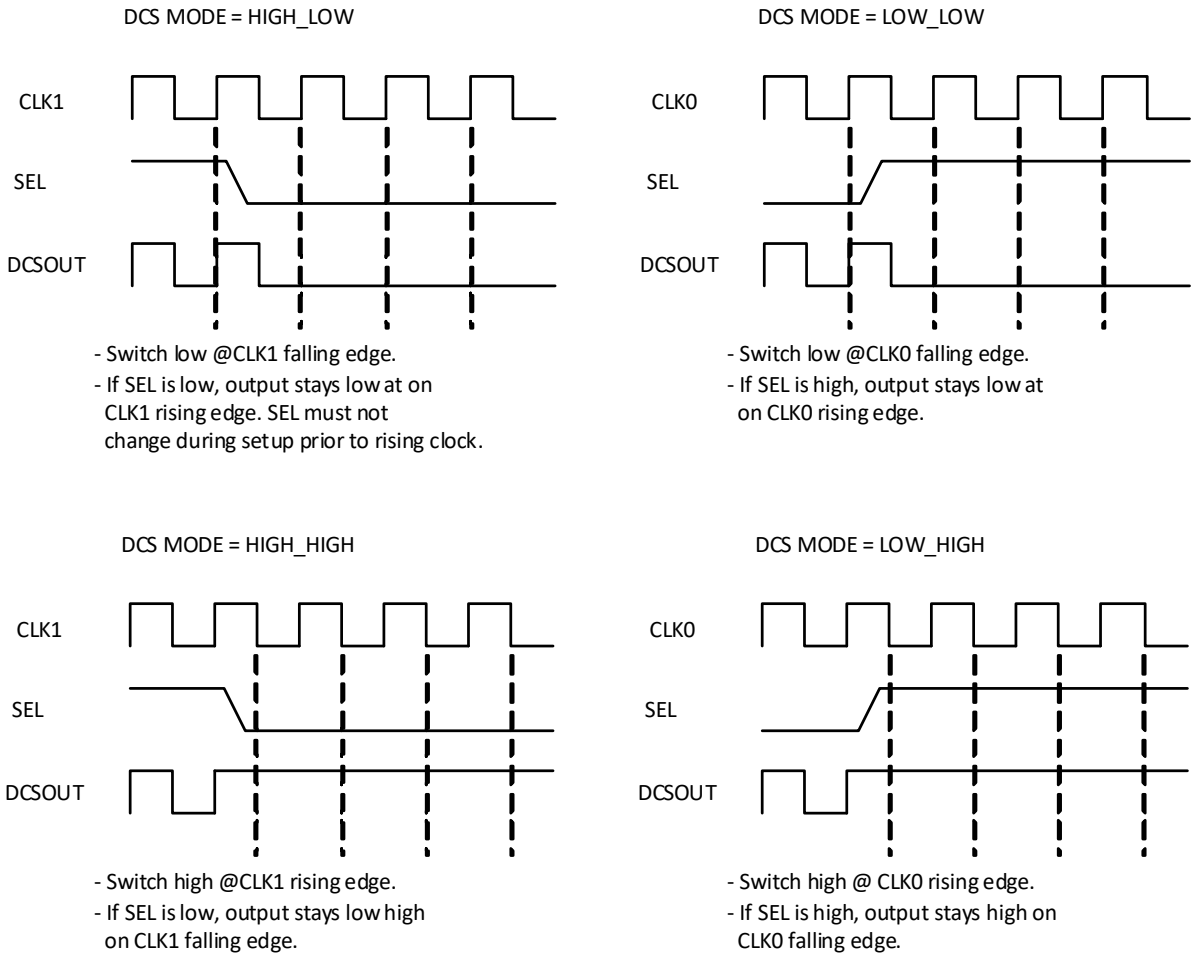

Figure 21.2. Timing Diagrams by DCS MODE *(Continued)*

21.3. DCS Usage with VHDL - Example

```
COMPONENT DCS
-- synthesis translate_off
        GENERIC    (
            DCSMODE : string := "POS"
      ;
-- synthesis translate_on

        PORT       (

END COMPONENT;

    attribute DCSMODE                : string;
    attribute DCSMODE of DCSinst0    : label is "POS";

begin

DCSInst0: DCS
-- synthesis translate_off

        GENERIC MAP (
                  DCSMODE  => "POS"
            )
-- synthesis translate_on

        PORT MAP (
      SEL   =>  clksel,
      CLK0  =>  dcsclk0,
      CLK1  =>  sysclk1,
      DCSOUT      =>  dcsclk
      );
```

21.4. DCS Usage with Verilog - Example

```
module dcs(clk0,clk1,sel,dcsout);

input clk0, clk1, sel;
output dcsout;

DCS DCSInst0 (.SEL(sel),.CLK0(clk0),.CLK1(clk1),.DCSOUT(dcsout));
defparam DCSInst0.DCSMODE = "CLK0";

endmodule
```

22. Oscillator (OSCE)

There is a dedicated oscillator in the LatticeXP2 device whose output is made available for users.

The oscillator frequency output is routed through a divider which is used as an input clock to the clock tree. The available outputs of the divider are shown in Table 22.1. The oscillator frequency output can be further divided by internal logic (user logic) for lower frequencies, if desired. The oscillator is powered down when not in use.

The output of this oscillator is not a precision clock. It is intended as an extra clock that does not require accurate clocking.

Primitive Name: OSCE

Table 22.1. OSCE Port Definition

I/O	Name	Description
Output	OSC	Oscillator Clock Output

Table 22.2. OSCE Attribute Definition

User Attribute	Attribute Name	Value (MHz)	Default Value
Nominal Frequency	NOM_FREQ	2.5, 3.14, 4.3, 5.4, 6.9, 8.1, 9.2, 10, 13, 15, 20, 26, 32, 40, 54, 80, 163	2.5

22.1. OSC Primitive Symbol (OSCE)

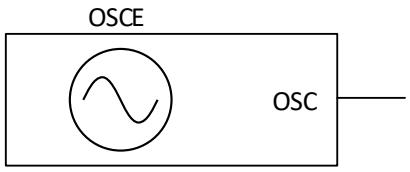

Figure 22.1. OSC Symbol

22.2. OSC Usage with VHDL - Example

```
COMPONENT OSCE
-- synthesis translate_off
      GENERIC (NOM_FREQ: string := "2.5");
-- synthesis translate_on
      PORT (OSC:OUT std_logic);
END COMPONENT;

attribute NOM_FREQ : string;
attribute NOM_FREQ of OSCinst0 : label is "2.5";

begin
OSCInst0: OSCE
-- synthesis translate_off
      GENERIC MAP ( NOM_FREQ => "2.5")
-- synthesis translate_on
      PORT MAP ( OSC => osc_int);
```

22.3. OSC Usage with Verilog - Example

```
module OSC_TOP(OSC_CLK);

output OSC_CLK;

OSCE OSCinst0 (.OSC(OSC_CLK));

defparam OSCinst0.NOM_FREQ = "2.5" ;
endmodule
```

22.4. Setting Clock Preferences

Designers can use clock preferences to implement clocks to the desired performance. Preferences can be set in the Pre-Map Preference Editor (Design Planner) or in preference files. Frequently used preferences are de-scribed in Appendix C.

23. Power Supplies

Each PLL has its own power supply pin, VCCPLL. Since VCCAUX and VCCPLL are normally the same 3.3V, it is recommended that they are driven from the same power supply on the circuit board, thus minimizing leakage. In addition, each of these supplies should be independently isolated from the main 3.3V supply on the board using proper board filtering techniques to minimize the noise coupling between them.

Appendix A. Primary Clock Sources and Distribution

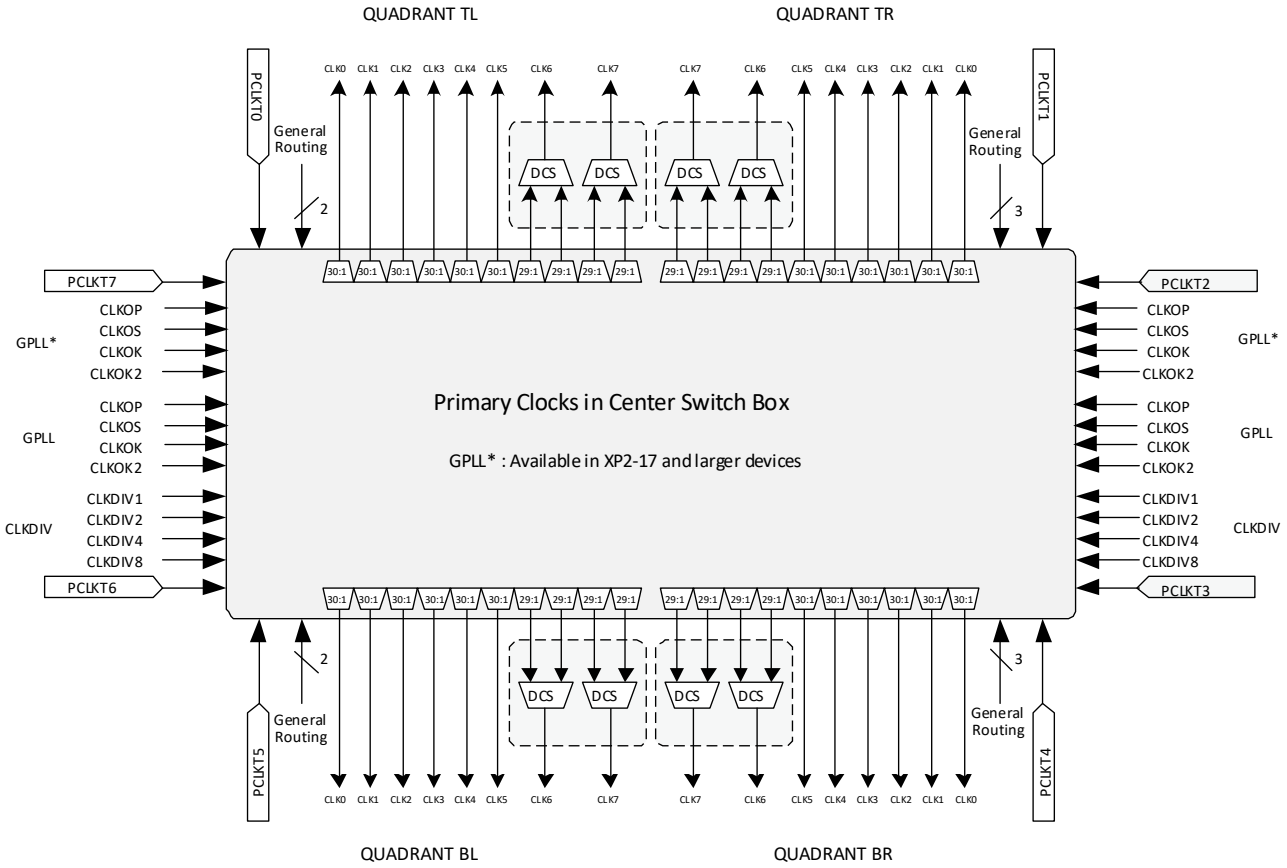

Figure A.1. LatticeXP2 Primary Clock Sources and Distribution

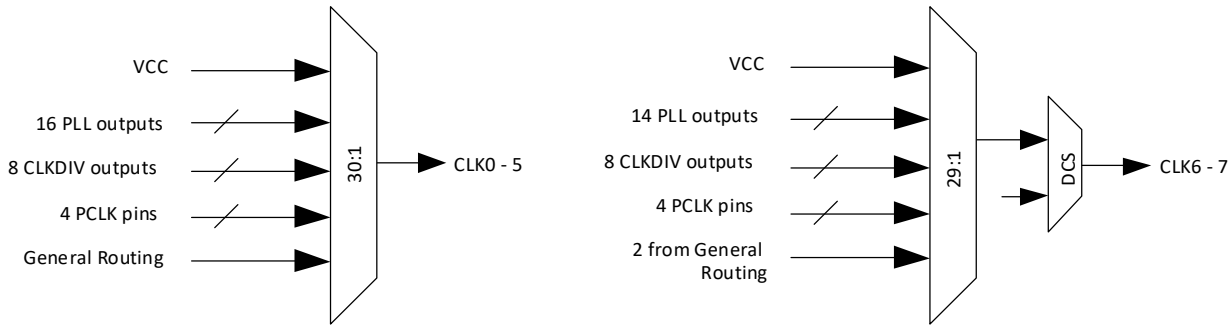

Figure A.2. LatticeXP2 Primary Clock Muxes

Appendix B. PLL, CLKIDV and ECLK Locations and Connectivity

Figure B.1 shows the locations, site names and connectivity of the PLLs, CLKDIVs and ECLKs

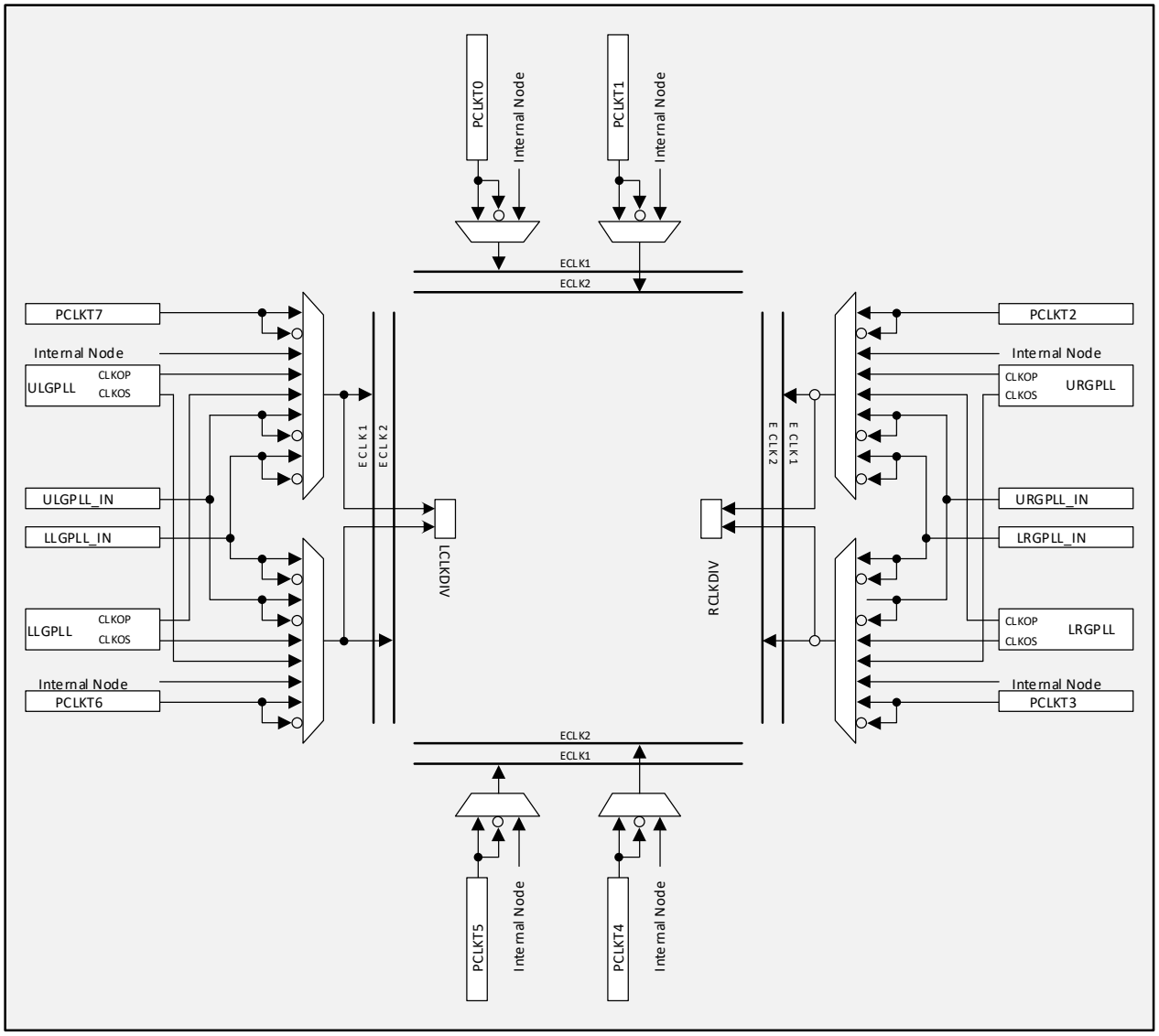

Figure B.1. PLL, CLKIDV and ECLK Locations and Connectivity

Appendix C. Clock Preferences

A few key clock preferences are introduced below. Refer to the 'Help' file for other preferences and detailed information.

ASIC

The following preference command assigns a phase of 90 degrees to the CIMDLLA CLKOP.

```
ASIC "my_dll" TYPE "CIMDLLA" CLKOP_PHASE=90;
```

FREQUENCY

The following physical preference command assigns a frequency of 100 MHz to a net named clk1:

```
FREQUENCY NET "clk1" 100 MHz;
```

The following preference specifies a hold margin value for each clock domain:

```
FREQUENCY NET "RX_CLKA_CMOS_c" 100.000 MHz HOLD_MARGIN 1 ns;
```

MAXSKEW

The following command assigns a maximum skew of 5 nanoseconds to a net named NetB:

```
MAXSKEW NET "NetB" 5 NS;
```

MULTICYCLE

The following command will relax the period to 50 nanoseconds for the path starting at COMPA to COMPB (NET1):

```
MULTICYCLE "PATH1" START COMP "COMPA" END COMP "COMPB" NET "NET1" 50 NS ;
```

PERIOD

The following command assigns a clock period of 30 nanoseconds to the port named Clk1:

```
PERIOD PORT "Clk1" 30 NS;
```

PROHIBIT

This command prohibits the use of a primary clock to route a clock net named bf_clk:

```
PROHIBIT PRIMARY NET "bf_clk";
```

USE PRIMARY

Use a primary clock resource to route the specified net:

```
USE PRIMARY NET clk_fast;
USE PRIMARY DCS NET "bf_clk";
USE PRIMARY PURE NET "bf_clk" QUADRANT_TL;
```

USE SECONDARY

Use a secondary clock resource to route the specified net:

```
USE SECONDARY NET "clk_lessfast" QUADRANT_TL;
```

USE EDGE

Use a edge clock resource to route the specified net:

```
USE EDGE NET "clk_fast";
```

CLOCK_TO_OUT

Specifies a maximum allowable output delay relative to a clock.

Here are two preferences using both the CLKPORT and CLKNET keywords showing the corresponding scope of TRACE reporting.

The CLKNET will stop tracing the path before the PLL, so you will not get PLL compensation timing numbers.

```
CLOCK_TO_OUT PORT "RxAddr_0" 6.000000 ns CLKNET "pll_rxclk" ;
```

The above preference will yield the following clock path:

Clock path pll_inst/pll_utp_0_0 to PFU_33:

Name	Fanout	Delay (ns)	Site	Resource
ROUTE	49	2.892	ULPPLL.MCLK to	R3C14.CLK0 pll_rxclk

	2.892	(0.0% logic, 100.0% route),	0 logic levels.	

If CLKPORT is used, the trace is complete back to the clock port resource and provides PLL compensation timing numbers.

```
CLOCK_TO_OUT PORT "RxAddr_0" 6.000000 ns CLKPORT "RxClk" ;
```

The above preference will yield the following clock path:

Clock path RxClk to PFU_33:

Name	Fanout	Delay (ns)	Site	Resource
IN_DEL	---	1.431	D5.PAD to	D5.INCK RxClk
ROUTE	1	0.843	D5.INCK to	ULPPLL.CLKIN RxClk_c
MCLK_DEL	---	3.605	ULPPLL.CLKIN to	ULPPLL.MCLK pll_inst/pll_utp_0_0
ROUTE	49	2.892	ULPPLL.MCLK to	R3C14.CLK0 pll_rxclk

	8.771	(57.4% logic, 42.6% route), 2 logic levels.		

INPUT_SETUP

Specifies a setup time requirement for input ports relative to a clock net.

```
INPUT_SETUP PORT "datain" 2.000000 ns HOLD 1.000000 ns CLKPORT "clk"
PLL_PHASE_BACK ;
```

PLL_PHASE_BACK

This preference is used with INPUT_SETUP when a user needs a trace calculation based on the previous clock edge.

This preference is useful when setting the PLL output phase adjustment. Since there is no negative phase adjustment provided, the PLL_PHASE_BACK preference works as if negative phase adjustment is available.

For example:

If phase adjustment of -90° of CLKOS is desired, a user can set the Phase to 270° and set the INPUT_SETUP preference with PLL_PHASE_BACK.

PLL_PHASE_BACK Usage in Pre-Map Preference Editor

The Pre-Map Preference Editor can be used to set the PLL_PHASE_BACK attribute.

1. Open the Design Planner (Pre-Map).
2. In the Design Planner control window, select View -> Spreadsheet View.
3. In the Spreadsheet View window, select Input_setup/Clock_to_out...

The INPUT_SETUP/CLOCK_TO_OUT Preference window is shown in Figure C.1.

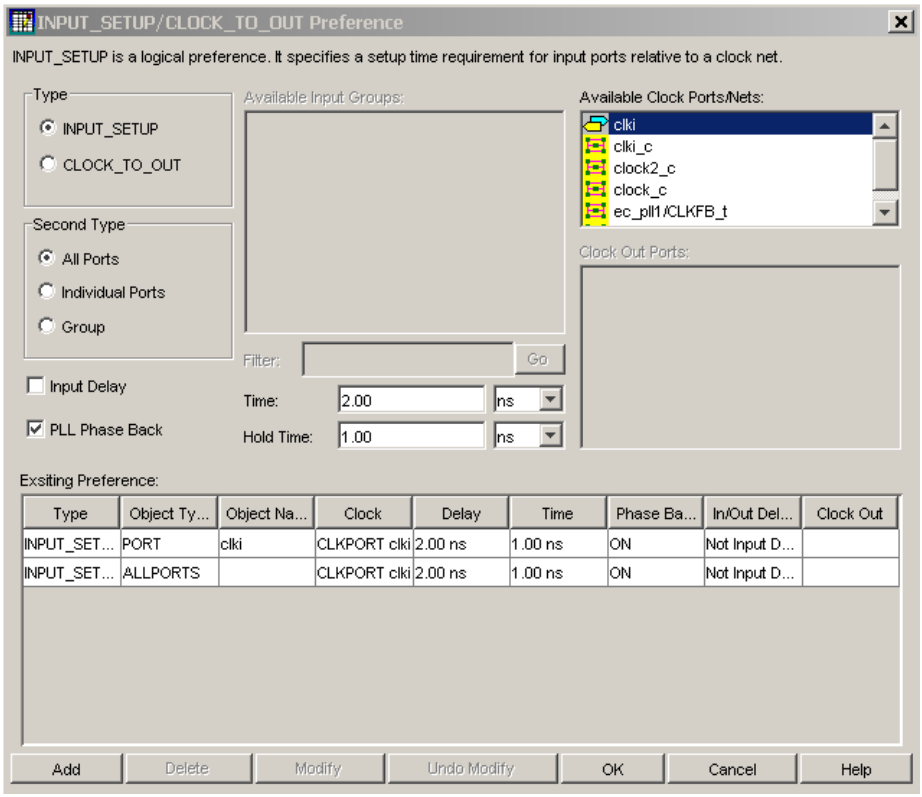

Figure C.1. INPUT_SETUP/CLOCK_TO_OUT Preference Window

Technical Support Assistance

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

Revision History

Revision 1.4, September 2021

Section	Change Summary
All	• Revised document number from TN1126 to FPGA-TN-02092. • Used straight quotes in code. • Updated page, table and figure numbering. • Changed document title from Usage Guide to User Guide for consistency
Clock Dividers	Corrected line in CLKDIV Usage with Verilog - Example section to defparam CLKDIBint0.GSR = "DISABLED";
Technical Support Assistance	Updated information.
Oscillator (OSCE)	• Removed this section from the DCS (Dynamic Clock Select) section. • Added VHDL and Verolog code examples for usage of NOM_FREQ in Section 22.2 OSC Usage with VHDL - Example and Section 22.3 OSC Usage with Verilog - Example respectively.

Revision 1.3, March 2012

Section	Change Summary
Appendix A	Updated LatticeXP2 Primary CLock Sources and Distribution figure and LatticeXP2 Primary Clock Muxes figure.

Revision 1.2, February 2012

Section	Change Summary
All	Updated document with new corporate logo.
Appendix A	Updated LatticeXP2 Primary CLock Sources and Distribution figure and LatticeXP2 Primary Clock Muxes figure.

Revision 1.1, February 2010

Section	Change Summary
All	Reconciled LOCK description among MachXO, LatticeXP2, LatticeECP2/M and LatticeECP3.

Revision 1.0, February 2007

Section	Change Summary
All	Initial release.

www.latticesemi.com