

Phase-Locked Loops

Introduction

The clock management tile (CMT) in Spartan-6 FPGAs includes two DCMs and one PLL. There are dedicated routes within a CMT to couple together various components. Each block within the tile can be treated separately, however, there exists a dedicated routing between blocks creating restrictions on certain connections. Using these dedicated routes frees up global resources for other design elements. Additionally, the use of local routes within the CMT provides an improved clock path because the route is handled locally, reducing chances for noise coupling.

The CMT diagram (Figure 3-1) shows a high-level view of the connection between the various clock input sources and the DCM-to-PLL and PLL-to-DCM dedicated routing. The six (total) PLL output clocks are MUXed into a single clock signal for use as a reference clock to the DCMs. Two output clocks from the PLL can drive the DCMs. These two clocks are 100% independent. PLL output clock 0 could drive DCM1 while PLL output clock 1 could drive DCM2. Each DCM output can be MUXed into a single clock signal for use as a reference clock to the PLL. Only one DCM can be used as the reference clock to the PLL at any given time. A DCM can not be inserted in the feedback path of the PLL. Both the PLLs or DCMs of a CMT can be used separately as stand-alone functions. The outputs from the PLL are not spread spectrum.

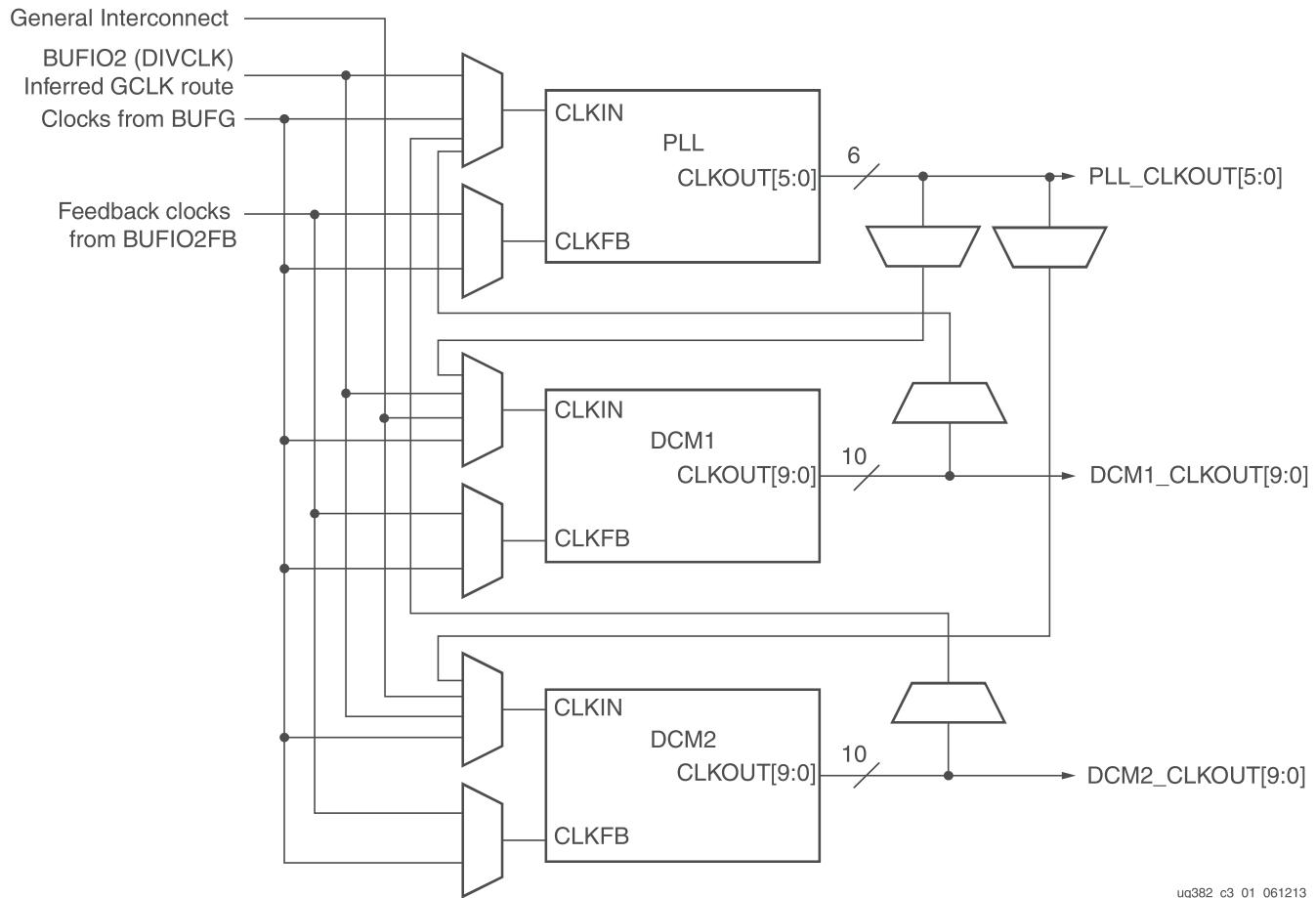


Figure 3-1: Block Diagram of the Spartan-6 FPGA CMT

To support the high-performance SDR clock rates associated with the PLL, each BUFPLL is directly routed to a restricted number of PLL locations. [Table 3-1](#) lists the PLL locations that can be connected to each BUFPLL.

Table 3-1: PLLs with Direct Connections to BUFPLL

| BUFPLL Location | Bank | Valid PLL Locations by Device Type | | |
|-----------------|-------|------------------------------------|--------------|--|
| | | LX4, LX9, LX16, LX25/LX25T | LX45/LX45T | LX75/LX75T LX100/LX100T LX150/LX150T |
| BUFPLL_X1Y5 | 0 | PLL_ADV_X0Y1 | PLL_ADV_X0Y3 | PLL_ADV_X0Y5 |
| BUFPLL_X1Y4 | | PLL_ADV_X0Y0 | PLL_ADV_X0Y2 | PLL_ADV_X0Y3 |
| BUFPLL_MCB_X1Y9 | | | PLL_ADV_X0Y1 | PLL_ADV_X0Y2 |
| BUFPLL_X2Y2 | 1 (5) | PLL_ADV_X0Y1 | PLL_ADV_X0Y3 | PLL_ADV_X0Y5 |
| BUFPLL_X2Y3 | | PLL_ADV_X0Y0 | PLL_ADV_X0Y2 | PLL_ADV_X0Y3 |
| BUFPLL_MCB_X2Y5 | | | PLL_ADV_X0Y1 | PLL_ADV_X0Y2 |
| BUFPLL_X1Y0 | 2 | PLL_ADV_X0Y1 | PLL_ADV_X0Y3 | PLL_ADV_X0Y5 |
| BUFPLL_X1Y1 | | PLL_ADV_X0Y0 | PLL_ADV_X0Y2 | PLL_ADV_X0Y3 |
| BUFPLL_MCB_X1Y5 | | | PLL_ADV_X0Y1 | PLL_ADV_X0Y2 |
| BUFPLL_X0Y2 | 3 (4) | PLL_ADV_X0Y1 | PLL_ADV_X0Y3 | PLL_ADV_X0Y5 |
| BUFPLL_X0Y3 | | PLL_ADV_X0Y0 | PLL_ADV_X0Y2 | PLL_ADV_X0Y3 |
| BUFPLL_MCB_X0Y5 | | | PLL_ADV_X0Y1 | PLL_ADV_X0Y2 |

Phase Lock Loop (PLL)

Spartan-6 devices contain up to six CMT tiles. The main purpose of PLLs is to serve as a frequency synthesizer for a wide range of frequencies, and to serve as a jitter filter for either external or internal clocks in conjunction with the DCMs of the CMT.

The PLL block diagram shown in [Figure 3-2](#) provides a general overview of the PLL components.

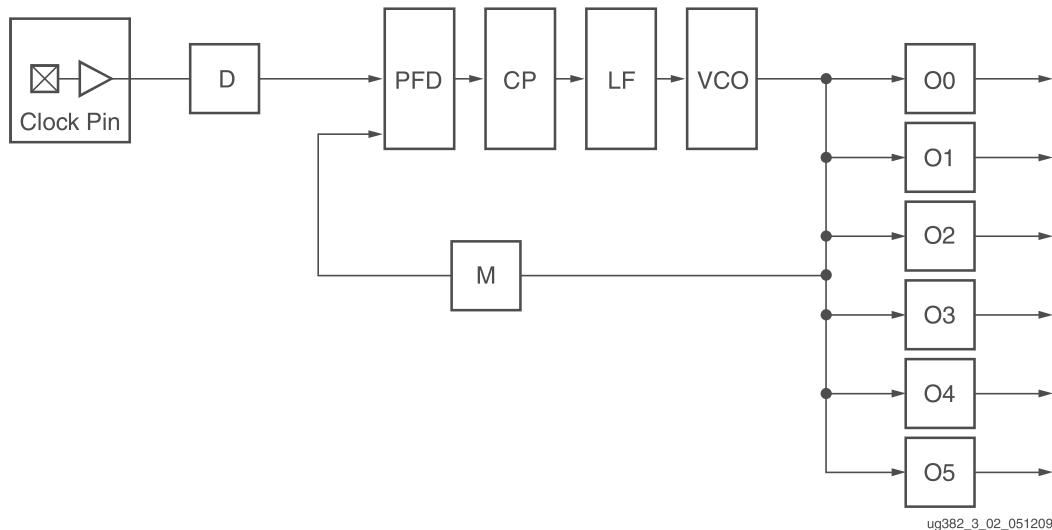
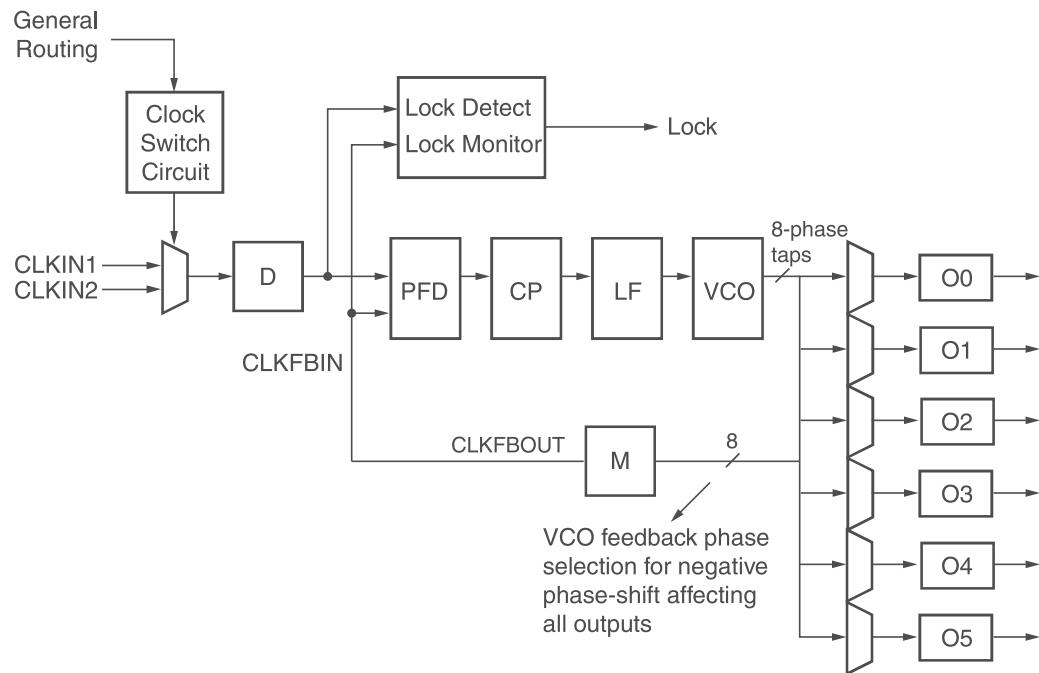


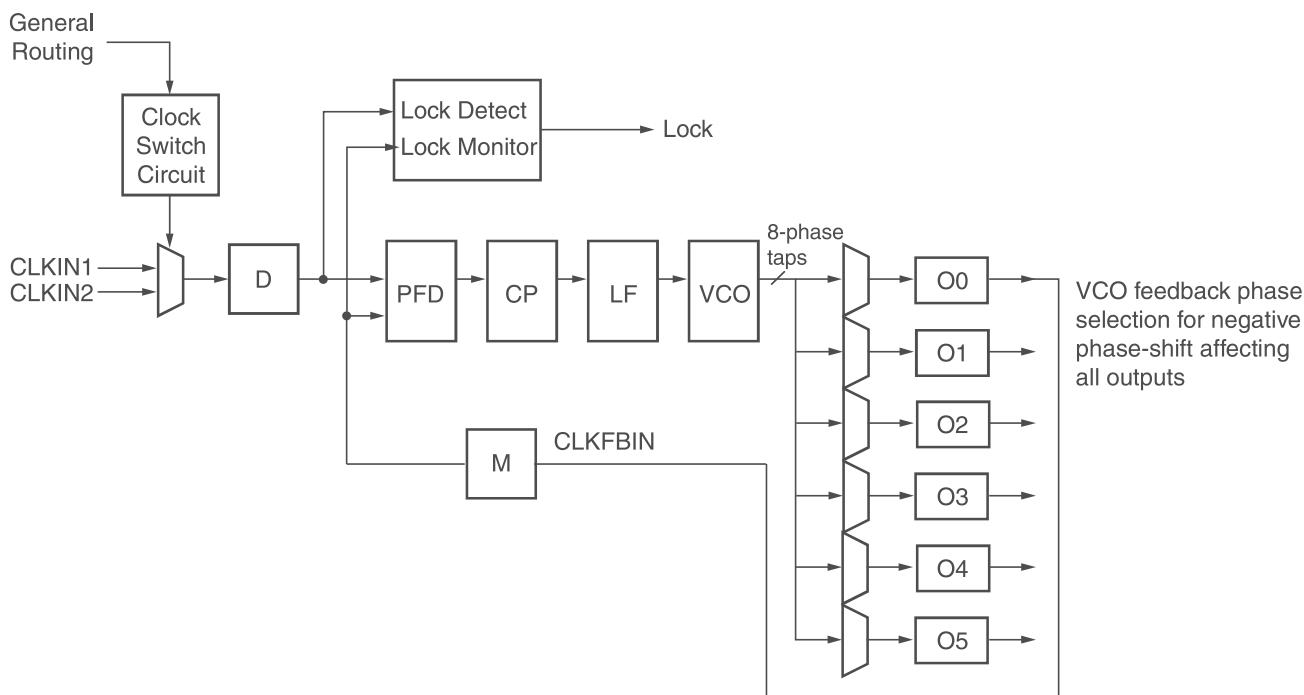
Figure 3-2: Block Diagram of the Spartan-6 FPGA PLL

Input MUXes select the reference and feedback clocks from either the IBUFG, BUFG, IBUF, PLL outputs, or one of the DCMs. Each clock input has a programmable counter D. The Phase-Frequency Detector (PFD) compares both phase and frequency of the input (reference) clock and the feedback clock. Only the rising edges are considered because as long as a minimum High/Low pulse is maintained, the duty cycle is not important. The PFD is used to generate a signal proportional to the phase and frequency between the two clocks. This signal drives the Charge Pump (CP) and Loop Filter (LF) to generate a reference voltage to the Voltage Controlled Oscillator (VCO). The PFD produces an up or down signal to the charge pump and loop filter to determine whether the VCO should operate at a higher or lower frequency. When VCO operates at too high of a frequency, the PFD activates a down signal, causing the control voltage to be reduced, which decreases the VCO operating frequency. When the VCO operates at too low of a frequency, an up signal will increase voltage. The VCO produces eight output phases. Each output phase can be selected as the reference clock to the output counters. See [Figure 3-3](#) and [Figure 3-4](#). Each counter can be independently programmed for a given customer design. A special counter, M, is also provided. This counter controls the feedback clock of the PLL allowing a wide range of frequency synthesis.



ug382_c3_03a_020510

Figure 3-3: Detailed PLL Block Diagram: CLK_FEEDBACK = CLKFBOUT



ug382_c3_04new_021110

Figure 3-4: Detailed PLL Block Diagram: CLK_FEEDBACK = CLKOUT0

Aligning PLL using CLK_FEEDBACK and BUFI02FB

The Spartan-6 FPGA PLL contains dedicated feedback routing used to minimize phase noise and increase the performance of the PLL clocking beyond the BUFG performance limitations. To use this dedicated PLL routing, CLK_FEEDBACK must be set to CLKOUT0 and use the BUFPPLL and BUFI02FB as shown in Figure 3-5.

Note: BUFPPLL and BUFPPLL_MCB input clocks can be connected to either CLKOUT0 or CLKOUT1 from the PLL.

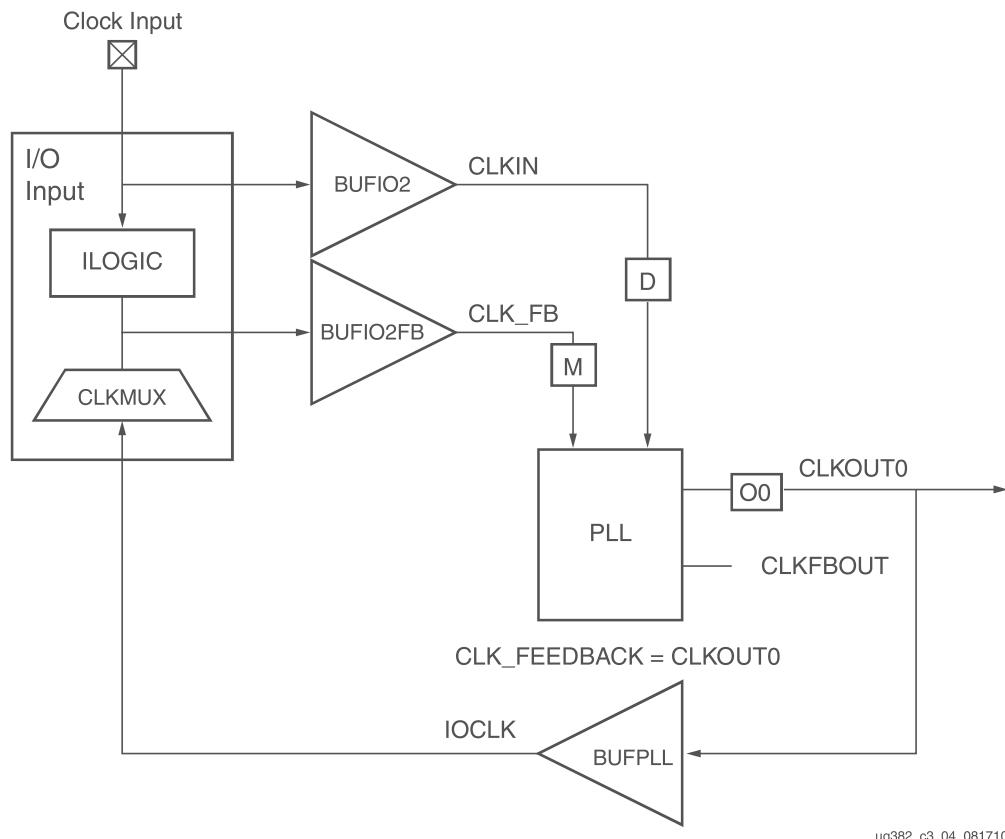


Figure 3-5: PLL Using CLKOUT0 Feedback

In high-speed source-synchronous designs, the feedback clock runs at the IOCLK frequency. To allow the feedback clock to match the input clock, the feedback clock is further divided using the CLKFBOUT_MULT as shown in Figure 3-4. Both CLKOUT0_DIVIDE and CLKFBOUT_MULT affect the VCO frequency.

$$f_{VCO} = \frac{f_{IN} \times CLKFBOUT_MULT \times CLKOUT0_DIVIDE}{DIVCLK_DIVIDE} \quad \text{Equation 3-1}$$

To accurately deskew the input routing, the BUFI02FB buffer must be used as shown in Figure 1-40, page 56. The BUFI02FB is matched to the BUFI02 buffer limiting deskewing to a single PLL.

When using CLK_FEEDBACK = CLKOUT0, the frequency for the CLKFBIN can be different than the frequency at the PFD. Be careful when setting CLK_FEEDBACK = CLKOUT0, because the PFD matches the frequency between the input clock ([Equation 3-2](#)) and the frequency from the feedback clock.

$$F_{PFD_CLKIN} = \frac{F_{CLKIN}}{DIVCLK_DIVIDE} \quad \text{Equation 3-2}$$

When CLK_FEEDBACK = CLKFBOUT, the frequency of CLKFBOUT ([Equation 3-3](#)) matches the feedback frequency for the PFD.

$$F_{CLKFBOUT} = \frac{F_{VCO}}{CLKFBOUT_MULT} \quad \text{Equation 3-3}$$

$$F_{PFD_CLKFBOUT} = F_{CLKFB} \quad \text{Equation 3-4}$$

When CLK_FEEDBACK = CLKOUT0, the output frequency for CLKOUT0 ([Equation 3-5](#)) can be different than the feedback frequency depending on CLKFB_MULT. As shown in [Figure 3-4](#), the frequency for CLKOUT0 is divided by CLKFB_MULT resulting in a PFD frequency that is dependent on CLKFB_MULT ([Equation 3-6](#)).

$$F_{CLKOUT0} = \frac{F_{VCO}}{CLKOUT0_DIVIDE} \quad \text{Equation 3-5}$$

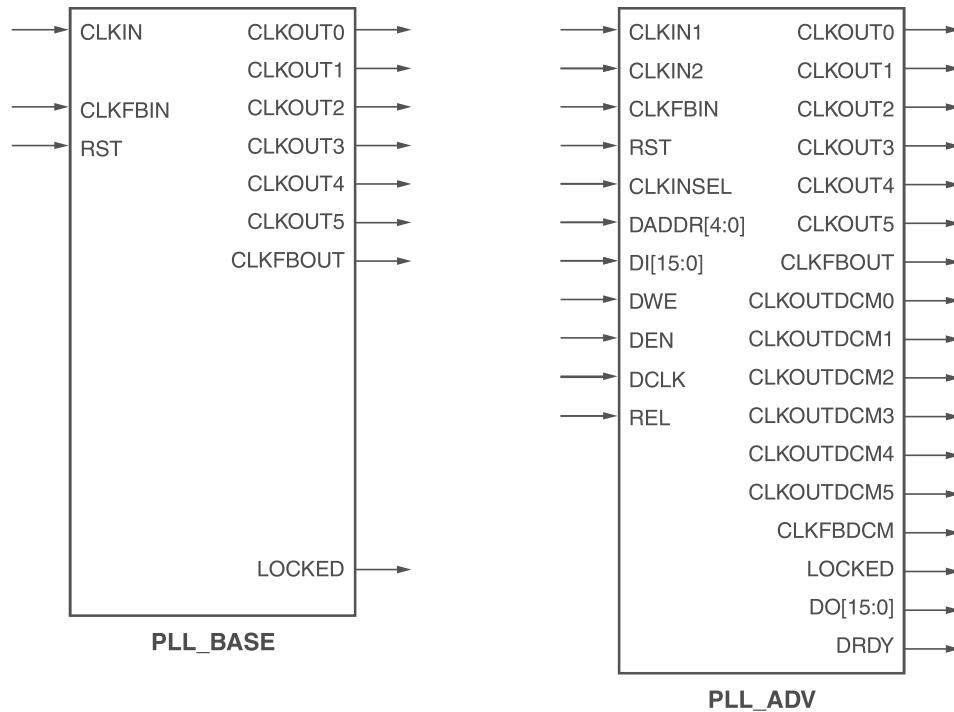
$$F_{PFD_CLKOUT0} = \frac{F_{CLKOUT0}}{CLKFB_MULT} \quad \text{Equation 3-6}$$

From [Equation 3-5](#), the output frequency for CLKOUT0 can be different than the VCO frequency.

General Usage Description

PLL Primitives

The two Spartan-6 FPGA PLL primitives, PLL_BASE and PLL_ADV, are shown in Figure 3-6.



ug382_c3_05_060710

Figure 3-6: PLL Primitives

PLL_BASE Primitive

The PLL_BASE primitive provides access to the most frequently used features of a stand alone PLL. Clock deskew, frequency synthesis, coarse phase shifting, and duty cycle programming are available to use with the PLL_BASE. The ports are listed in Table 3-2.

Table 3-2: PLL_BASE Ports

| Description | Port |
|-------------------------|------------------------------|
| Clock Input | CLKIN, CLKFBIN |
| Control Inputs | RST |
| Clock Output | CLKOUT0 to CLKOUT5, CLKFBOUT |
| Status and Data Outputs | LOCKED |

PLL_ADV Primitive

The PLL_ADV primitive provides access to all PLL_BASE features. PLL_ADV is provided for designs that dynamically reconfigure the PLL. For most design situations, use the PLL_BASE primitive or the clocking wizard. The ports are listed in [Table 3-3](#).

Table 3-3: PLL_ADV Ports

| Description | Port |
|------------------------|--|
| Clock Input | CLKIN1, CLKIN2, CLKFBIN, DCLK |
| Control and Data Input | RST, CLKINSEL, (Static 1 or Static 0), DWE, DEN, DADDR, DI |
| Clock Output | CLKOUT0 to CLKOUT5, CLKFBOUT, CLKOUTDCM0 to CLKOUTDCM5, CLKFBDCM |
| Status and Data Output | LOCKED |

The Spartan-6 FPGA PLL is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction. These three modes of operation are discussed in more detail within this section. The VCO operating frequency can be determined by using the following relationships:

CLK_FEEDBACK = CLKFBOUT is described in [Equation 3-7](#).

$$F_{VCO} = F_{CLKIN} \times \frac{M}{D} \quad \text{Equation 3-7}$$

CLK_FEEDBACK = CLKOUT0 is described in [Equation 3-8](#).

$$F_{VCO} = F_{CLKIN} \times \frac{M \times O_0}{D} \quad \text{Equation 3-8}$$

where the M, D, and O counters are shown in [Figure 3-3](#). O0 affects the VCO only when CLK_FEEDBACK = CLKOUT0. [Equation 3-9](#) shows the output frequency for CLKOUT[5:0]. [Equation 3-10](#) shows the output frequency for CLKFBOUT.

$$F_{OUT} = \frac{F_{VCO}}{O} \quad \text{Equation 3-9}$$

$$F_{OUT_CLKFBOUT} = \frac{F_{VCO}}{M} \quad \text{Equation 3-10}$$

The six “O” counters can be independently programmed. For example, O0 can be programmed to do a divide-by-two while O1 is programmed for a divide by three. The only constraint is that the VCO operating frequency must be the same for all the output counters since a single VCO drives all the counters.

Clock Network Deskew

In many cases, designers do not want to incur the delay on a clock network in their I/O timing budget therefore they use a PLL or DLL to compensate for the clock network delay. Spartan-6 FPGA PLLs support this feature. A clock output matching the reference clock CLKIN frequency (usually CLKFBOUT or CLKOUT) is connected to a BUFG and fed back to the CLKFBIN feedback pin of the PLL. The remaining outputs can still be used to divide

the clock down for additionally synthesized frequencies. In this case, all output clocks have a defined phase relationship to the input reference clock.

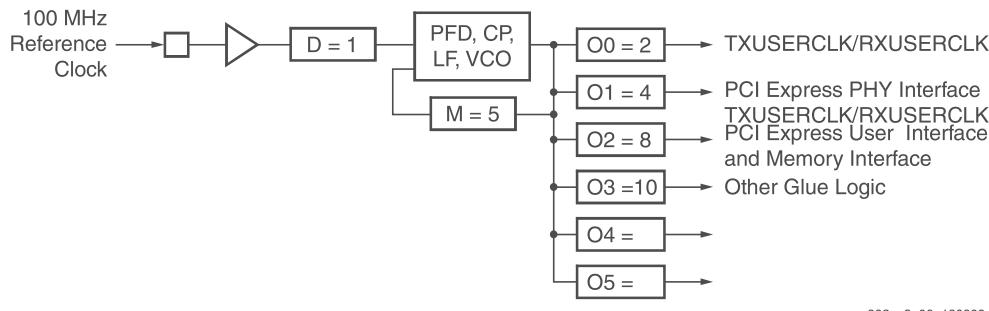
To accurately deskew the input routing, the BUFI02FB buffer must be used as shown in Figure 1-40.

Frequency Synthesis Only

PLLs can also be used for stand-alone frequency synthesis. In this application, a PLL can not be used to deskew a clock network, however, it is used generate an output clock frequency for other blocks. In this mode, the PLL feedback path should be set to INTERNAL since it keeps all the routing local and should minimize the jitter. Figure 3-7 shows the PLL configured as a frequency synthesizer. In this example, a clocking configuration for PCI Express x1 Gen1 is given. A 100 MHz reference clock is fed from the REFCLKOUT of the GTP transceiver. Setting the counters M = 5 and D = 1 makes the VCO oscillate at 500 MHz (100 MHz x 5). Ensure the VCO frequency meets the range specified in the Spartan-6 FPGA data sheet. Four of the six PLL outputs are programmed to provide:

- 250 MHz clock to the GTP transceiver's TXUSRCLK and RXUSRCLK
- 125 MHz clock for the PCI Express PHY interface as well as the GTP transceiver's TXUSRCLK2 and RXUSRCLK2 for data exchange in a 2-byte mode
- 62.5 MHz clock for the PCI Express User Interface and block RAM interface
- 50 MHz clock for other glue logic

In this example, no phase relationships between the reference clock and the output clocks are required, however, there are phase relationships required between the output clocks.



ug382_c3_06_120809

Figure 3-7: PLL as a Frequency Synthesizer

Jitter Filter

PLLs always reduce the jitter inherent on a reference clock. The PLL can be instantiated as a standalone function to simply support filtering jitter from an external clock before it is driven into the another block (including the DCM). As a jitter filter, it is usually assumed that the PLL acts as a buffer and regenerates the input frequency on the output (e.g., $F_{IN} = 100 \text{ MHz}$, $F_{OUT} = 100 \text{ MHz}$). In general, greater jitter filtering is possible by using the PLL attribute BANDWIDTH set to Low. Setting the BANDWIDTH to Low can incur an increase in the static offset of the PLL.

Limitations

The PLL has some restrictions that must be adhered to. These are summarized in the PLL electrical specification in the *Spartan-6 FPGA Data Sheet*. In general, the major limitations are VCO operation range, input frequency, duty cycle programmability, and phase shift.

VCO Operating Range

The minimum and maximum VCO operating frequencies are defined in the electrical specification of the *Spartan-6 FPGA Data Sheet*. These values can also be extracted from the speed specification.

Minimum and Maximum Input Frequency

The minimum and maximum CLKIN input frequency are defined in the electrical specification of the *Spartan-6 FPGA Data Sheet*.

Duty Cycle Programmability

Only discrete duty cycles are possible given a VCO operating frequency. The counter settings to determine the output duty cycle is further discussed under [Counter Control](#).

Phase Shift

In many cases, there needs to be a phase shift between clocks. The phase shift resolution in time units is defined as: $PS = 1/8 F_{VCO}$ or $D/8MF_{IN}$ since the VCO can provide eight phase shifted clocks at 45° each.

The higher the VCO frequency, the smaller the phase shift resolution. Since the VCO has a distinct operating range, it is possible to bound the phase shift resolution using from $1/8 F_{VCO_MIN}$ to $1/8 F_{VCO_MAX}$.

Each output counter is individually programmable allowing each counter to have a different phase shift based on the output frequency of the VCO.

Note: Phase shifts other than 45° are possible. A finer phase shift resolution depends on the output duty cycle and 0 value. Consult the clocking wizard for other phase-shift settings.

PLL Programming

Programming of the PLL must follow a set flow to ensure configuration that guarantees stability and performance. This section describes how to program the PLL based on certain design requirements. A design can be implemented in two ways, directly through the GUI interface (the clocking wizard) or directly implementing the PLL through instantiation. Regardless of the method selected, the following information is necessary to program the PLL:

- Reference clock period
- Output clock frequencies (up to six maximum)
- Output clock duty cycle (default is 50%)
- Output clock phase shift relative in number of clock cycles relative to the fastest output clock.
- Desired bandwidth of the PLL (default is OPTIMIZED and the bandwidth is chosen in software)
- Compensation mode (automatically determined by the software)

- Reference clock jitter in UI (i.e., a percentage of the reference clock period)

Determine the Input Frequency

The first step is to determine the input frequency. This allows all possible output frequencies to be determined by using the minimum and maximum input frequencies to define the D counter range, the VCO operating range to determine the M counter range, and the output counter range since it has no restrictions. There can be a very large number of frequencies. In the worst case, there will be $128 \times 128 \times 128 = 2,097,152$ possible combinations. In reality, the total number of different frequencies is less since the entire range of the M and D counters cannot be realized and there is overlap between the various settings. As an example, consider $F_{IN} = 100$ MHz. If the minimum PFD frequency is 19 MHz, then D can only go from 1 to 5. For D = 1, M can only have values from four to 10. If D = 2, M can have values from 8 to 20. In addition, D = 1 M = 4 is a subset of D = 2 M = 8 allowing the D = 2 M = 8 case to be dropped.

This drastically reduces the number of possible output frequencies. The output frequencies are sequentially selected. The desired output frequency should be checked against the possible output frequencies generated. Once the first output frequency is determined, an additional constraint can be imposed on the values of M and D. This can further limit the possible output frequencies for the second output frequency. Continue this process until all the output frequencies are selected.

The constraints used to determine the allowed M and D values are shown in the following equations:

$$D_{MIN} = \text{roundup} \frac{f_{IN}}{f_{PFD\ MAX}} \quad \text{Equation 3-11}$$

$$D_{MAX} = \text{rounddown} \frac{f_{IN}}{f_{PFD\ MIN}} \quad \text{Equation 3-12}$$

$$M_{MIN} = \left(\text{roundup} \frac{f_{VCOMIN}}{f_{IN}} \right) \times D_{MIN} \quad \text{Equation 3-13}$$

$$M_{MAX} = \text{rounddown} \frac{D_{MAX} \times f_{VCOMAX}}{f_{IN}} \quad \text{Equation 3-14}$$

Determine the M and D Values

Determining the input frequency can result in several possible M and D values. The next step is to determine the optimum M and D values. The starting M value is first determined. This is based off the VCO target frequency, the ideal operating frequency of the VCO.

$$M_{IDEAL} = \frac{D_{MIN} \times f_{VCOMAX}}{f_{IN}} \quad \text{Equation 3-15}$$

The goal is to find the M value closest to the ideal operating point of the VCO. The minimum D value is used to start the process. The goal is to make D and M values as small as possible while keeping f_{VCO} as high as possible.

PLL Ports

Table 3-4 summarizes the PLL ports. Table 3-5 lists the PLL attributes.

Table 3-4: PLL Ports

| Pin Name | I/O | Pin Description |
|-------------------------------|--------|---|
| CLKIN | Input | General clock input. |
| CLKIN1 | Input | PLL_ADV pin used for retargeting. General clock input. |
| CLKIN2 | Input | PLL_ADV pin used for retargeting. Secondary clock input. |
| CLKFBIN | Input | Feedback clock input. |
| CLKINSEL | Input | PLL_ADV pin used for retargeting. Connect to a static High or static Low to control the choice of clock input for PLL_ADV. High = CLKIN1, Low = CLKIN2. |
| RST | Input | Asynchronous reset signal. The RST signal is an asynchronous reset for the PLL. The PLL will synchronously re-enable itself when this signal is released (i.e., PLL re-enabled). A reset is required when the input clock conditions change (e.g., frequency). |
| DADDR[4:0] | Input | PLL_ADV dynamic reconfiguration address (DADDR) input bus. Provides reconfiguration address. When not used, all bits must be assigned zeros. |
| DI[15:0] | Input | PLL_ADV dynamic reconfiguration data input (DI) bus. Provides reconfiguration data. When not used, all bits must be assigned zeros. |
| DWE | Input | PLL_ADV dynamic reconfiguration write enable (DWE) bus. Provides the enable control signal to access the dynamic reconfiguration feature. When not used, DWE must be tied to zero. |
| DEN | Input | PLL_ADV dynamic reconfiguration enable (DEN) input. Provides the enable control signal to access the dynamic reconfiguration feature. When not used, DEN must be tied to zero. |
| DCLK | Input | PLL_ADV dynamic reconfiguration clock (DCLK) input. Provides reference clock for the dynamic reconfiguration port. When using a global clock buffer, only the upper eight BUFGMUXs can drive DCLK: BUFGMUX_X2Y1, BUFGMUX_X2Y2, BUFGMUX_X2Y3, BUFGMUX_X2Y4, BUFGMUX_X3Y5, BUFGMUX_X3Y6, BUFGMUX_X3Y7, and BUFGMUX_X3Y8. |
| REL | Input | PLL_ADV reserved pin. |
| CLKOUT[0:5] ⁽¹⁾ | Output | User configurable clock outputs (0 through 5) that can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 128. The input clock and output clocks are phase aligned. ⁽²⁾ |
| CLKFBOUT | Output | Dedicated PLL feedback output. ⁽²⁾ |
| CLKOUTDCM[0:5] ⁽¹⁾ | Output | PLL_ADV pin used for retargeting. User configurable clocks (0 through 5) that can only connect to the DCM within the same CMT as the PLL. ⁽²⁾ |
| CLKFBDCM | Output | PLL_ADV pin used for retargeting. PLL feedback used to compensate if the PLL is driving the DCM. If the CLKFBOUT pin is used for this purpose, the software will automatically map to the correct port. ⁽²⁾ |

Table 3-4: PLL Ports (Cont'd)

| Pin Name | I/O | Pin Description |
|----------|--------|---|
| LOCKED | Output | Asynchronous output from the PLL that indicates when the PLL has achieved phase alignment within a predefined window and frequency matching within a predefined PPM range. The PLL automatically locks after power on, no extra reset is required. LOCKED will be deasserted within one PFD clock cycle if the input clock stops or the phase alignment is violated (e.g., input clock phase shift). The PLL must be reset after LOCKED is deasserted. The FPGA waits for all DCMs and PLLs to be locked when LCK_CYCLE is set to control the startup cycles without setting the STARTUP_WAIT attribute on any DCM_SP port. In the starting configuration sequence, GTS must be deasserted for the PLL to lock. |
| DO[15:0] | Output | PLL_ADV dynamic reconfiguration output data bus. |
| DRDY | Output | PLL_ADV dynamic reconfiguration ready output (DRDY). Provides the response to DEN signal for the PLLs dynamic reconfiguration feature. |

Notes:

1. CLKOUT_N and CLKOUTDCM_N are utilizing the same output counters and can not be operated independently.
2. PLL clock outputs must use either a horizontal clock (default) or a global clock buffer.

PLL Attributes

Table 3-5: PLL Attributes

| Attribute | Type | Allowed Values | Default | Description |
|--------------------|---------|--|--------------------|---|
| COMPENSATION | String | SYSTEM_SYNCHRONOUS SOURCE_SYNCHRONOUS EXTERNAL | SYSTEM_SYNCHRONOUS | Specifies the PLL phase compensation for the incoming clock. SYSTEM_SYNCHRONOUS attempts to compensate all clock delay for 0 hold time. SOURCE_SYNCHRONOUS is used when a clock is provided with data and thus phased with the clock. EXTERNAL is used to compensate by routing the clock external to the FPGA. Additional attributes automatically selected by the ISE software: INTERNAL DCM2PLL PLL2DCM |
| BANDWIDTH | String | HIGH LOW OPTIMIZED | OPTIMIZED | Specifies the PLL programming algorithm affecting the jitter, phase margin and other characteristics of the PLL. |
| CLKOUT[0:5]_DIVIDE | Integer | 1 to 128 ⁽¹⁾ | 1 | Specifies the amount to divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency. |

Table 3-5: PLL Attributes (Cont'd)

| Attribute | Type | Allowed Values | Default | Description |
|------------------------|---------|-----------------|---------|---|
| CLKOUT[0:5]_PHASE | Real | -360.0 to 360.0 | 0.0 | Allows specification of the output phase relationship of the associated CLKOUT clock output in number of degrees offset (i.e., 90 indicates a 90° or ¼ cycle offset phase offset while 180 indicates a 180° offset or ½ cycle phase offset). When setting CLK_FEEDBACK = CLKOUT0, phase shifting results in a negative phase shift of all remaining clock outputs. |
| CLKOUT[0:5]_DUTY_CYCLE | Real | 0.01 to 0.99 | 0.50 | Specifies the Duty Cycle of the associated CLKOUT clock output in percentage (i.e., 0.50 will generate a 50% duty cycle). |
| CLKFBOUT_MULT | Integer | 1 to 64 | 1 | Specifies the amount to multiply all CLKOUT clock outputs if a different frequency is desired. This number, in combination with the associated CLKOUT#_DIVIDE value and DIVCLK_DIVIDE value, will determine the output frequency. |
| DIVCLK_DIVIDE | Integer | 1 to 52 | 1 | Specifies the division ratio for all output clocks with respect to the input clock. |
| CLKFBOUT_PHASE | Real | 0.0 to 360.0 | 0.0 | Specifies the phase offset in degrees of the clock feedback output. Shifting the feedback clock results in a negative phase shift of all output clocks to the PLL. |
| REF_JITTER | Real | 0.000 to 0.999 | 0.100 | Allows specification of the expected jitter on the reference clock in order to better optimize PLL performance. A bandwidth setting of OPTIMIZED will attempt to choose the best parameter for input clocking when unknown. If known, then the value provided should be specified in terms of the UI percentage (the maximum peak to peak value) of the expected jitter on the input clock. |
| CLKIN1_PERIOD | Real | 1.408 to 52.630 | 0.000 | Specifies the input period in ns to the PLL CLKIN1 input. Resolution is down to the ps. This information is mandatory and must be supplied. |
| CLKIN2_PERIOD | Real | 1.408 to 52.630 | 0.000 | Specifies the input period in ns to the PLL CLKIN2 input. Resolution is down to the ps. This information is mandatory and must be supplied. |

Table 3-5: PLL Attributes (Cont'd)

| Attribute | Type | Allowed Values | Default | Description |
|-----------------------|--------|------------------------------------|----------|--|
| CLK_FEEDBACK | String | CLKFBOUT or CLKOUT0 ⁽¹⁾ | CLKFBOUT | Specifies the clock source to drive CLKFB_IN. See Figure 3-5, page 96 for correct usage of feedback resources and calculating VCO frequency. |
| RESET_ON_LOSS_OF_LOCK | String | FALSE | FALSE | Must be set to FALSE, not supported in silicon. |

Notes:

- When CLK_FEEDBACK = CLKOUT0, CLKOUT0_DIVIDE is further restricted to ensure valid PFD and VCO frequencies. CLKOUT0_DIVIDE * CLKFBOUT_MULT must be 1 to 64.

PLL Clock Input Signals

The PLL clock source can come from several sources including:

- IBUFG - Global clock input buffer, the PLL will compensate the delay of this path.
- BUFG - Internal global clock buffer, the PLL will not compensate the delay of this path.
- IBUF - Not recommended since the PLL can not compensate for the delay of the general route. An IBUF clock input must route to a BUFG before routing to a PLL.
- DCMOUT - Any DCM output to PLL will compensate the delay of this path.
- BUFIO2 - When used with IBUFG, the DIVCLK output directly connects to the PLL clock input. The PLL compensates for this delay.

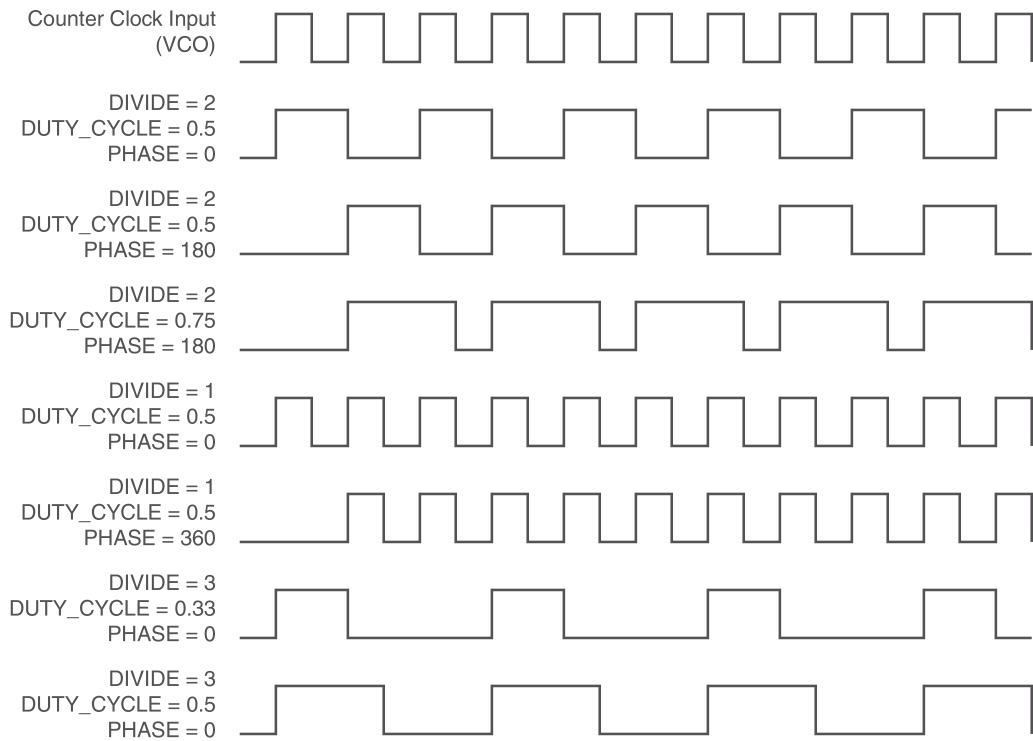
When global clock inputs are used to connect to a PLL, a BUFIO2 clock buffer will be inferred for optimal performance, as shown in [Figure 1-16, page 33](#). Each BUFIO2 either routes to the CMT on the top-half or the bottom-half of the device. The inferred BUFIO2 buffer can restrict routing to ensure proper phase alignment.

BUFIO2 buffers from BUFIO2 clocking regions TL, TR, RT, and LT route to the DCM/PLL on the top half of the device. Similarly, BUFIO2 buffers from BUFIO2 clocking regions BL, BR, RB, and LB connect to the DCM/PLL on the bottom half of the device.

Counter Control

The PLL output counters provide a wide variety of synthesized clocks using a combination of DIVIDE, DUTY_CYCLE, and PHASE. [Figure 3-8](#) illustrates how the counter settings impact the counter output.

The top waveform represents the output from the VCO in PLL mode.

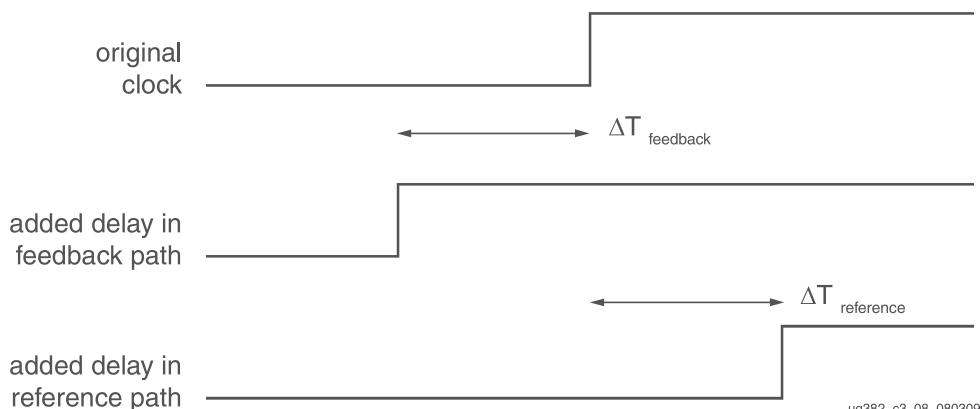


ug382_c3_07_080309

Figure 3-8: Output Counter Clock Synthesis Examples

Clock Shifting

The PLL output clocks can be shifted by inserting delay by selecting one of the eight phases in either the reference or the feedback path. Figure 3-9 shows the effect on a clock signal edge at the output of the PLL without any shifting versus the two cases (delay inserted in the feedback path and delay inserted in the reference path).



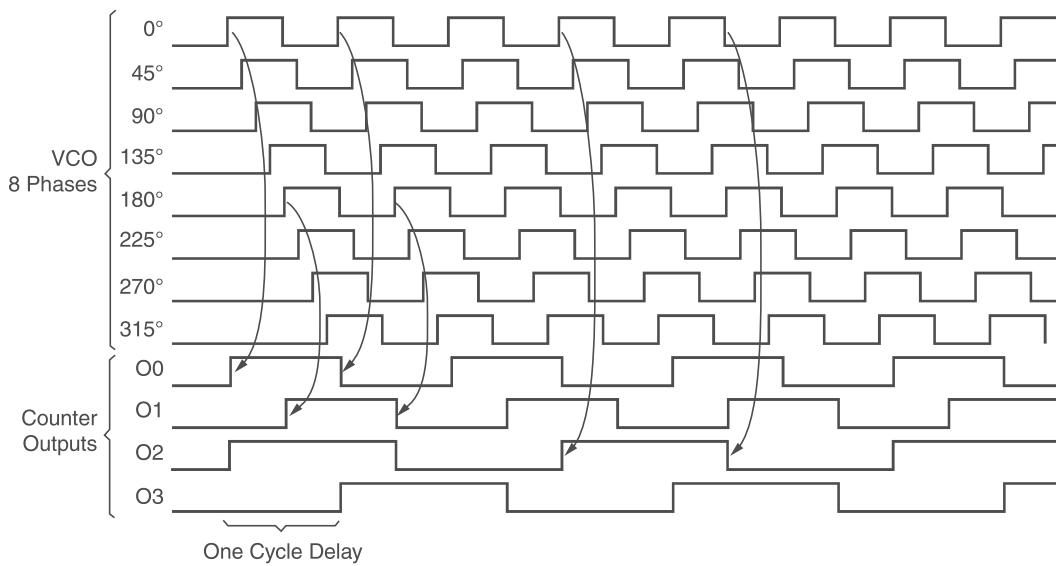
ug382_c3_08_080309

Figure 3-9: Basic Output Clock Shifting

Detailed VCO and Output Counter Waveforms

Figure 3-10 shows the eight VCO phase outputs and four different counter outputs. Each VCO phase is shown with the appropriate start-up sequence. The phase relationship and start-up sequence are guaranteed to insure the correct phase is maintained. This means the rising edge of the 0° phase will happen before the rising edge of the 45° phase. The O0 counter is programmed to do a simple divide by two with the 0° phase tap as the reference clock. The O1 counter is programmed to do a simple divide by two but uses the 180° phase tap from the VCO. Phase shifts greater than one VCO period are possible. This counter setting could be used to generate a clock for a DDR interface where the reference clock is edge aligned to the data transition. The O2 counter is programmed to do a divide by three. The O3 output has the same programming as the O2 output except the phase is set for a one cycle delay.

If the PLL is configured to provide a certain phase relationship and the input frequency is changed, then this phase relationship is also changed since the VCO frequency changes and therefore the absolute shift in picoseconds will change. This aspect must be considered when designing with the PLL. When an important aspect of the design is to maintain a certain phase relationship amongst various clock outputs, (e.g., CLK and CLK90) then this relationship will be maintained regardless of the input frequency.



ug382_c3_09_080309

Figure 3-10: Selecting VCO Phases

All “O” counters are equivalent, anything O0 can do, O1 can do. The PLL outputs are flexible when connecting to the global clock network since they are identical. In most cases, this level of detail is imperceptible to the designer as the software and clocking wizard determines the proper settings through the PLL attributes and Wizard inputs.

Missing Input Clock or Feedback Clock

When the input clock or feedback clock is lost, the PLL will drive the output clocks to a lower or higher frequency, causing all of the output clocks to increase/decrease in frequency. The frequency increase/decrease can cause the clock output frequencies to change to as much as six times the original configuration.

PLL Use Models

There are several methods to design with the PLL. The clocking wizard in ISE software can assist with generating the various PLL parameters. Additionally, the PLL can be manually instantiated as a component. It is also possible for the PLL to be merged with an IP core. The IP core would contain and manage the PLL.

Clock Network Deskew

One of the predominant uses of the PLL is for clock network deskew. Figure 3-11 shows the PLL in this mode. The clock output from one of the O counters is used to drive logic within the fabric and/or the I/Os. The feedback counter is used to control the exact phase relationship between the input clock and the output clock (if, for example a 90° phase shift is required). The associated clock waveforms are shown to the right for the case where the input clock and output clock need to be phase aligned. This configuration is the most flexible, but it does require two global clock networks (Figure 3-11).

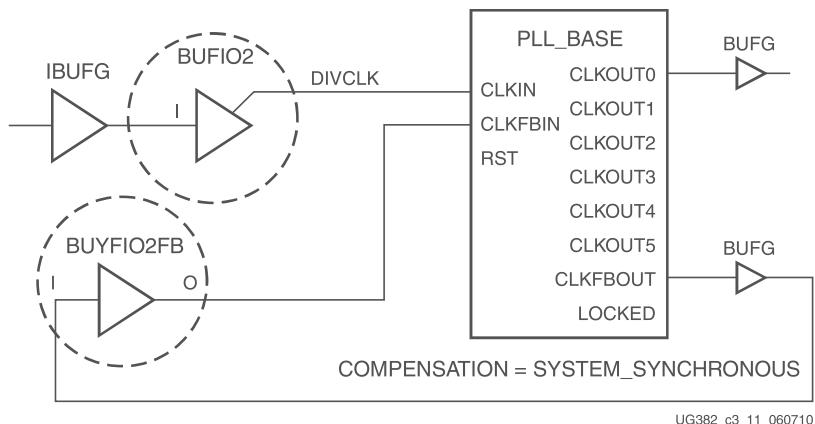


Figure 3-11: Clock Deskew Using Two BUFGs

There are certain restrictions on implementing the feedback. The CLKFBOUT output can be used to provide the feedback clock signal. The fundamental restriction is that both input frequencies to the PFD must be identical. Therefore, the following relationship must be met:

$$\frac{f_{IN}}{D} = f_{FB} = \frac{f_{VCO}}{M} \quad \text{Equation 3-16}$$

As an example, if f_{IN} is 166 MHz, $D = 1$, $M = 3$, and $O = 1$, then VCO and the clock output frequency are both 498 MHz. Since the M value in the feedback path is 3, both input frequencies at the PFD are 166 MHz.

In another more complex scenario has an input frequency of 66.66 MHz and $D = 2$, $M = 15$, and $O = 2$. The VCO frequency in this case is 500 MHz and the O output frequency is 250 MHz. Therefore, the feedback frequency at the PFD is 500/15 or 33.33 MHz, matching the 66.66MHz/2 input clock frequency at the PFD.

PLL with Internal Feedback

The PLL feedback can be internal to the PLL when the PLL is used as a synthesizer or jitter filter and there is no required phase relationship between the PLL input clock and the PLL output clock. The PLL performance should increase since the feedback clock is not subjected to noise on the core supply since it never passes through a block powered by this supply. Of course, noise introduced on the CLKIN signal and the BUFG will still be present (Figure 3-12).

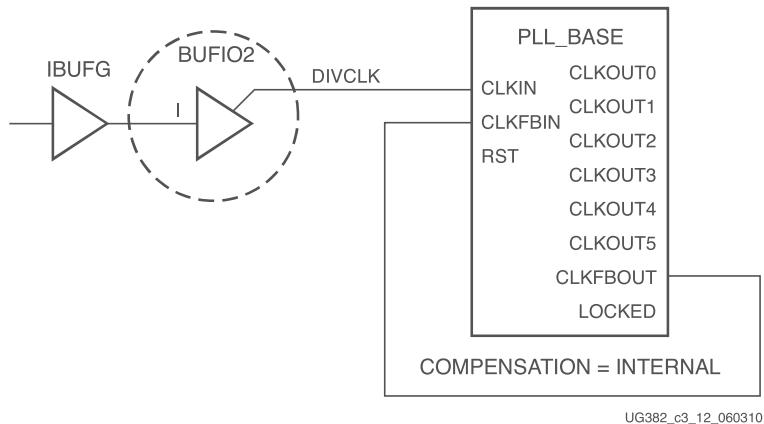


Figure 3-12: PLL with Internal Feedback

Zero Delay Buffer

The PLL can also be used to generate a zero delay buffer clock. A zero delay buffer can be useful for applications where there is a single clock signal fan out to multiple destinations with a low skew between them. This configuration is shown in Figure 3-13 for single-ended clocks. In Spartan-6 FPGAs, the BUFIO2FB and BUFIO2 placements must be matched and the feedback clock must be placed on the differential pair of the GCLK pin being used. In the case of a differential clock input where both pins of the differential clock are already being used, Figure 3-14 shows a circuit description using the equivalent GCLK in the neighboring BUFIO2 clocking region.

Note: There is an additional inversion of the OB output of the IBUFGDS_DIFF_OUT. This circuit will directly connect the differential GCLK to the matched BUFIO2FB.

The feedback signal is driven off the devices and the board-trace feedback is designed to match the trace to the external components. In this configuration, it is assumed that the clock edges are aligned at the input of the FPGA and the input of the external component. In this example, CLK_FEEDBACK = CLKFBOUT and the output logic is matched to a BUFG driving output logic with all signals using a single-ended I/O standard.

In some cases precise alignment is not possible because of the difference in loading between the input capacitance of the external component and the feedback path capacitance of the FPGA. For example, external components with an input capacitance of 1 pF to 4 pF where the FPGA has an input capacitance of around 8 pF. There is a difference in the signal slope, which is basically skew. To ensure timing, designers need to be aware of this effect.

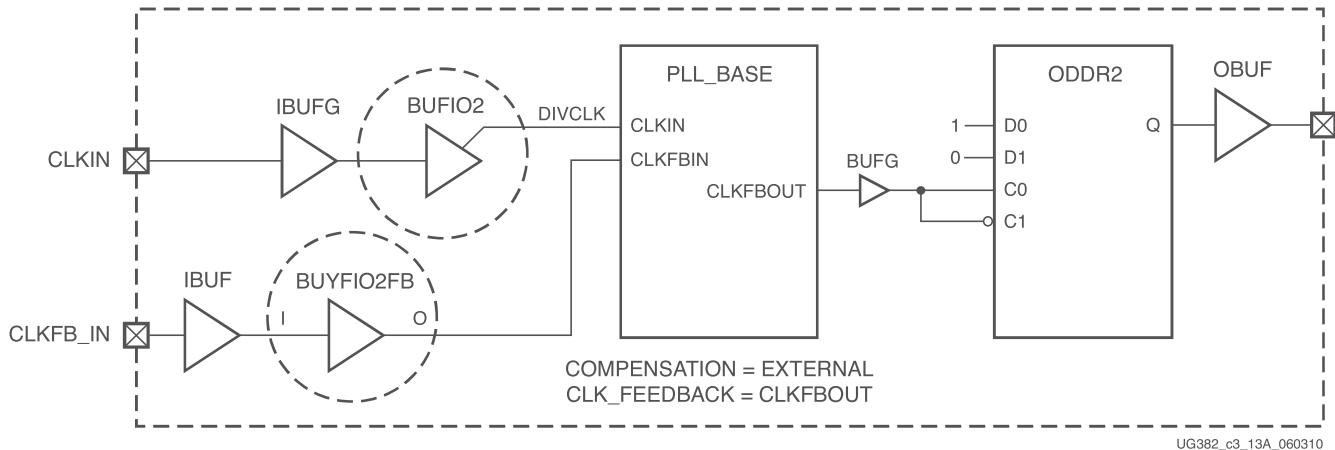


Figure 3-13: Zero Delay Buffer for Single-Ended Clocks

Note: When planning pinouts for a design using external feedback, the tools automatically match the BUFI02FB and BUFI02 to ensure optimal phase alignment.

For optimal placement, the input pad for the feedback clock must be placed on the differential pair. For example, if GCLK19 is used for CLKIN, then use GCLK18 for the CLKFB_IN. See [Table 1-6](#) for a complete placement list of global clock input pins.

As shown in [Figure 3-14](#), a BUFI02 from an adjacent BUFI02 clocking region is used if either the input clock or the feedback clock are differential clocks. For example, since CLKIN_P and CLKIN_N are connected to GCLK19 and GCLK18 respectively, then CLKFB_IN_P is connected to GCLK15 and CLKFB_IN_N is connected to GCLK14.

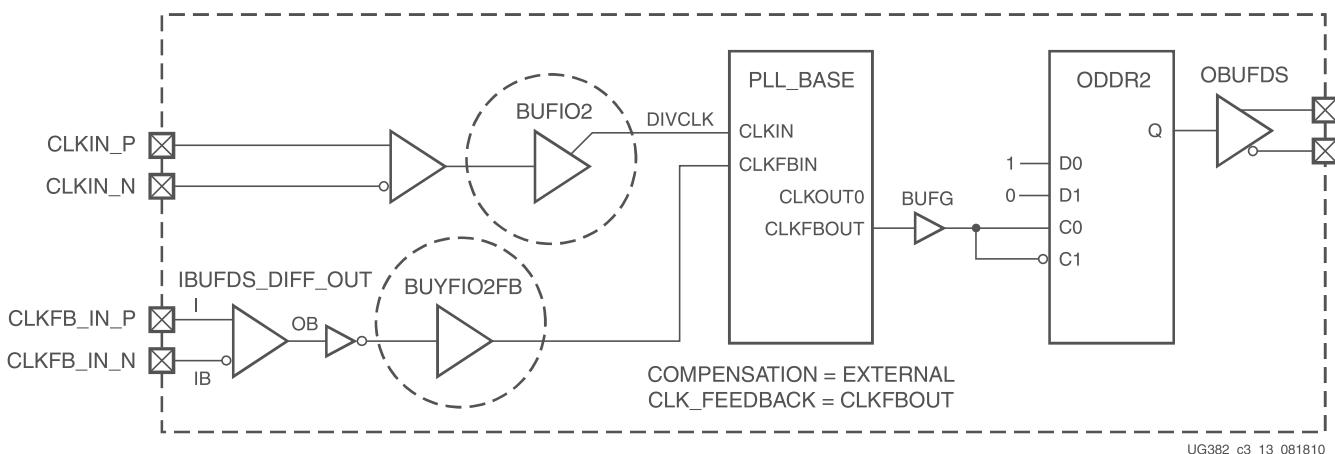


Figure 3-14: Zero Delay Buffer for Differential Clocks

For a differential feedback clock, IBUFGDS_DIFFOUT and an additional logic inversion must be used (as shown in [Figure 3-14](#)). The differential feedback clock pins are directly connected to the BUFI02FB. An additional logic inversion of the BUFI02FB input is used to compensate for the use of the OB output of the IBUFGDS_DIFFOUT. Verilog and VHDL examples are described in this section.

Differential BUFI02FB Zero Delay Buffer Example (Verilog)

```

IBUFDs_DIFF_OUT INST_IBUFDs_DIFF_OUT (
    .I      (CLKFB_IN_P),
    .IB     (CLKFB_IN_N),
    .OB     (CLKFB_IBUFDs_OB)) ;

BUFI02FB INST_BUFI02FB (
    .I   (~ CLKFB_IBUFDs_OB),
    .O   (CLK_FEEDBACK_TO_PLL));

```

Differential BUFI02FB Zero Delay Buffer Example (VHDL)

```

I_IBUFGDS: IBUFGDS_DIFF_OUT PORT MAP(
    I => CLKFB_IN_P, IB =>
    CLKFB_IN_N, OB => CLKFB_OB);

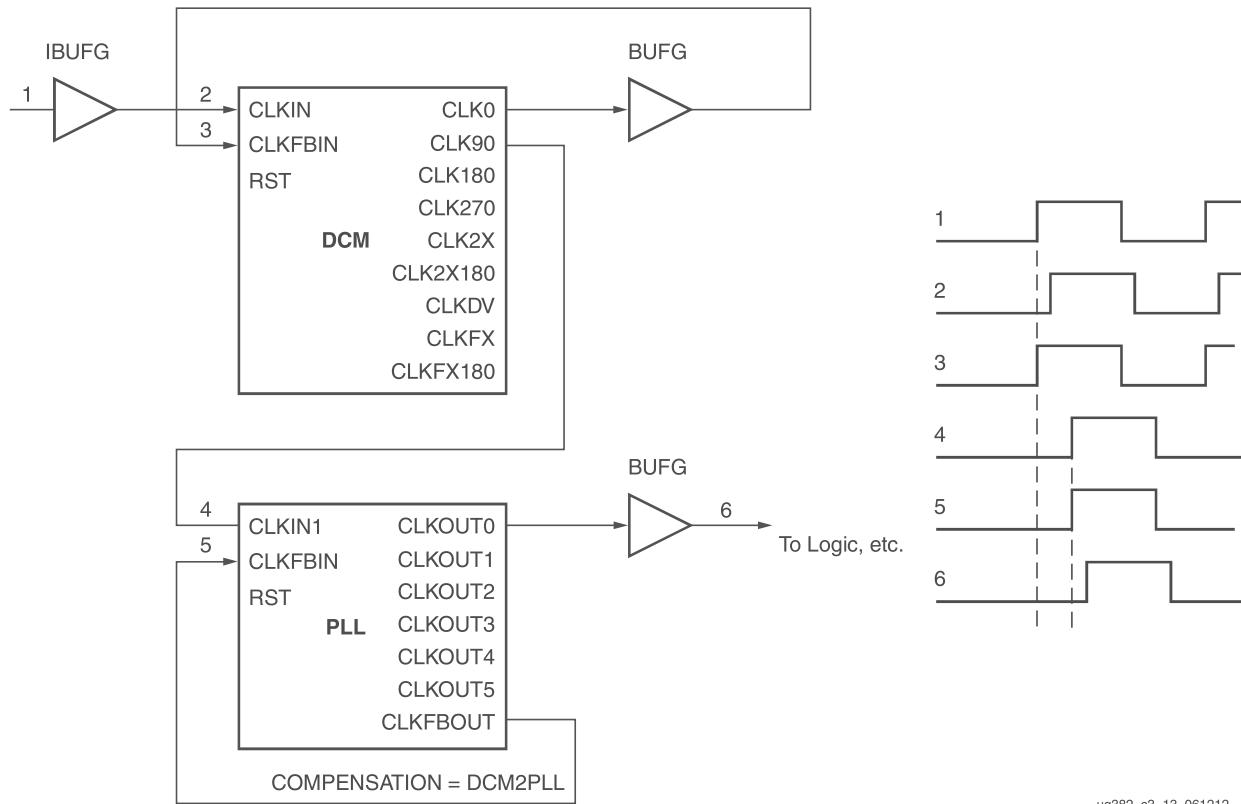
CLKFB_OB_180 <= not CLKFB_OB;

I_BUFI02FB: BUFI02FB PORT MAP (
    I => CLKFB_OB_180,
    O => CLKFB_BUFI02FB);

```

DCM Driving PLL

The DCM provides an excellent method for generating precision phase-shifted clocks. However, the DCM cannot reduce the jitter on the reference clock. The PLL can be used to reduce the output jitter of one DCM clock output. This configuration is shown in [Figure 3-15](#). The PLL is configured to not introduce any phase shift (zero delay through the PLL). The associated waveforms are shown to the right of the block diagram. When the output of the DCM is used to drive the PLL directly, both DCM and PLL *must* reside within the same CMT block. This is the preferred implementation since it produces a minimal amount of noise on the local, dedicated route. However, a connection can also be made by connecting the DCM to a BUFG and then to the CLKIN input of a PLL.



ug382_c3_13_061212

Figure 3-15: DCM Driving a PLL

PLL Driving DCM

A second option for reduce clock jitter is to use the PLL to clean-up the input clock jitter before driving into the DCM. This will improve the output jitter of all DCM outputs, but any added jitter by the DCM will still be passed to the clock outputs. Both PLL and DCM should reside in the same CMT block because dedicated resources exist between the PLL and DCM to support the zero delay mode. When the PLL and DCM do not reside in the same CMT, then the only connection is through a BUFG hindering the possibility of deskew.

One PLL can drive multiple DCMs as long as the reference frequency can be generated by a single PLL. For example, if a 33 MHz reference clock is driven into the PLL, and the design uses one DCM to operate at 200 MHz and the other to run at 100 MHz, then the VCO can be operated at 600 MHz ($M1 = 18$). The VCO frequency can be divided by three to generate a 200 MHz clock and another counter can be divided by six to generate the 100 MHz clock. For the example in Figure 3-16, one PLL drives two DCMs.

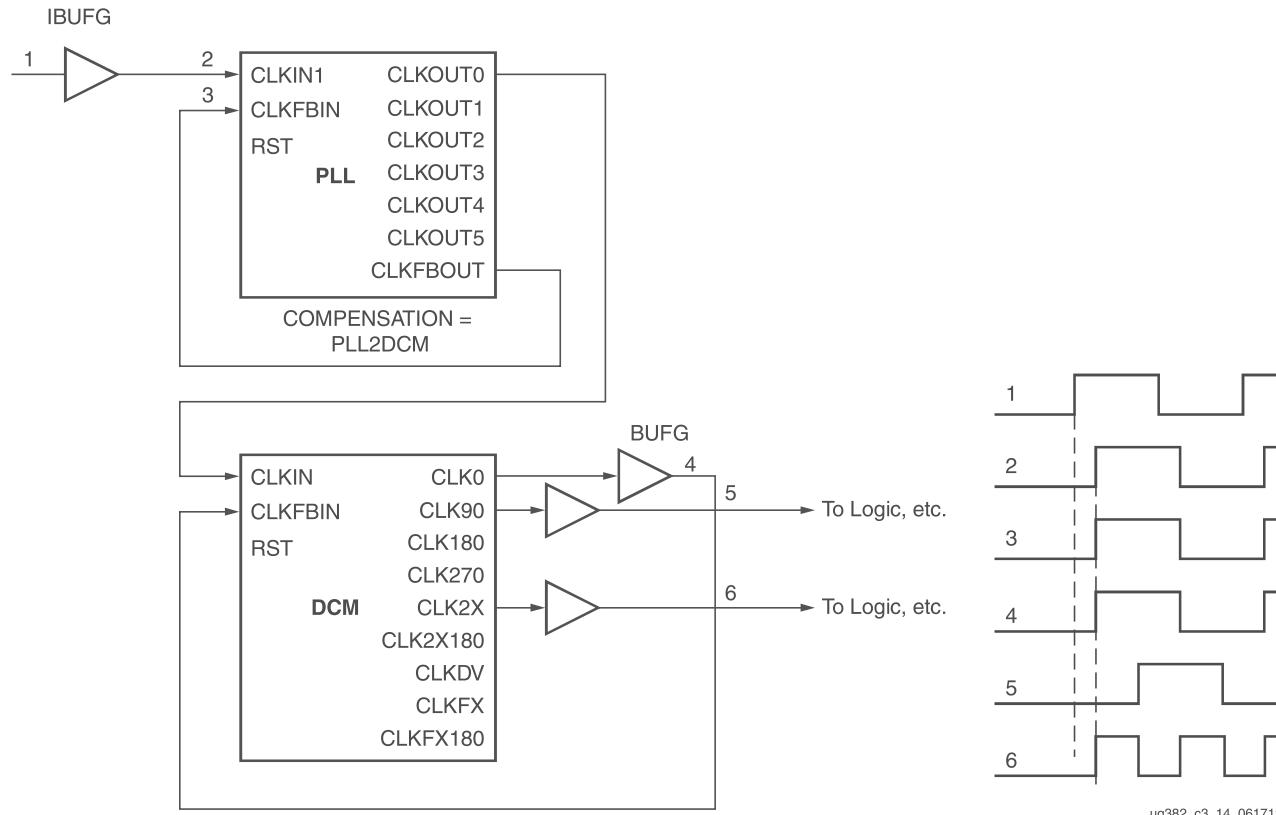


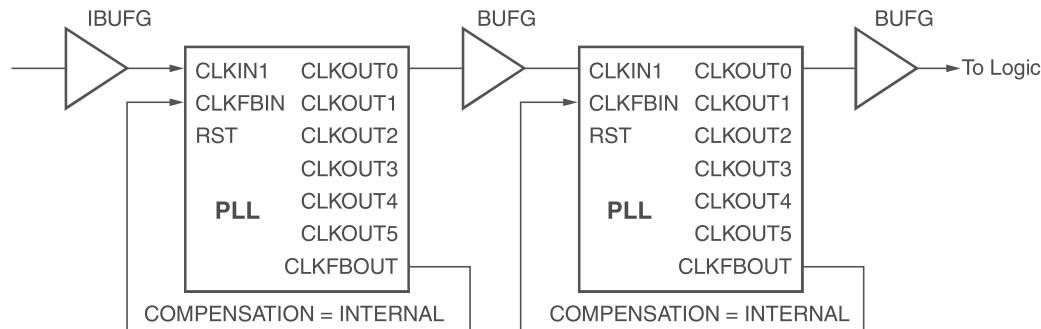
Figure 3-16: PLL Driving a DCM

PLL to PLL Connection

The PLL can be cascaded to allow generation of a greater range of clock frequencies. The frequency range restrictions still apply. Equation 3-17 shows the relationship between the final output frequency and the input frequency and counter settings of the two PLLs (Figure 3-17.) The phase relationship between the output clock of the second PLL and the input clock is undefined. To cascade PLLs, route the output of the first PLL to a BUFG and then to the CLKIN pin of the second PLL. This path provides the lowest device jitter.

$$f_{OUTPLL2} = f_{OUTPLL1} \frac{M_{PLL2}}{D_{PLL2} \times O_{PLL2}} = f_{IN} \frac{M_{PLL1}}{D_{PLL1} \times O_{PLL1}} \times \frac{M_{PLL2}}{D_{PLL2} \times O_{PLL2}}$$

Equation 3-17



ug382_c3_15_060710

Figure 3-17: Cascading Two PLLs

Dynamic Reconfiguration Port

Details of the supported Spartan-6 FPGAs PLL DRP operations are described in [XAPP879, PLL Dynamic Reconfiguration](#).

Application Guidelines

This section summarizes when to select a DCM over a PLL, or a PLL over a DCM.

Spartan-6 FPGA PLLs support up to six independent outputs. Designs using several different outputs should use PLLs. An example of designs using several different outputs follows. The PLL is an ideal solution for this type of application because it can generate a configurable set of outputs over a wide range while the DCM has a fixed number of predetermined outputs based off the reference clock. When the application requires a fine phase shift or a dynamic variable phase shift, a DCM could be a better solution.

PLL Application Example

The following PLL attribute settings result in a wide variety of synthesized clocks:

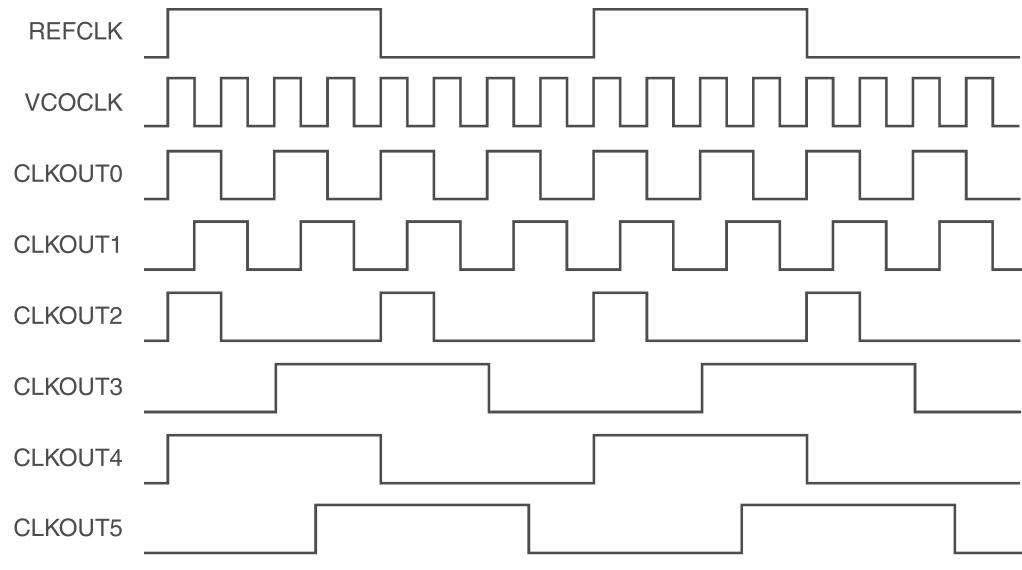
```

CLKOUT0_PHASE = 0;
CLKOUT0_DUTY_CYCLE = 0.5;
CLKOUT0_DIVIDE = 2;
CLKOUT1_PHASE = 90;
CLKOUT1_DUTY_CYCLE = 0.5;
CLKOUT1_DIVIDE = 2;
CLKOUT2_PHASE = 0;
CLKOUT2_DUTY_CYCLE = 0.25;
CLKOUT2_DIVIDE = 4;
CLKOUT3_PHASE = 90;
CLKOUT3_DUTY_CYCLE = 0.5;
CLKOUT3_DIVIDE = 8;
CLKOUT4_PHASE = 0;
CLKOUT4_DUTY_CYCLE = 0.5;
CLKOUT4_DIVIDE = 8;
CLKOUT5_PHASE = 135;
CLKOUT5_DUTY_CYCLE = 0.5;
CLKOUT5_DIVIDE = 8;
CLKFBOUT_PHASE = 0;
CLKFBOUT_MULT = 8;
DIVCLK_DIVIDE = 1;

```

```
CLKIN1_PERIOD = 10.0;
```

Figure 3-18 displays the resulting waveforms.



UG382_c3_16_080309

Figure 3-18: Example Waveform