

UltraScale Architecture Clocking Resources

Advance Specification User Guide

UG572 (v1.0) December 10, 2013



Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at www.xilinx.com/legal.htm#tos.

Automotive Applications Disclaimer

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.

This document contains preliminary information and is subject to change without notice. Information provided herein relates to products and/or services not yet available for sale, and provided solely for information purposes and are not intended, or to be construed, as an offer for sale or an attempted commercialization of the products and/or services referred to herein.

© Copyright 2013 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. PCI, PCIe, and PCI Express are trademarks of PCI-SIG and used under license. All other trademarks are the property of their respective owners.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/10/2013	1.0	Initial Xilinx release.

Table of Contents

Revision History	2
Chapter 1: Overview	
Introduction to UltraScale Architecture	5
Clocking Overview	5
Clocking Architecture Overview	6
Clock Routing Resources Overview	6
CMT Overview	7
Clocking Differences from Previous FPGA Generations	8
Key Differences from 7 Series FPGAs	8
Chapter 2: Clocking Resources	
Global Clock Inputs	10
Byte Clock Inputs	11
Clock Buffers and Clock Routing	11
Clock Structure	11
Clock Buffers	14
BUFGCTRL Clock Buffer Primitives	15
BUFGCE Clock Buffers	25
BUFG Clock Buffer	26
BUFGCE_LEAF Clock Buffer	26
BUFGCE_DIV	26
BUFG_GT	27
Chapter 3: Clock Management Tile	
Introduction	30
MMCMs	30
General Usage Description	31
Clock Network Deskew	33
Frequency Synthesis Only Using Integer Divide	33
Frequency Synthesis Using Fractional Divide in the MMCM	34
Jitter Filter	34
Limitations	35
Dynamic Phase Shift Interface in the MMCM	37
Clock Divide Dynamic Change	38
MMCM Counter Cascading	39
MMCM Programming	39
MMCM Ports	41
MMCM Attributes	46

MMCM Clock Input Signals	50
Counter Control	50
Detailed VCO and Output Counter Waveforms	51
Reference Clock Switching	52
MMCM Use Models	52
Zero Delay Buffer	54
CMT to CMT Connection	55
Spread-Spectrum Clock Generation	55
MMCM Application Example	60
PLLs	61
General Usage Description	62
PLL Ports	63
PLL Attributes	66
VHDL and Verilog Templates and the Clocking Wizard	68

Appendix A: Additional Resources

Xilinx Resources	69
Solution Centers	69
References	69

Overview

Introduction to UltraScale Architecture

Xilinx® UltraScale™ architecture is a revolutionary approach to creating programmable devices capable of addressing the massive I/O and memory bandwidth requirements of next generation applications while efficiently routing and processing the data brought on chip. UltraScale architecture-based FPGAs address a vast spectrum of high-bandwidth, high-utilization system requirements through industry-leading technical innovations. UltraScale architecture-based devices share many building blocks to provide optimized scalability across the product range, as well as numerous new power reduction features for low total power consumption.

Kintex® UltraScale FPGAs provide high performance with a focus on optimized performance per watt for applications including wireless, wired, and signal or image processing. High DSP and block RAM-to-logic ratios, and next generation transceivers are combined with low-cost packaging to enable an optimum blend of capability for these applications.

Virtex® UltraScale FPGAs provide the highest system capacity, bandwidth, and performance. Delivering unprecedented logic capacity, serial I/O bandwidth, and on-chip memory, the Virtex UltraScale family pushes the performance envelope ever higher.

This user guide describes the UltraScale architecture clocking resources and is part of the UltraScale Architecture documentation suite available at: www.xilinx.com/ultrascale.

Clocking Overview

This chapter provides an overview of clocking, a comparison between clocking in the UltraScale architecture and previous FPGA generations, and a summary of clocking connectivity. For detailed information on usage of clocking resources, see [Chapter 2, Clocking Resources](#) and [Chapter 3, Clock Management Tile](#).

Clocking Architecture Overview

The UltraScale architecture clocking resources manage complex and simple clocking requirements with dedicated global clocks distributed on clock routing and clock distribution resources. The clock management tiles (CMTs) provide clock frequency synthesis, deskew, and jitter filtering functionality. Non-clock resources such as local routing are not recommended when designing for clock functions.

- The fabric is subdivided into columns and rows of segmented clock regions (CRs). CRs differ from previous families because they are arranged in tiles and do not span half the width of a device. A CR contains configurable logic blocks (CLBs), DSP slices, block RAMs, interconnect, and associated clocking. The height of a CR is 60 CLBs, 24 DSP slices, and 12 block RAMs with a horizontal clock spine (HCS) at its center. The HCS contains the horizontal routing and distribution resources, leaf clock buffers, clock network interconnections, and the root of the clock network. Clock buffers drive directly into the HCS. There are 52 I/Os per bank and four Gigabit transceivers (GTs) that are pitch matched to the CRs. A core column contains configuration, System Monitor (SYSMON), and PCIe® blocks to complete a basic device.
- Adjacent to the input/output block columns are the physical layer (PHY) blocks with CMTs, global clock buffers, global clock multiplexing structures, and I/O logic management functions. The clocking drives vertical and horizontal connectivity through separate clock routing and clock distribution resources via HCS into the fabric.
- Horizontal clock routing and distribution tracks drive horizontally into the CRs. Vertical routing and distribution tracks drive vertically adjacent CRs. The tracks are segmentable at the CR boundaries in both the horizontal and vertical directions. This allows for the creation of device-wide global clocks or local clocks of variable size.
- The distribution tracks drive the clocking of synchronous elements across the device. Distribution tracks are driven by routing tracks or directly by the clocking structures in the PHY.
- I/Os are directly driven from the PHY clocking and/or an adjacent PHY via routing tracks.
- A CMT contains one mixed-mode clock manager (MMCM) and two phase-locked loops (PLLs).

Clock Routing Resources Overview

Each I/O bank contains global clock input pins to bring user clocks onto the device clock management and routing resources. The global clock inputs bring user clocks onto:

- Clock buffers in the PHY adjacent to the same bank
- CMTs in the PHY adjacent to the same bank

Each device has three global clock buffers: BUFGCTRL, BUFGCE, and BUFGCE_DIV. In addition, there is a local BUFGCE_LEAF clock buffer for driving leaf clocks from horizontal distribution to various blocks in the device. BUFGCTRL has derivative software representations of types BUFGMUX, BUFGMUX1, BUFGMUX_CTRL, and BUFGCE_1. BUFGCE is for glitchless clock gating and has software derivative BUFG (BUFGCE with clock enable tied High). The global clock buffers drive routing and distribution tracks into the FPGA logic via HCS rows. There are 24 routing and 24 distribution tracks in each HCS row. There is also a BUFG_GT that generates divided clocks for GT clocking. The clock buffers:

- Can be used as a clock enable circuit to enable or disable clocks either globally, locally, or within a CR for fine-grained power control.
- Can be used as a glitch-free multiplexer to:
 - select between two clock sources
 - switch away from a failed clock source
- Are often driven by a CMT to:
 - eliminate the clock distribution delay
 - adjust clock delay relative to another clock

[Chapter 2, Clocking Resources](#), has further details on global clocks, I/O and GT clocking. It also describes which clock routing resources to utilize for various applications.

CMT Overview

Each device has a CMT as part of the PHY next to each of the I/O banks. A CMT consists of one MMCM and two PLLs. The MMCM is the primary block for frequency synthesis for a wide range of frequencies, and serves as a jitter filter for either external or internal clocks, and deskew clocks among a wide range of other functions. The PLL's primary purpose is to provide clocking to the PHY I/Os, but can also be used for clocking other resources in the FPGA in a limited fashion. The device clock input connectivity allows multiple resources to provide the reference clock(s) to the MMCM and PLL.

MMCMs have infinite fine phase shift capability in either direction and can be used in dynamic phase shift mode. MMCMs also have a fractional counter in either the feedback path or in one output path, enabling further granularity of frequency synthesis capabilities.

The LogiCORE™ IP clocking wizard is available to assist in utilizing MMCMs and PLLs to create clock networks in UltraScale architecture designs. The GUI interface is used to collect clock network parameters. The clocking wizard chooses the appropriate CMT resource and optimally configures the CMT resource and associated clock routing resources.

[Chapter 3, Clock Management Tile](#), has further details on the CMT block features and connectivity.

Clocking Differences from Previous FPGA Generations

UltraScale architecture-based devices have significant innovations in the clocking architecture. In general, there is a minimal difference between global and local clock buffers. Thus, the 7 series regional clock buffers have been replaced by new clock buffers with more global reach while automatically utilizing local clock buffers for local distribution of the clocks. The CMT block consists of one MMCM and two PLLs. The MMCM is very similar to the 7 series family while the PLL has new features for the I/O PHY clocking, but a reduced set of functionality and connectivity for fabric clocking.

Key Differences from 7 Series FPGAs

- BUFMRs, BUFRs, and BUFIOs, and the associated routing resources have been removed from this architecture and are replaced by new clock buffers, clock routing, and a completely new I/O clocking architecture.
- The BUFGCTRL and its derivatives are still available. Two new global clock buffer resources BUFGCE and BUFGCE_DIV have been introduced in the new architecture. At the local clocking level, a new BUFCE_LEAF clock buffer provides local, vertical clocking with additional features.
- A BUFG_GT buffer for clock division of GT clocks has been added.
- A new and improved clock routing architecture is available. There are now two types of global routing tracks called routing and distribution. Both types of routing provide a segmentable clock network at the CR level. Both types can be driven by the global clock buffers. The distribution tracks can be driven by routing tracks or directly by clock buffer resources. The distribution tracks provide clock connectivity to the various fabric clocking points.
- The CMTs now have two PLLs instead of one.
- MMCMs are similar to the MMCM in the 7 series devices. PLLs have new features related to I/O PHY clocking. However, the fabric-related clocking has reduced functionality and connectivity as compared to the 7 series FPGAs. For example, the PLLs do not support phase compensation or external feedback, have fewer outputs, share a voltage-controlled oscillator (VCO) with the PHY clocking, and have other features removed as compared to the 7 series devices. For this reason, most customers should use the MMCM for general clocking. However, leftover PLLs are also available for use.
- The MMCM output clock frequencies can be dynamically changed without resetting the MMCM.
- The definition of clock region has changed. A clock region no longer spans half a device width in the horizontal direction. UltraScale architecture clock regions have a

rectangular shape with a fixed width and height and are organized in tiles. Horizontal and vertical clock tracks are segmented at the clock region boundaries.

Clocking Resources

UltraScale architecture-based devices have several clock routing resources to support various clocking schemes and requirements, including high fanout, short propagation delay, and extremely low skew. To best utilize the clock routing resources, the designer must understand how to get user clocks from the PCB to the FPGA, decide which clock routing resources are optimal, and then access those clock routing resources by utilizing the appropriate I/O and clock buffers.

Global Clock Inputs

External global user clocks must be brought into the FPGA on differential clock pin pairs called global clock inputs (GC). There are four GC pin pairs in each bank that have direct access to the global clock buffers, MMCMs, and PLLs that are in the CMT adjacent to the same I/O bank. GC inputs provide dedicated, high-speed access to the internal global and regional clock resources. GC inputs use dedicated routing and must be used for clock inputs where the timing of various clocking features is imperative. General-purpose I/O with local interconnects should not be used for clock signals.

Each I/O bank is located in a single clock region and includes 52 I/O pins. Of the 52 I/O pins in each I/O bank in every I/O column, there are four clock-capable input pin pairs (a total of eight pins). Each clock-capable input:

- Can be connected to a differential or single-ended clock on the PCB
- Can be configured for any I/O standard, including differential I/O standards
- Has a P-side (master), and an N-side (slave)

Single-ended clock inputs must be assigned to the P (master) side of the GC input pin pair. If a single-ended clock is connected to the P-side of a differential clock pin pair, the N-side cannot be used as another single-ended clock pin—it can only be used as a user I/O. For pin naming conventions, refer to the *UltraScale Architecture Packaging and Pinout User Guide* (UG575) [Ref 1].

GC inputs can be used as regular I/O if not used as clocks. When used as regular I/O, clock-capable input pins can be configured as any single-ended or differential I/O standard.

GC inputs can connect to the PHY adjacent to the banks they reside in.

Byte Clock Inputs

Byte lane clock (DBC and QBC) input pin pairs are dedicated clock inputs directly driving source synchronous clocks to the bit slices in the I/O banks. In memory applications, these are also known as DQS. For more information, consult the *UltraScale Architecture SelectIO Resources User Guide* (UG571) [Ref 2].

Clock Buffers and Clock Routing

Global clocks are a dedicated network of interconnects specifically designed to reach all clock inputs to the various resources in a device. These networks are designed to have low skew and low duty cycle distortion, low power, and improved jitter tolerance. They are also designed to support very high frequency signals.

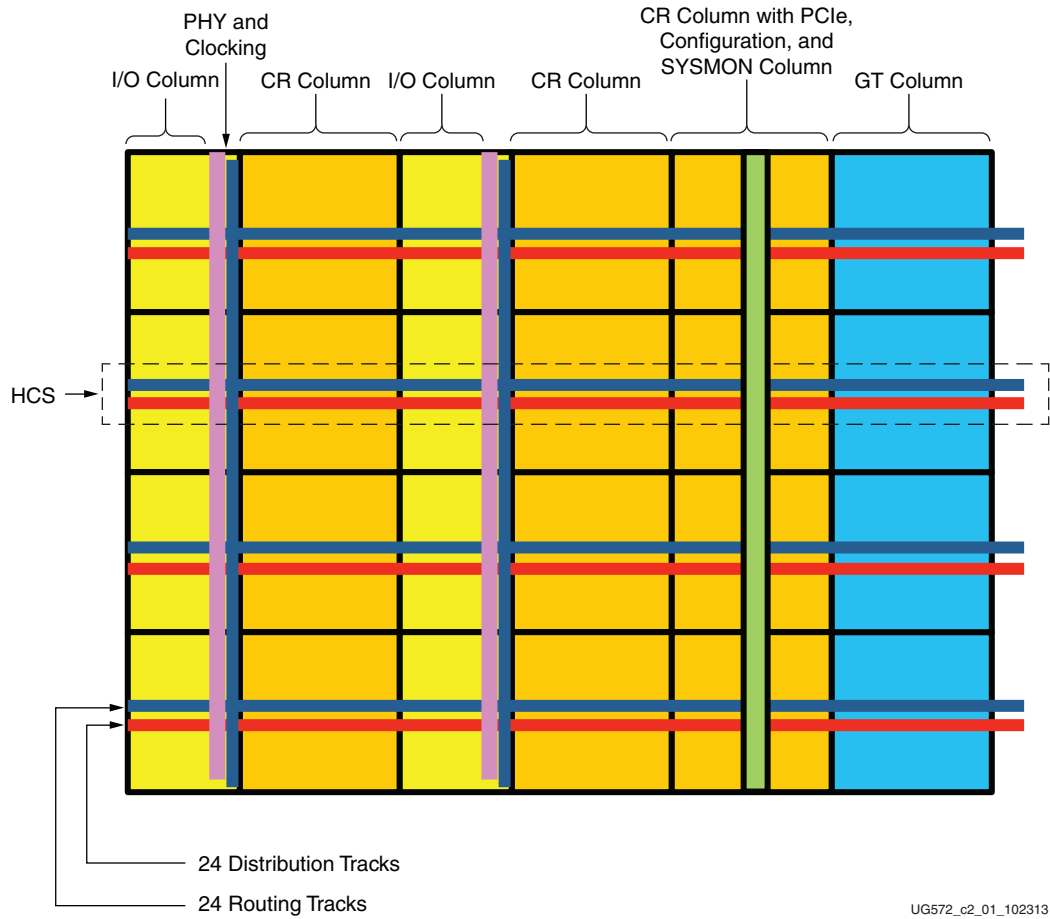
Understanding the signal path for a global clock expands the understanding of the various global clocking resources. The global clocking resources and network consist of these paths and components:

- [Clock Structure, page 11](#)
- [Clock Buffers, page 14](#)
- [BUFGCTRL Clock Buffer Primitives, page 15](#)
- [Additional BUFGCTRL Use Models, page 23](#)
- [BUFGCE Clock Buffers, page 25](#)
- [BUFG Clock Buffer, page 26](#)
- [BUFCE_LEAF Clock Buffer, page 26](#)

Clock Structure

The basic fabric is composed of blocks of CRs. CRs are organized in a tile fashion and thus build columns and rows. Each CR contains slices (CLBs), DSPs, and 36K block RAM blocks. The mix of slice, DSP, and block RAM columns in each CR can be different, but are always identical when stacked in the vertical direction thus building columns of those resources for the entire device. I/O and GT columns are then inserted with columns of CRs. In addition, there is a single column that contains the configuration logic, SYSMON, and PCIe blocks. An HCS runs horizontally through the device in the center of each row of CRs, I/Os, and GTs. The HCS contains the horizontal routing and distribution tracks as well as leaf clock buffers and clock network interconnects between horizontal/vertical routing and distribution. Vertical tracks of routing and distribution connect all CRs in a column while vertical routing spans an entire I/O column. There are 24 horizontal routing and 24 distribution tracks

(Figure 2-1), and 24 vertical routing and 24 distribution tracks (Figure 2-2).



UG572_c2_01_102313

Figure 2-1: Horizontal Clocking

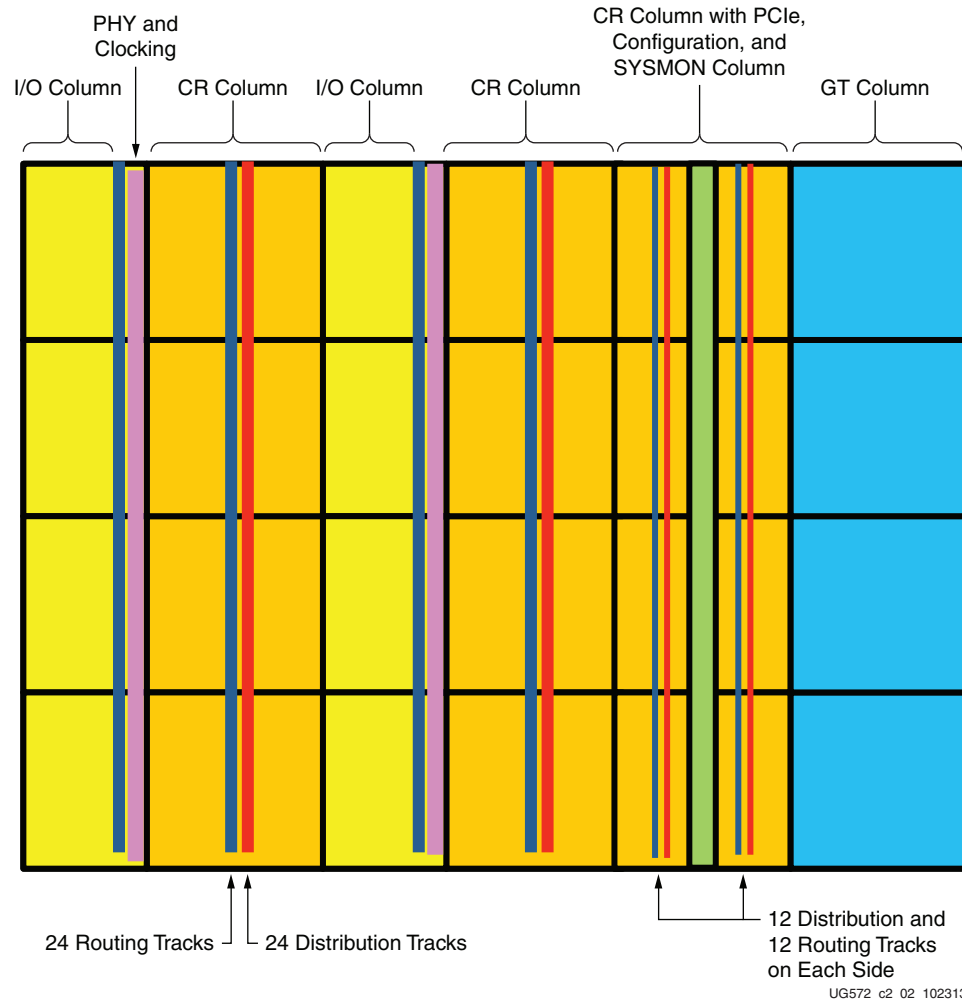


Figure 2-2: Vertical Clocking

The clocks can be distributed from their sources in one of two ways (Figure 2-3):

- The clocks can go onto routing tracks that take the clocks to a central point in a CR without going to any loads. The clocks can then drive the distribution tracks unidirectionally from which the clock networks fan out. In this way, the clock buffers can drive to a specific point in the CRs from which the clock buffers travel vertically and then horizontally on the distribution tracks to drive the clocking points. The clocking points are driven via leaf clocks with clock enable (CE) in that CR and adjacent CRs, if needed. Distribution tracks cannot drive routing tracks.

This distribution scheme is used to move the root for all the loads to be at a specific location for improved, localized skew. Furthermore, both routing and distribution tracks can drive into horizontally or vertically adjacent CRs in a segmented fashion. Routing tracks can drive both routing and distribution tracks in the adjacent CRs while the distribution tracks can drive other horizontal distribution tracks in adjacent CRs. The CR boundary segmentation allows construction of either truly global, device-wide clock networks or more local clock networks of variable sizes by reusing clocking tracks.

- Alternatively, clock buffers can drive straight onto the distribution tracks and distribute the clock in that manner. This reduces the clock insertion delay.

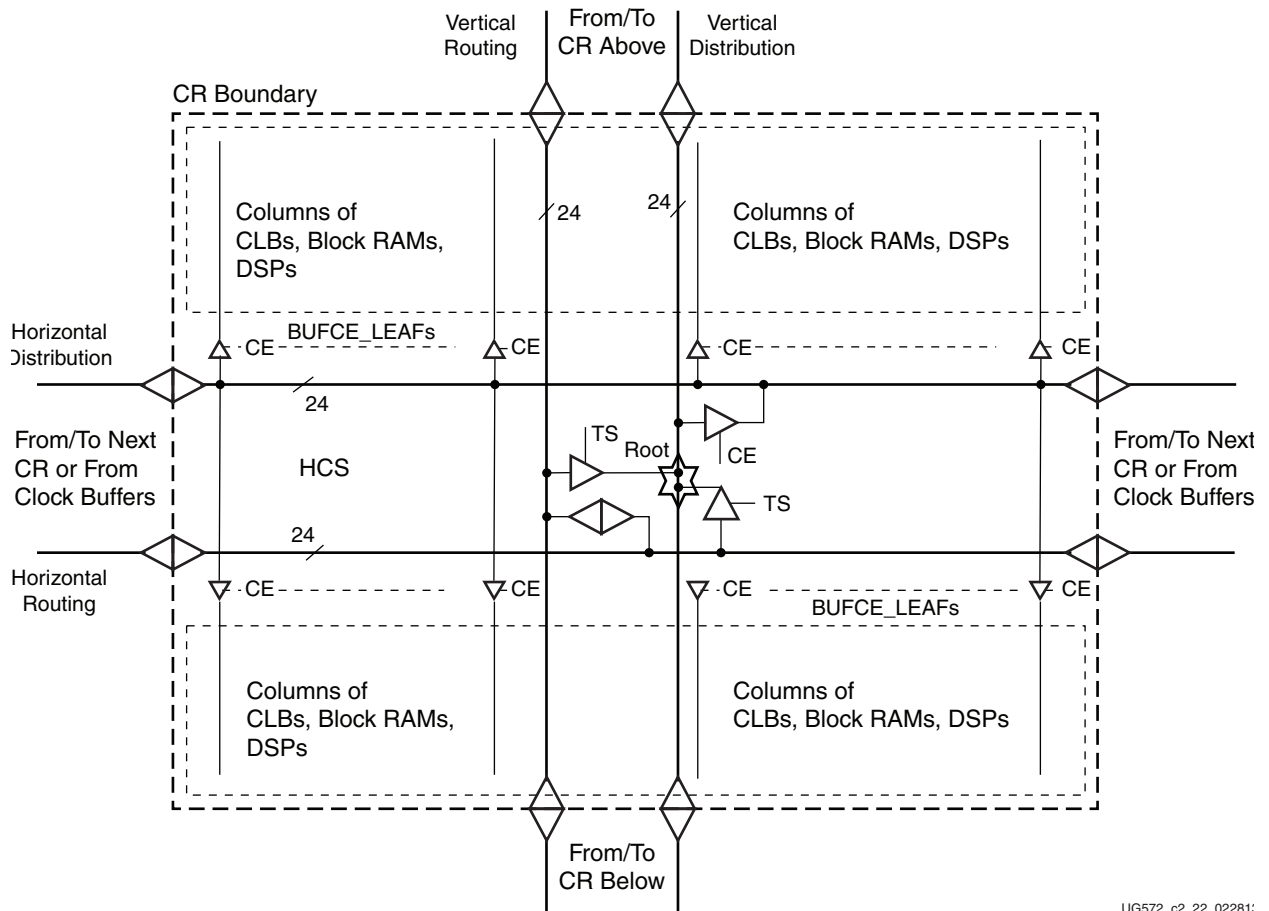
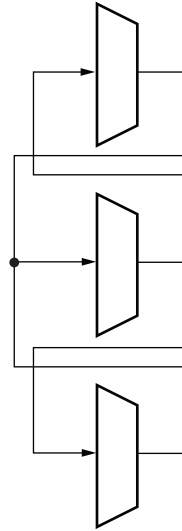


Figure 2-3: Clock Region Clcking

Clock Buffers

The PHY global clocking contains several sets of BUFGCTRLs, BUFGCEs, and BUFGCE_DIVs. Each set can be driven by four GC pins from the adjacent bank, MMCMs, PLLs in the same PHY, and interconnect. The clock buffers then drive the routing and distribution resources across the entire device. Each PHY contains 24 BUFGCEs, 8 BUFGCTRLs, and 4 BUFGCE_DIVs but only 24 of them can be used at the same time.

In the clocking architecture, BUFGCTRL multiplexers and all derivatives can be cascaded to adjacent clock buffers, effectively creating a ring of eight BUFGMUXes (BUFGCTRL multiplexers). Figure 2-4 shows a simplified diagram of cascading BUFGCTRLs.



UG572_c2_03_110112

Figure 2-4: Cascading BUFCTRLs

The following subsections detail the various configurations, primitives, and use models of the clock buffers.

BUFCTRL Clock Buffer Primitives

The primitives in Table 2-1 are different configurations of the clock BUFCTRL buffers. The Vivado® tools manage the configuration of all these primitives, and the *Vivado Design Suite User Guide: Using Constraints* (UG903) [Ref 3] describes the LOC constraint.

Table 2-1: BUFCTRL Clock Buffer Primitives

Primitive	Input	Output	Control
BUFCTRL	I0, I1	O	CE0, CE1, IGNORE0, IGNORE1, S0, S1
BUFGCE_1	I	O	CE
BUFGMUX	I0, I1	O	S
BUFGMUX_1	I0, I1	O	S
BUFGMUX_CTRL	I0, I1	O	S

BUFCTRL

The BUFCTRL primitive shown in Figure 2-5 can switch between two asynchronous clocks. All other global clock buffer primitives are derived from certain configurations of BUFCTRL.

BUFCTRL has four select lines, S0, S1, CE0, and CE1. It also has two additional control lines, IGNORE0 and IGNORE1. These six control lines are used to control the inputs I0 and I1.

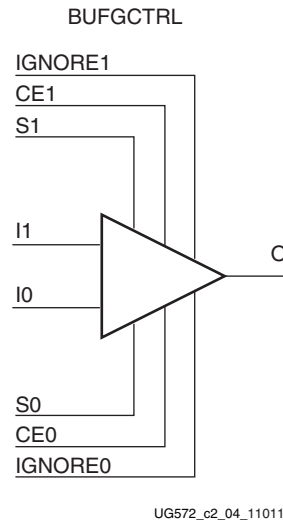


Figure 2-5: BUFGCTRL Primitive

BUFGCTRL is designed to switch between two clock inputs without the possibility of a glitch. When the presently selected clock transitions from High to Low after S0 and S1 change, the output is kept Low until the other (to-be-selected) clock transitions from High to Low. Then, the new clock starts driving the output. The default configuration for BUFGCTRL is falling-edge sensitive and held at Low prior to the input switching. BUFGCTRL can also be rising-edge sensitive and held at High prior to the input switching by using the INIT_OUT attribute.

In some applications, the conditions previously described are not desirable. Asserting the IGNORE pins bypasses the BUFGCTRL from detecting the conditions for switching between two clock inputs. In other words, asserting IGNORE causes the MUX to switch the inputs at the instant the select pin changes. IGNORE0 causes the output to switch away from the I0 input immediately when the select pin changes, while IGNORE1 causes the output to switch away from the I1 input immediately when the select pin changes.

Selection of an input clock requires a “select” pair (S0 and CE0, or S1 and CE1) to be asserted High. If either S or clock enable (CE) is not asserted High, the desired input is not selected. In normal operation, both S and CE pairs (all four select lines) are not expected to be asserted High simultaneously. Typically, only one pin of a “select” pair is used as a select line, while the other pin is tied High. The truth table is shown in Table 2-2.

Table 2-2: Truth Table for Clocking Resources

CE0	S0	CE1	S1	O
1	1	0	X	I0
1	1	X	0	I0
0	X	1	1	I1
X	0	1	1	I1

Table 2-2: Truth Table for Clocking Resources (Cont'd)

CE0	S0	CE1	S1	O
1	1	1	1	Old Input ⁽¹⁾

Notes:

1. Old input refers to the valid input clock before this state is achieved.
2. For all other states, the output becomes the value of INIT_OUT and does not toggle.

Although both S and CE are used to select a desired output, only S is suggested for glitch-free switching. This is because when using CE to switch clocks, the change in clock selection can be faster than when using S. A violation in the setup/hold time of the CE pins causes a glitch at the clock output. On the other hand, using the S pins allows the user to switch between the two clock inputs without regard to setup/hold times. As a result, using S to switch clocks does not result in a glitch. See [BUFGMUX_CTRL](#), page 21.

The timing diagram in [Figure 2-6](#) illustrates various clock switching conditions using the BUFGCTRL primitives. Exact timing numbers are best found using the speed specification.

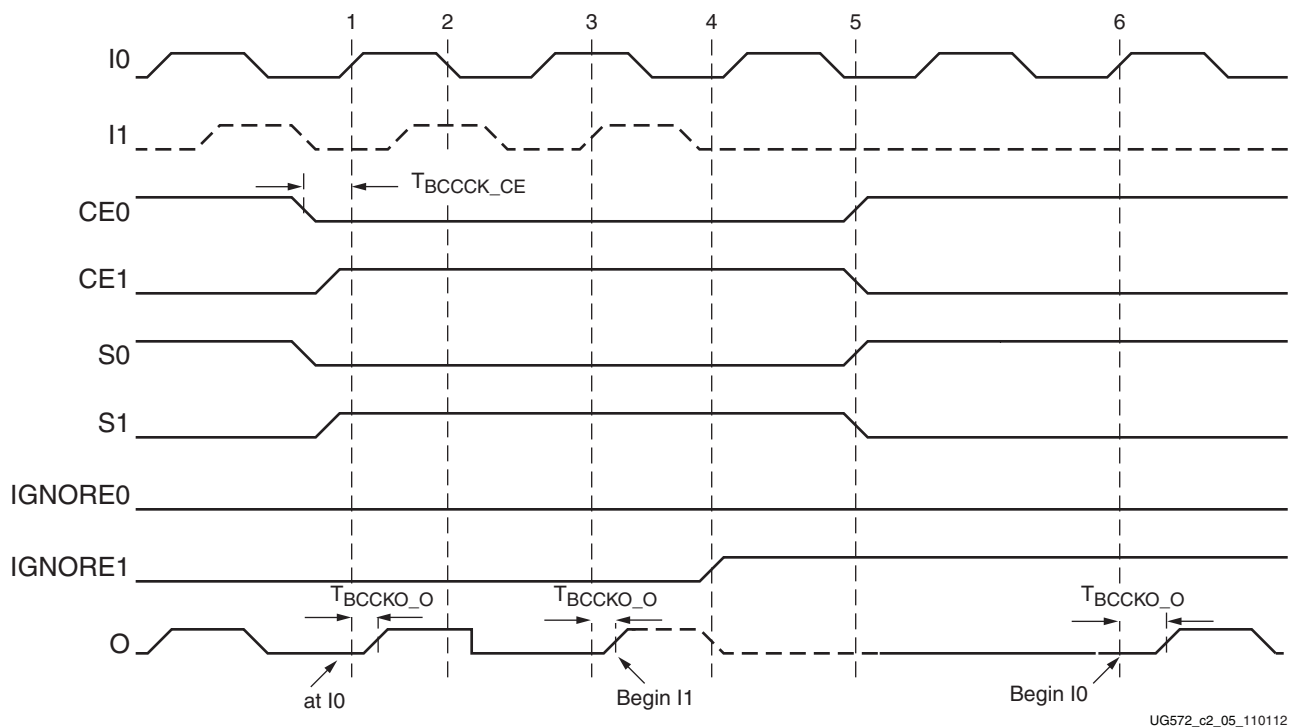


Figure 2-6: BUFGCTRL Timing Diagram

- Before time event 1, output O uses input I0.
- At time T_{BCCCK_CE}, before the rising edge at time event 1, both CE0 and S0 are deasserted Low. At about the same time, both CE1 and S1 are asserted High.
- At time T_{BCCCKO_O}, after time event 3, output O uses input I1. This occurs after a High-to-Low transition of I0 (event 2) followed by a High-to-Low transition of I1.

- At time event 4, IGNORE1 is asserted.
- At time event 5, CE0 and S0 are asserted High while CE1 and S1 are deasserted Low. At TBCKO_O, after time event 6, output O has switched from I1 to I0 without requiring a High-to-Low transition of I1.

Other capabilities of BUFGCTRL are:

- Pre-selection of the I0 and I1 inputs are made after configuration but before device operation.
- The initial output after configuration can be selected as either High or Low.
- Clock selection using CE0 and CE1 only (S0 and S1 tied High) can change the clock selection without waiting for a High-to-Low transition on the previously selected clock.

Table 2-3 summarizes the attributes for the BUFGCTRL primitive.

Table 2-3: BUFGCTRL Attributes

Attribute Name	Description	Possible Values
INIT_OUT	Initializes the BUFGCTRL output to the specified value after configuration. Sets the positive or negative edge behavior. Sets the output level when changing clock selection.	0 (default), 1
PRESELECT_I0	If TRUE, BUFGCTRL output uses the I0 input after configuration. ⁽¹⁾	FALSE (default), TRUE
PRESELECT_I1	If TRUE, BUFGCTRL output will use the I1 input after configuration. ⁽¹⁾	FALSE (default), TRUE

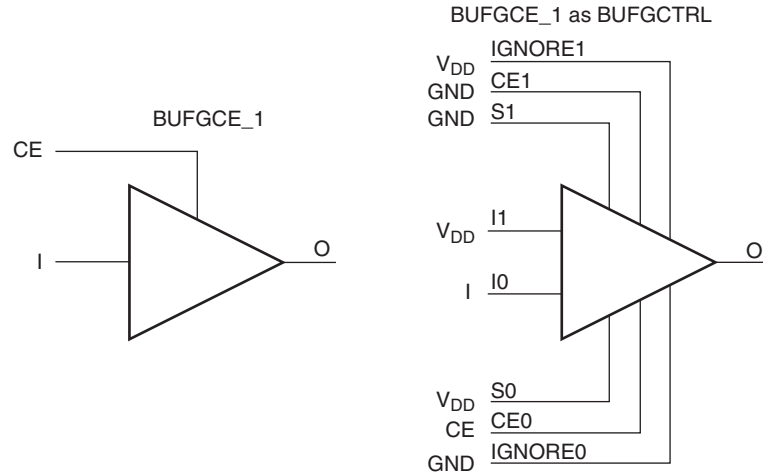
Notes:

1. Both PRESELECT attributes cannot be TRUE at the same time.

BUFGCE_1

BUFGCE_1 is a clock buffer with one clock input, one clock output, and a clock enable line. This primitive is based on BUFGCTRL with some pins connected to logic High or Low.

Figure 2-7 illustrates the relationship of BUFGCE and BUFGCTRL. The LOC constraint is available for manually placing the BUFGCE_1 location. See the *Vivado Design Suite User Guide: Using Constraints* (UG903) [Ref 3] for more information.



UG572_c2_08_022713

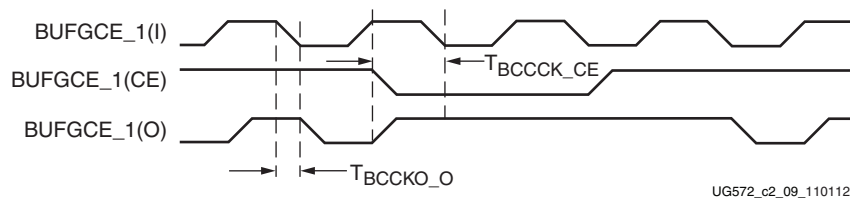
Figure 2-7: **BUFGE_1 as BUFCTRL**

The switching condition for BUFGE_1 is similar to BUFCTRL with INIT_OUT set to 1. If the CE input is Low prior to the incoming falling clock edge, the following clock pulse does not pass through the clock buffer, and the output stays High. Any level change of CE during the incoming clock Low pulse has no effect until the clock transitions High. The output stays High when the clock is disabled. However, when the clock is being disabled, it completes the clock Low pulse.



IMPORTANT: Because the clock enable line uses the CE pin of the BUFCTRL, the select signal must meet the setup time requirement. Violating this setup time can result in a glitch.

Figure 2-8 illustrates the timing diagram for BUFGE_1.



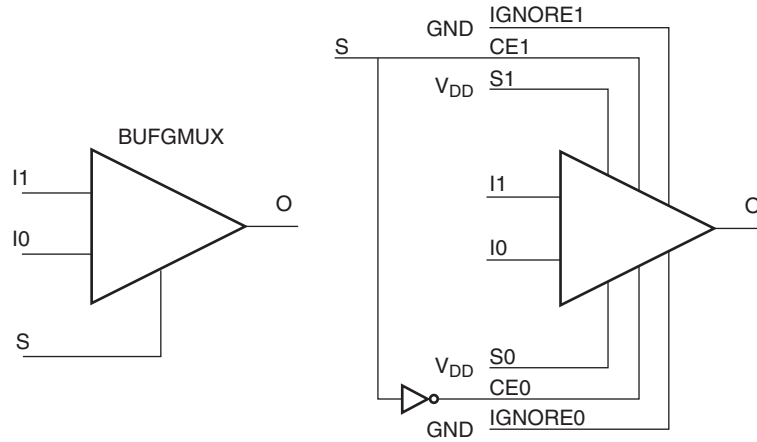
UG572_c2_09_110112

Figure 2-8: **BUFGE_1 Timing Diagram**

BUFGMUX and BUFGMUX_1

BUFGMUX is a clock buffer with two clock inputs, one clock output, and a select line. This primitive is based on BUFCTRL with some pins connected to logic High or Low.

Figure 2-9 illustrates the relationship of BUFGMUX and BUFCTRL. The LOC constraint is available for manually placing the BUFGMUX and BUFCTRL locations. See the *Vivado Design Suite User Guide: Using Constraints* (UG903) [Ref 3] for more information.



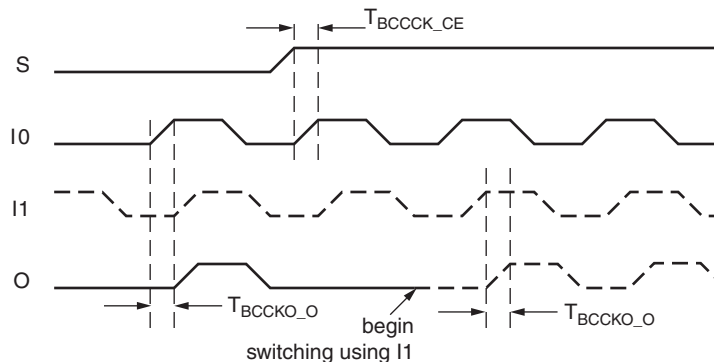
UG572_c2_10_110112

Figure 2-9: BUFGMUX as BUFGCTRL



IMPORTANT: Because BUFGMUX uses the CE pins as select pins, when using the select, the setup time requirement must be met. Violating this setup time can result in a glitch.

Switching conditions for BUFGMUX are the same as the CE pins on BUFGCTRL. Figure 2-10 illustrates the timing diagram for BUFGMUX.



UG572_c2_11_110112

Figure 2-10: BUFGMUX Timing Diagram

In Figure 2-10:

- The current clock is I0.
- S is activated High.
- If I0 is currently High, the multiplexer waits for I0 to deassert Low.
- After I0 is Low, the multiplexer output stays Low until I1 transitions from High to Low.
- When I1 transitions from High to Low, the output switches to I1.
- If setup/hold times are met, no glitches or short pulses can appear on the output.

BUFGMUX_1 is rising-edge sensitive and held at High prior to input switch. [Figure 2-11](#) illustrates the timing diagram for BUFGMUX_1. The LOC constraint is available for manually placing the BUFGMUX and BUFGMUX_1 locations. See the *Vivado Design Suite User Guide: Using Constraints* (UG903) [\[Ref 3\]](#) for more information.

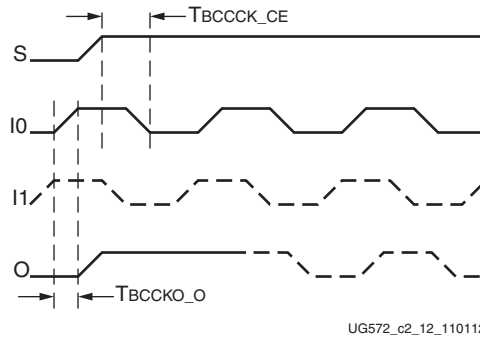


Figure 2-11: BUFGMUX_1 Timing Diagram

In [Figure 2-11](#):

- The current clock is I0.
- S is activated High.
- If I0 is currently Low, the multiplexer waits for I0 to be asserted High.
- After I0 is High, the multiplexer output stays High until I1 transitions from Low to High.
- When I1 transitions from Low to High, the output switches to I1.
- If setup/hold times are met, no glitches or short pulses can appear on the output.

[Table 2-4](#) summarizes the attributes for the BUFGMUX primitive.

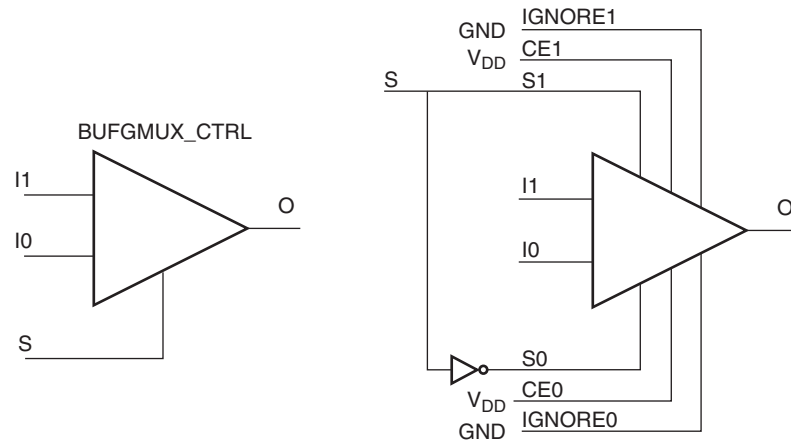
Table 2-4: BUFGMUX Attributes

Attribute Name	Description	Possible Values
CLK_SEL_TYPE	Specifies synchronous or asynchronous clock switching.	SYNC (default), ASYNC

BUFGMUX_CTRL

BUFGMUX_CTRL is a clock buffer with two clock inputs, one clock output, and a select line. This primitive is based on BUFGCTRL with some pins connected to logic High or Low.

[Figure 2-12](#) illustrates the relationship of BUFGMUX_CTRL and BUFGCTRL.



UG572_c2_13_110112

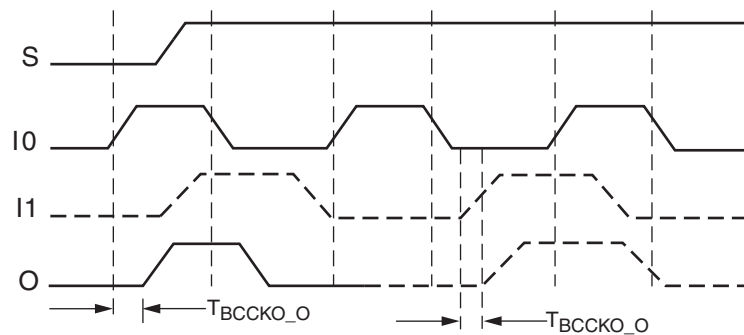
Figure 2-12: **BUFGMUX_CTRL as BUFGCTRL**

BUFGMUX_CTRL uses the S pins as select pins. S can switch anytime without causing a glitch. The setup/hold time on S is for determining whether the output passes an extra pulse of the previously selected clock before switching to the new clock. If S changes as shown in Figure 2-13 prior to the setup time T_{BCKCK_S} and before I0 transitions from High to Low, the output does not pass an extra pulse of I0. If S changes following the hold time for S, the output passes an extra pulse. If S violates the setup/hold requirements, the output might pass the extra pulse but it will not glitch. In any case, the output changes to the new clock within three clock cycles of the slower clock.

The setup/hold requirements for S0 and S1 are with respect to the falling clock edge, not the rising edge as for CE0 and CE1.

Switching conditions for BUFGMUX_CTRL are the same as the S pin of BUFGCTRL.

Figure 2-13 illustrates the timing diagram for BUFGMUX_CTRL.



UG572_c2_14_031513

Figure 2-13: **BUFGMUX_CTRL Timing Diagram**

Other capabilities of the BUFGMUX_CTRL primitive are:

- I0 and I1 inputs can be preselected after configuration.

- Initial output can be selected as High or Low after configuration.

Additional BUFCTRL Use Models

Asynchronous MUX Using BUFCTRL

In some cases an application requires immediate switching between clock inputs or bypassing the edge sensitivity of BUFCTRL. An example is when one of the clock inputs is no longer switching. If this happens, the clock output would not have the proper switching conditions because the BUFCTRL never detected a clock edge. This case uses the asynchronous MUX. Figure 2-14 illustrates an asynchronous MUX with BUFCTRL design example.

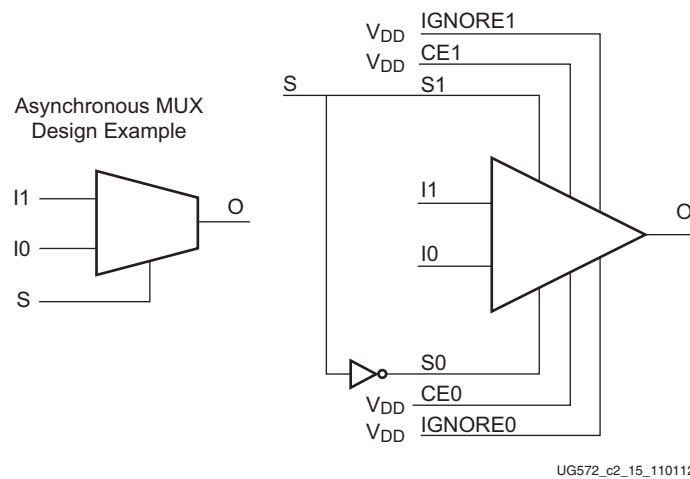


Figure 2-14: Asynchronous MUX with BUFCTRL Design Example

Figure 2-15 shows the asynchronous MUX timing diagram.

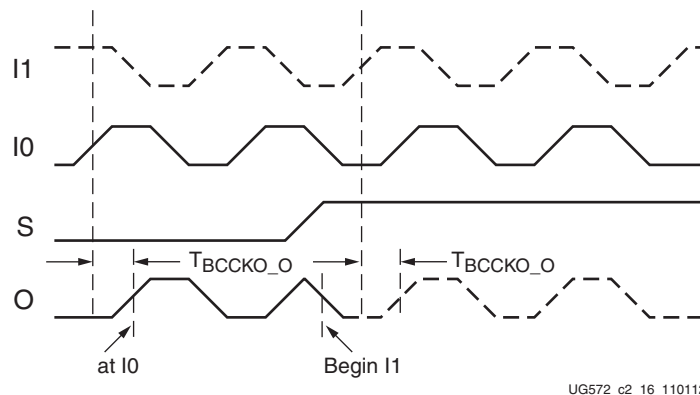


Figure 2-15: Asynchronous MUX Timing Diagram

In Figure 2-15:

- The current clock is from I0.

- S is activated High.
- The clock output immediately switches to I1.
- When ignore signals are asserted High, glitch protection is disabled.

BUFGMUX_CTRL with a Clock Enable

A BUFGMUX_CTRL with a clock enable BUFGCTRL configuration allows the user to choose between the incoming clock inputs. If needed, the clock enable is used to disable the output. Figure 2-16 illustrates the BUFGCTRL usage design example and Figure 2-17 shows the timing diagram.

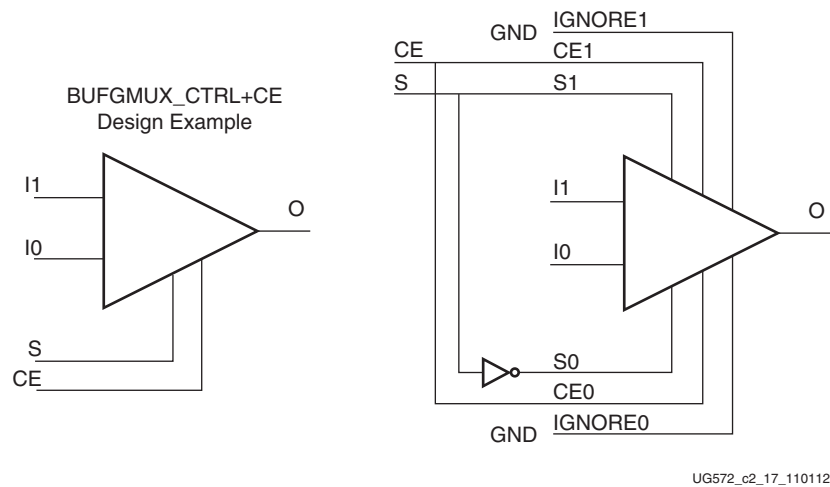


Figure 2-16: BUFGMUX_CTRL with a CE and BUFGCTRL

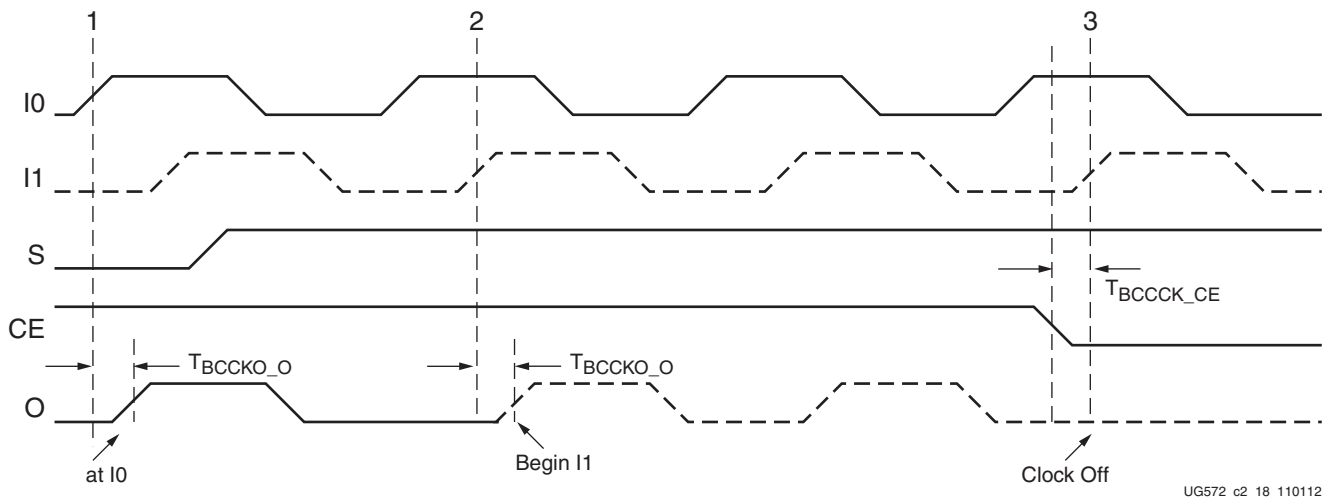


Figure 2-17: BUFGMUX_CTRL with a CE Timing Diagram

In Figure 2-17:

- At time event 1, output O uses input I0.

- Before time event 2, S is asserted High.
- At time TBCKO_O, after time event 2, output O uses input I1. This occurs after a High-to-Low transition of IO followed by a High-to-Low transition of I1 is completed.
- At time TBCKCK_CE, before time event 3, CE is asserted Low. To avoid any output clock glitches, the clock output is switched Low and kept at Low until after a High-to-Low transition of I1 is completed.

BUFGCE Clock Buffers

BUFGCE is a clock buffer with one clock input, one clock output, and a clock enable line (Figure 2-18). This buffer provides glitchless clock gating. BUFGCE can directly drive the routing resources and is a clock buffer with a single gated input. Its O output is 0 when CE is Low (inactive). When CE is High, the I input is transferred to the O output.

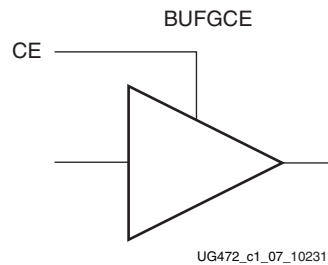


Figure 2-18: BUFGCE Buffer Type

Table 2-5 lists the BUFGCE pins.

Table 2-5: BUFGCE Pins

Pin Name	Type	Invertible	Description
CE	Input	TRUE	Clock enable
I	Input	FALSE	Clock buffer
O	Output	FALSE	Clock buffer

Table 2-6 shows the BUFGCE attributes.

Table 2-6: BUFGCE Attributes

Attribute Name	Values	Default	Type	Description
CE_TYPE	SYNC, ASYNC	SYNC	STRING	Sets the clock enable behavior where SYNC allows for glitchless transition while ASYNC allows immediate transition.

Figure 2-19 shows the BUFGCE timing diagram.

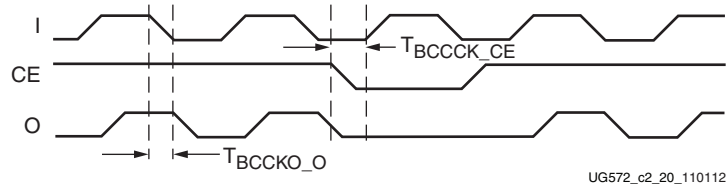


Figure 2-19: BUFGCE Timing Diagram

BUFG Clock Buffer

BUFG is a clock buffer with one clock input and one clock output. This primitive is based on BUFGCE with the CE pin connected to High, as shown in Figure 2-20.

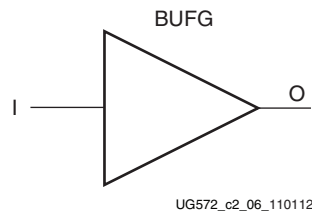


Figure 2-20: BUFG

BUFCE_LEAF Clock Buffer

BUFCE_LEAF is a clock buffer with CE for leaf driving off horizontal HCS row. This buffer is an interconnect leaf clock buffer driving the clocking point of the various blocks with a single gated input. Its O output is 0 when CE is Low (inactive). When CE is High, the I input is transferred to the O output. Table 2-7 shows the BUFCE_LEAF attributes.

Table 2-7: BUFCE_LEAF Attributes

Attribute Name	Values	Default	Type	Description
CE_TYPE	SYNC, ASYNC	SYNC	STRING	Sets the clock enable behavior where SYNC allows for glitchless transition while ASYNC allows immediate transition.

BUFGCE_DIV

BUFGCE_DIV is a clock buffer with one clock input (I), one clock output (O), one clear input (CLR) and a clock enable (CE) input. BUFGCE_DIV can directly drive the routing and distribution resources and is a clock buffer with a single gated input and a reset. Its O output is 0 when CE is Low (inactive). When CE is High, the I input is transferred to the O output. CE is synchronous to the clock for glitch-free operation. CLR is an asynchronous

reset assertion and synchronous reset deassertion to this buffer. BUFGCE_DIV can also divide the input clock by 1 to 8.

When CLR (reset) is deasserted, the output clock transitions High on the next input clock edge delayed by the divide value. The output clock then toggles at the divided frequency. When CLR is asserted, the clock stops toggling after some clock-to-out time. For an odd divide, the duty cycle is not 50% because the clock is High one cycle less than it is Low. For example, for a divide value of 7, the clock is High for 3 cycles and Low for 4 cycles.

When CE is deasserted, the output stops at its current state, High or Low. When CE is reasserted, the internal counter restarts from where it stopped. For example, if the divide value is 8 and CE is deasserted two input clock cycles after the last output High transition, the output stays High. Then when CE is reasserted, the output transitions Low after two input clock cycles. If the reset input is used, upon assertion the output transitions Low immediately if the current output is High, otherwise it stays Low.

Since reset is synchronously de-asserted, when reset is de-asserted in the above example, the output transitions high at the next input clock edge and transitions low 4 input clock cycles later.

Table 2-8 shows the BUFGCE_DIV pins.

Table 2-8: BUFGCE_DIV Pins

Pin Name	Type	Invertible	Description
I	Input	FALSE	Clock input
CLR	Input	TRUE	Reset
CE	Input	TRUE	Clock Enable
O	Output	FALSE	Clock output

Table 2-9 shows the BUFGCE_DIV attributes.

Table 2-9: BUFGCE_DIV Attributes

Attribute Name	Values	Default	Type	Description
BUFGCE_DIVIDE	1, 2, 3, 4, 5, 6, 7, 8	1	STRING	Defines whether the output clock is a divided version of the input clock.

BUFG_GT

BUFG_GT (Figure 2-21) is a clock buffer with one clock input (I), one clock output (O), one clear input (CLR) with CLR mask input (CLRMASK), a clock enable (CE) input with a CE mask input (CEMASK) and a 3-bit divide (DIV[2:0]) input. This buffer can directly drive the routing and distribution resources and is a clock buffer with a single gated input and a reset. When CE is deasserted (Low) the output stops at its current state, High or Low. When CE is High, the I input is transferred to the O output. CE is automatically synchronized to the clock for glitch-free operation. CLR is an asynchronous reset assertion and synchronous reset

deassertion to this buffer. The synchronizers have two stages, and if a deterministic latency is required, the setup/hold times need to be met. This buffer can also divide the input clock by 1 to 8. The DIV[2:0] value is the actual divide minus 1 (i.e., 3'b000 corresponds to 1 while 3'b111 corresponds to 8). The divide value (DIV inputs), ENMASK, and CLRMASK must be changed while the buffer is held in reset. The input clock is allowed to change while CE is deasserted or reset is asserted. However, there is a minimum deassertion/assertion time for those control signals.

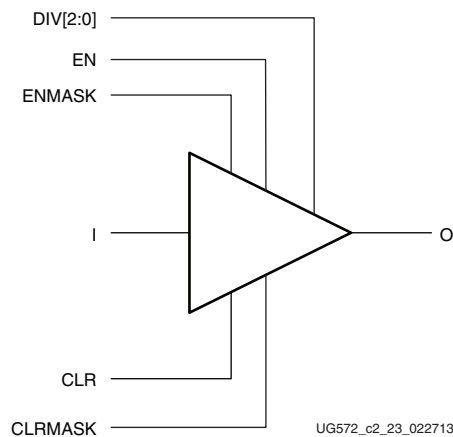


Figure 2-21: BUFG_GT Primitive

UltraScale architecture-based devices have 24 BUFG_GT per GT Quad. Any of 10 GT output clocks can be multiplexed to the 24 BUFG_GT. Corresponding to the 10 clocks are 10 CE and CLR sources that can control the 24 BUFG_GT. Each of these buffers has an individual mask for both CE and CLR (24). All BUFG_GT driven by the same clock source must also have a common CE and CLR signal. Tying off CE and CLR to a constant signal in this case is not allowed, but a mask can be set to provide the same functionality. The output clocks of the BUFG_GT connected to the same input clock are synchronized (phase aligned) to each other when coming out of reset (CLR) or on CE assertion. Individual mask pins can be used to control which BUFG_GT(s) out of the group of 24 respond to CE and CLR and therefore are synchronized to each other or retain their previous phase and divide value. These clock buffers are located in the HCS and are directly driven by the GT output clocks. Their purpose is to directly drive the fabric via routing and distribution resources. GTs have no other direct, dedicated connections to other clock resources. However, they can connect to the CMT via the BUFG_GT and the clock routing resources.

When CLR (reset) is deasserted, the output transitions High at the next input clock edge and transitions Low divide_value/2 input clock cycles later. The output transitions High divide-value clock cycles later, after which the output clock toggles at the divided frequency. When CLR is asserted, the clock stops toggling at Low after some clock-to-out time. For an odd divide, the duty cycle is not 50% because the clock is High one cycle less than it is Low. For example, for a divide value of 7, the clock is High for 3 cycles and Low for 4 cycles.

When CE is deasserted, the output stops at its current state, High or Low. When CE is reasserted, the internal counter restarts from where it stopped. For example, if the divide value is 8 and CE is deasserted two input clock cycles after the last output High transition, the output stays High. Then, when CE is reasserted, the output transitions Low four input clock cycles later (two for synchronization and two to complete the High time period of the output clock because of being a divide by 8). If the reset input is used, upon assertion the output transitions Low immediately if the current output is High, otherwise it stays Low. Because reset is synchronously deasserted, when reset is deasserted in the above example, the output transitions High two input clock cycles later due to synchronization and transitions Low four input clock cycles after that (divide by 8).

The mask pins (CEMASK and CLRMASK) control how a specific, single BUFG_GT responds to the CE/CLR control inputs. When a mask pin is deasserted, its respective control pin has their normal function. When a mask pin is asserted, the respective control pin is ignored, in effect allowing the clock to propagate through (i.e., CE is effectively High and reset is effectively Low). The internal synchronizers phase align the clock outputs of the BUFG_GTs that are not masked. Both edges of CE are synchronized while only the deassertion of reset is synchronized. Assertion of reset immediately causes the output of the BUFG_GT to go Low if it was previously High. This can cause a potential glitch or runt pulse. If this is not acceptable, CE should be used to stop the output. A reset should then be asserted after two input clock cycles plus half the "divide value." This ensures that the output clock High time (if the output clock happened to be disabled High) is no less than normal.



IMPORTANT: *While the synchronizers ensure that all BUFG_GTs driven by the same clock come out of reset in phase, they might not be in phase with BUFG_GTs that have not been reset (i.e., that have their reset mask asserted).*

Clock Management Tile

Introduction

In UltraScale architecture-based devices, the clock management tile (CMT) includes a mixed-mode clock manager (MMCM) and two phase-locked loops (PLLs). The main purpose of the PLL is to generate clocking for the I/Os. But it also contains a limited subset of the MMCM functions for fabric use.

The clock input connectivity allows multiple resources to provide the reference clock(s) to the MMCM. The number of output counters (dividers) is eight, with some of them capable of driving out an inverted clock signal (180° phase shift). MMCMs have infinite fine phase shift capability in either direction and can be used in dynamic phase shift mode. The resolution of the fine phase shift depends on the voltage-controlled oscillator (VCO) frequency. Fractional divide functionality in increments of 1/8th (0.125) for CLKFBOUT and CLKOUT0 are available to support greater clock frequency synthesis capability. UltraScale architecture-based devices have a spread spectrum capability. If the MMCM spread spectrum feature is not used, a spread spectrum on an external input clock will not be filtered and thus passed on to the output clock.

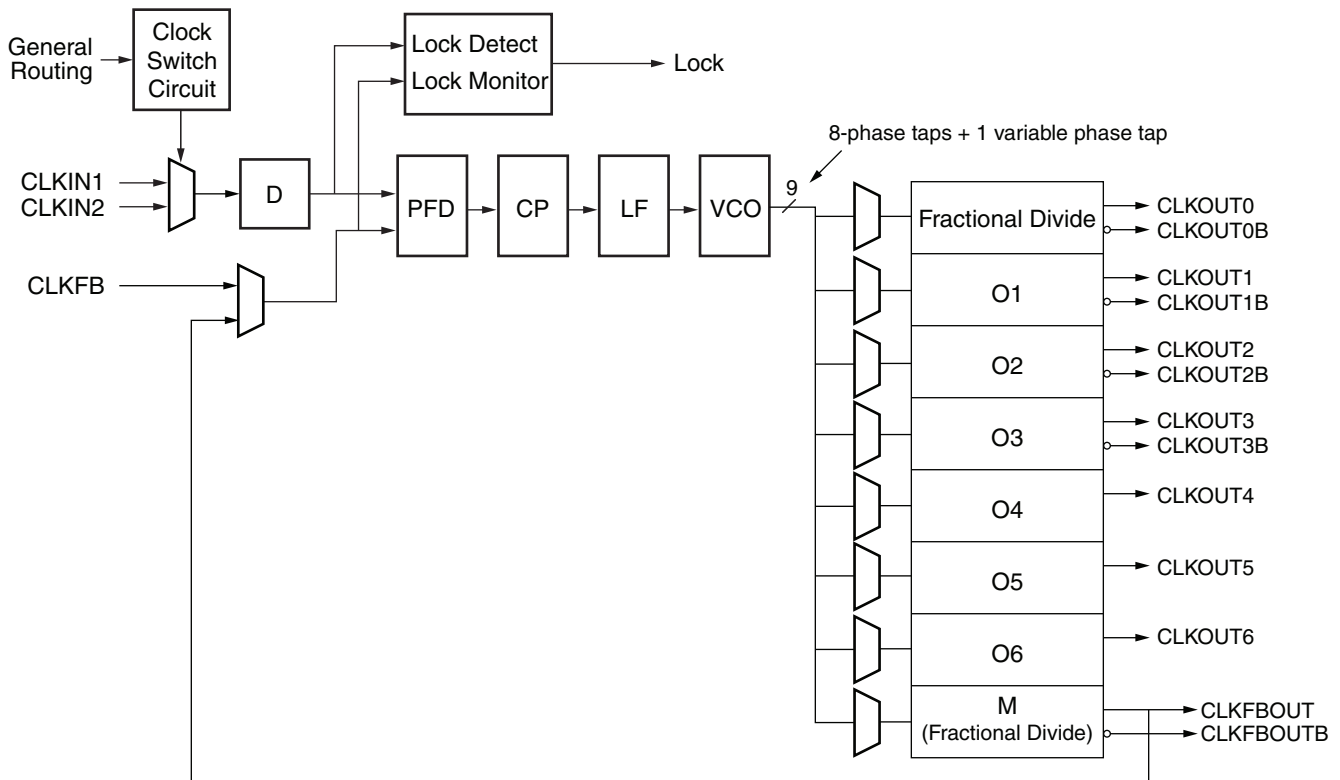
MMCMs

UltraScale architecture-based devices contain one CMT per I/O bank. The MMCMs serve as frequency synthesizers for a wide range of frequencies, and as jitter filters for either external or internal clocks, and deskew clocks.

Input multiplexers select the reference and feedback clocks from either the global clock I/Os or the clock routing or distribution resources. Each clock input has a programmable counter divider (D). The phase-frequency detector (PFD) compares both phase and frequency of the rising edges of both the input (reference) clock and the feedback clock. If a minimum High/Low pulse is maintained, the duty cycle is ancillary. 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 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 a frequency that is too high, the PFD activates a down signal causing the control

voltage to be reduced, thus decreasing the VCO operating frequency. When the VCO operates at a frequency that is too low, an up signal increases voltage. The VCO produces eight output phases and one variable phase for fine-phase shifting. Each output phase can be selected as the reference clock to the output counters (Figure 3-1). 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 MMCM, allowing a wide range of frequency synthesis.

In addition to integer divide output counters, MMCMs add a fractional counter for CLKOUT0 and CLKFBOUT.



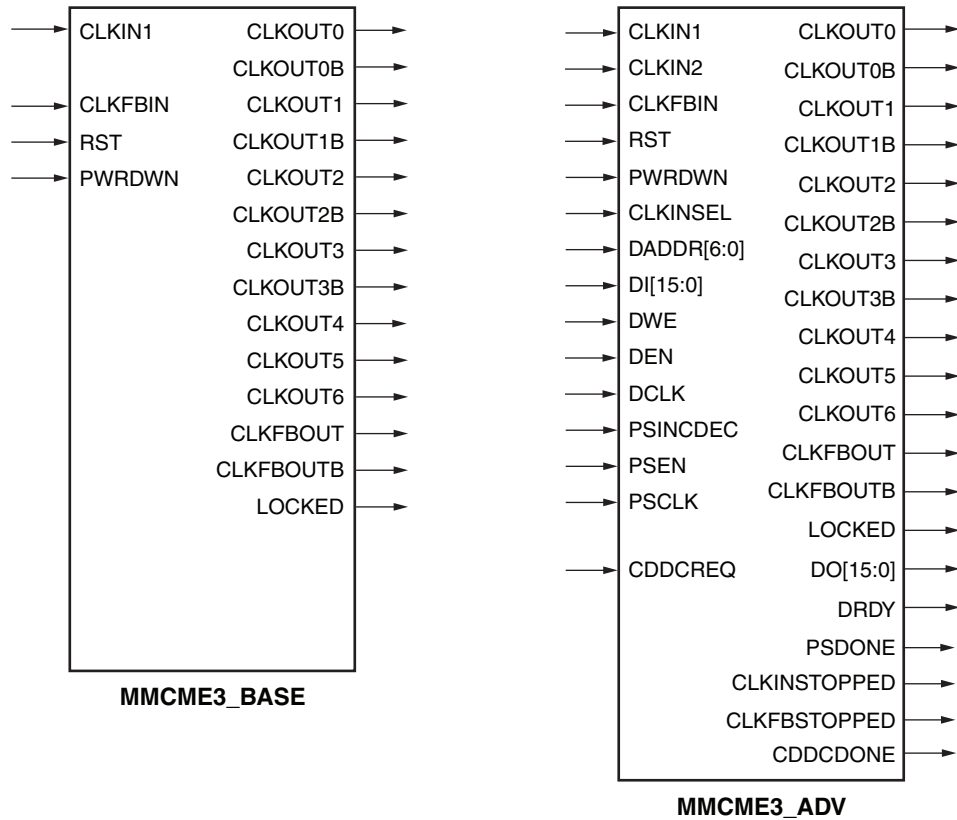
UG572_c3_01_110112

Figure 3-1: Detailed MMCM Block Diagram

General Usage Description

MMCM Primitive

The two MMCM primitives, MMCME3_BASE and MMCME3_ADV, are shown in Figure 3-2.



ug572_c3_02_102313

Figure 3-2: MMCM Primitives

MMCME3_BASE Primitive

The MMCME3_BASE primitive provides access to the most frequently used features of a stand-alone MMCM. Clock deskew, frequency synthesis, coarse phase shifting, and duty cycle programming are available to use with the MMCME3_BASE. The ports are listed in Table 3-1.

Table 3-1: MMCME3_BASE Ports

Description	Ports
Clock Input	CLKIN1, CLKFBIN
Control Inputs	RST
Clock Output	CLKOUT0 to CLKOUT6, CLKOUT0B to CLKOUT3B, CLKFBOUT, and CLKFBOUTB
Status and Data Outputs	LOCKED
Power Control	PWRDWN

MMCME3_ADV Primitive

The MMCME3_ADV primitive provides access to all MMCME3_BASE features plus additional ports for clock switching, access to the dynamic reconfiguration port (DRP), and dynamic fine-phase shifting. The MMCME3_ADV ports are listed in Table 3-2.

Table 3-2: MMCME3_ADV Ports

Description	Ports
Clock Input	CLKIN1, CLKIN2, CLKFBIN, DCLK, PSCLK
Control and Data Input	RST, CLKINSEL, DWE, DEN, DADDR, DI, PSINCDEC, PSEN, CDDCREQ
Clock Output	CLKOUT0 to CLKOUT6, CLKOUT0B to CLKOUT3B, CLKFBOUT, and CLKFBOUTB
Status and Data Output	LOCKED, DO, DRDY, PSDONE, CLKINSTOPPED, CLKFBSTOPPED, CDDCDONE
Power Control	PWRDWN

The MMCM 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 in this section. The VCO operating frequency can be determined by using the following relationship:

$$F_{VCO} = F_{CLKIN} \times \frac{M}{D} \quad \text{Equation 3-1}$$

$$F_{OUT} = F_{CLKIN} \times \frac{M}{D \times O} \quad \text{Equation 3-2}$$

where the M, D, and O counters are shown in [Figure 3-2, page 32](#). The value of M corresponds to the CLKFBOUT_MULT_F setting, the value of D to the DIVCLK_DIVIDE, and O to the CLKOUT_DIVIDE.

The seven “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 because 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, an MMCM is used to compensate for the clock network delay. This feature is supported in UltraScale architecture-based devices. A clock output matching the reference clock CLKIN frequency (always CLKFBOUT) is connected to a clock buffer of the same type driving the logic and fed back to the CLKFBIN feedback pin of the MMCM. 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.

Frequency Synthesis Only Using Integer Divide

The MMCMs can also be used for stand-alone frequency synthesis. In this application, the MMCM is not used to deskew a clock network. Rather, it generates an output clock frequency for other blocks. In this mode, the MMCM feedback paths are internal, which

keeps all the routing local, minimizing the jitter. Figure 3-3 shows the MMCM configured as a frequency synthesizer. In this example, an external 33 MHz reference clock is available. The reference clock can be a crystal oscillator or the output of another MMCM. Setting the M counter to 32 makes the VCO oscillate at 1056 MHz (33 MHz x 32). The MMCM outputs are programmed to provide (for example) a 528 MHz processor clock, a 264 MHz gasket clock, a 176 MHz clock, a 132 MHz memory interface clock, a 66 MHz interface, and a 33 MHz interface. In this example, there are no required phase relationships between the reference clock and the output clocks, but there are required relationships between the output clocks.

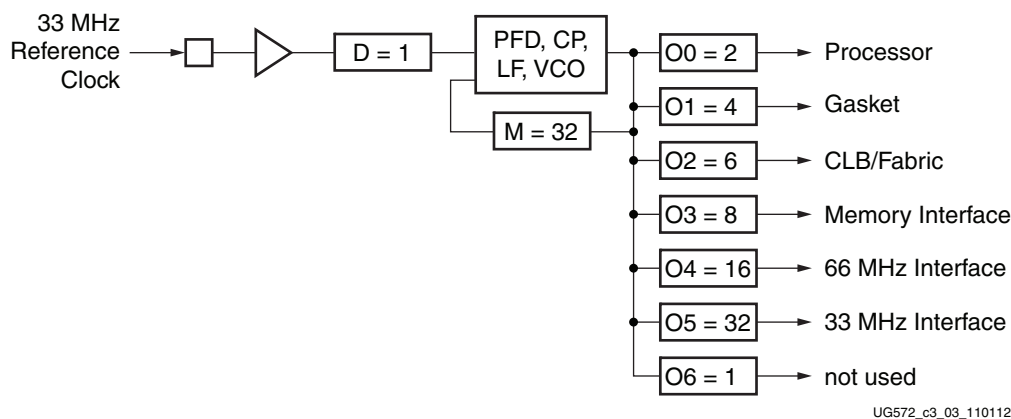


Figure 3-3: MMCM as a Frequency Synthesizer

Frequency Synthesis Using Fractional Divide in the MMCM

Devices support fractional (non-integer) divides in the CLKOUT0 output path. The resolution of the fractional divide is 1/8 or 0.125 degrees, effectively increasing the number of synthesizable frequencies by a factor of eight. For example, if the CLKIN frequency is 100 MHz and the M divide value is set to 8, the VCO frequency is 800 MHz. CLKOUT0 can be used to further fractionally divide the 800 MHz VCO frequency (e.g., CLKOUT0_DIVIDE = 2.5, resulting in a 320 MHz output frequency).

When using the fractional divider, the duty cycle is not programmable for outputs used in the fractional mode.

Jitter Filter

MMCMs reduce the jitter inherent on a reference clock. The MMCM can be instantiated as a stand-alone function to only support filtering jitter from an external clock before it is driven into another block. As a jitter filter, it is usually assumed that the MMCM act as a buffer and regenerate the input frequency on the output (e.g., $F_{IN} = 100$ MHz, $F_{OUT} = 100$ MHz). In general, greater jitter filtering is possible by using the MMCM attribute BANDWIDTH set to Low. Setting the BANDWIDTH to Low can incur an increase in the static offset of the MMCM.

Limitations

The MMCM has some restrictions that must be adhered to. These are summarized in the MMCM electrical specifications in the UltraScale device data sheets [Ref 4]. In general, the major limitations are VCO operation range, input frequency, duty cycle programmability, and phase shift. In addition, there are connectivity limitations to other clocking elements (pins, GTs, and clock buffers). Cascading MMCMs can only occur through the clock routing network.

VCO Operating Range

The minimum and maximum VCO operating frequencies are defined in the electrical specification of the UltraScale device data sheets [Ref 4]. These values can also be extracted from the speed specification.

Minimum and Maximum Input Frequencies

The minimum and maximum CLKIN input frequencies are defined in the electrical specification of the UltraScale device data sheets [Ref 4].

Duty Cycle Programmability

Only discrete duty cycles are possible given a VCO operating frequency. Depending on the CLKOUT_DIVIDE value, a minimum and maximum range is possible with a step size that is also dependent on the CLKOUT_DIVIDE value. The Clocking Wizard tool gives the possible values for a given CLKOUT_DIVIDE.

Phase Shift

In many cases, there needs to be a phase shift between clocks. The MMCM has multiple options to implement phase shifting. Static phase shifting can be achieved by selecting one of the eight VCO output phases with additional fine phase shifting available in the CLKOUT output counters depending on the CLKOUT divide value. There is also an interpolated phase shifting capability in either fixed or dynamic mode. The MMCM phase shifting capabilities are very powerful, which can lead to complex scenarios. By using the Clocking Wizard, the allowable phase shift values are determined based on the MMCM configuration settings.

Static Phase Shift Mode

The static phase shift (SPS) resolution in time units is defined as:

$$SPS = \frac{1}{8F_{VCO}} \text{period} \text{ or } \frac{D}{8MF_{IN}} \text{period} \quad \text{Equation 3-3}$$

Because the VCO can provide eight phase-shifted clocks at 45° each; always providing possible settings for 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° of phase shift. The higher

the VCO frequency is, the smaller the phase shift resolution. Because the VCO has a distinct operating range, it is possible to bound the phase shift resolution using from

$$\frac{1}{8F_{VCOMIN}} \text{ to } \frac{1}{8F_{VCOMAX}} \text{ period.}$$

Each CLKOUT output counter is individually programmable allowing each to have an additional phase shift resolution in degrees based on the phase of the VCO selected and the CLKOUT counter divide value. The granularity of the CLKOUT phase shift value can be calculated as $45^\circ/\text{CLKOUT_DIVIDE}$ value. The maximum phase shift range is also determined by the CLKOUT_DIVIDE value. The maximum phase shift is 360° when $\text{CLKOUT_DIVIDE} \leq 64$. When CLKOUT_DIVIDE is > 64 , the maximum phase shift is:

$$\text{Maximum Phase Shift} = \left(\frac{63}{\text{CLKOUT_DIVIDE}} \times 360 \right) + (7 \times \text{Phase Shift Value}) \quad \text{Equation 3-4}$$

It is possible to phase shift the CLKFBOUT feedback clock. In that case, all CLKOUT output clocks are negatively phase shifted with respect to CLKIN.

The two fractional counters (CLKFBOUT and CLKOUT0) also have static phase shift capability. A phase shift step is defined as:

$$\text{SPS}(\text{frac}) = \frac{360}{8 \times \text{fractional_divide_value}} \text{ or } \frac{45}{\text{fractional_divide_value}} \quad \text{Equation 3-5}$$

For example, if the fractional divide value is 2.125, a static phase shift step is $360/(2.125 \times 8) = 21.176$ degrees.

Interpolated Fine Phase Shift in Fixed or Dynamic Mode in the MMCM

Interpolated fine phase shift (IFPS) mode in the MMCM has linear shift behavior independent of the CLKOUT_DIVIDE value, and the phase shift resolution only depends on the VCO frequency. In this mode, the output clocks can be rotated 360° round robin

in linear increments of $\frac{1}{56F_{VCO}}$.

If the VCO runs at 600 MHz, the phase resolution is approximately (rounded) 30 ps, and at 1.6 GHz is approximately (rounded) 11 ps.

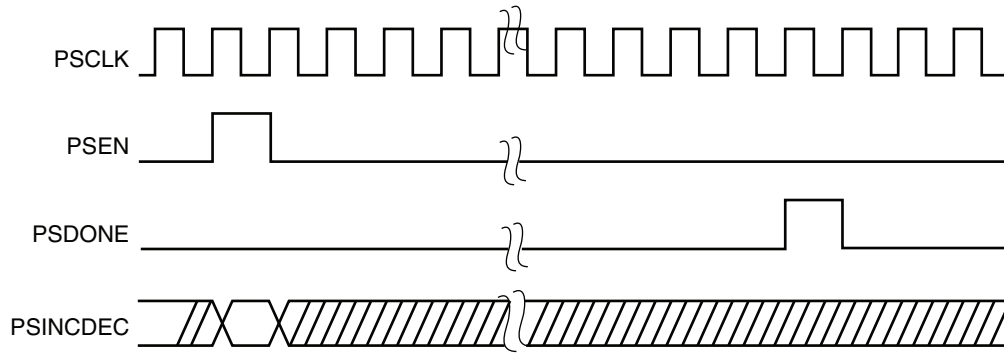
No initial phase shift value can be programmed during configuration. When using fine phase shift, no initial phase shift amount can be set. The phase always starts at zero and can then be dynamically incremented or decremented. The dynamic phase shift is controlled by the PS interface of the MMCME3_ADV. This phase shift mode equally affects all CLKOUT output clocks that are selected for this mode by setting the USE_FINE_PS attribute to TRUE. In interpolated fine phase shift mode, a clock must always be connected to the PSCLK pin of the MMCM. Regardless of the interpolated fine phase shift mode (fixed or dynamic) a clock is in, the clock must always be connected to the PSCLK pin of the MMCM. Each individual CLKOUT counter can independently either select the interpolated phase shift, the previously described static phase shift mode, or none. Fractional divide is not allowed in

either fixed or dynamic interpolated fine phase shift mode. Fixed or dynamic phase shifting of the feedback path results in a negative phase shift of all output clocks with respect to CLKIN. The dynamic phase shift interface cannot be used when the phase shift mode is set to fixed.

Dynamic Phase Shift Interface in the MMCM

The MMCME3_ADV primitive provides three inputs and one output for dynamic fine phase shifting. Each CLKOUT and the CLKFBOUT divider can be individually selected for phase shifting. The attributes CLKOUT[0:6]_USE_FINE_PS and CLKFBOUT_USE_FINE_PS select the output clocks to be dynamically phase shifted. The dynamic phase shift amount is common to all the output clocks selected.

The variable phase shift is controlled by the PSEN, PSINCDEC, PSCLK, and PSDONE ports (Figure 3-4). After the MMCM locks, the initial phase is determined by the CLKOUT_PHASE attribute. Most commonly, no initial phase shift is selected. The phase of the MMCM output clock(s) increments/decrements according to the interaction of PSEN, PSINCDEC, PSCLK, and PSDONE from the initial or previously performed dynamic phase shift. PSEN, PSINCDEC, and PSDONE are synchronous to PSCLK. When PSEN is asserted for one PSCLK clock period, a phase shift increment/decrement is initiated. When PSINCDEC is High, an increment is initiated and when PSINCDEC is Low, a decrement is initiated. Each increment adds to the phase shift of the MMCM clock outputs by 1/56th of the VCO period. Similarly, each decrement decreases the phase shift by 1/56th of the VCO period. PSEN must be active for one PSCLK period. PSDONE is High for exactly one clock period when the phase shift is complete. The number of PSCLK cycles is deterministic. After initiating the phase shift by asserting PSEN and the completion of the phase shift signaled by PSDONE, the MMCM output clocks gradually drift from their original phase shift to an increment/decrement phase shift in a linear fashion. The completion of the increment or decrement is signaled when PSDONE asserts High. After PSDONE has pulsed High, another increment/decrement can be initiated. There is no maximum phase shift or phase shift overflow. An entire clock period (360 degrees) can always be phase shifted regardless of frequency. When the end of the period is reached, the phase shift wraps around round-robin style.



UG572_c3_04_110112

Figure 3-4: Phase Shift Timing Diagram

Clock Divide Dynamic Change

The Clock Divide Dynamic Change (CDDC) feature supports the dynamic change of the clock output dividers (CLKOUT[6:0]_DIVIDE) in conjunction with the DRP interface without the need for resetting the MMCM. Effectively, one or more of the MMCM output clock frequencies can be changed while leaving other output clocks untouched and running continuously. Two pins (CDDCREQ and CDDCDONE) control the handshaking. The application requests a change of output counter values (the CLKOUT_DIVIDE value) by asserting the CDDCREQ signal. New values are written through the standard DRP interface one port at a time and governed by standard DRP protocol (DEN, DWE, and DRDY). The DRP address of the CLKOUT[6:0] counter written to determines which output clocks are affected. After a CLKOUTx counter is written to, the associated clock output stops toggling. This can be followed by more changes (DRP writes) to other CLKOUT counters in an identical fashion. When all DRP writes have been completed (after the last DRDY), the CDDCREQ input must be deasserted, after which the affected output counter(s) are synchronously restarted. The MMCM acknowledges that the changes have taken place and the new output clocks (frequencies) are available for use by asserting CDDCDONE.

CLKOUT ports not affected by the CDDC change continue to function uninterrupted during this operation and maintain their phase relationship to each other (Figure 3-5). However, the output clocks (ports) that were changed via the CDDC procedure are not phase aligned (synchronized) to the other output clocks not affected by CDDCREQ. Clocks affected by CDDCREQ should not be used after the signal has been asserted because the output might glitch and the clocks stop toggling. This feature is not available in fractional mode.

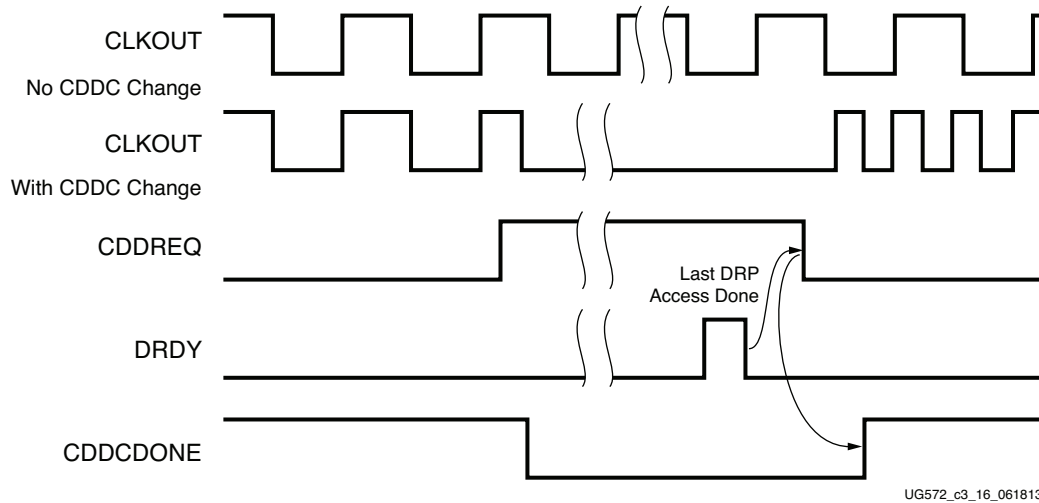


Figure 3-5: CDDC Timing Diagram

MMCM Counter Cascading

The CLKOUT6 divider (counter) can be cascaded with the CLKOUT4 divider. This provides the capability of an output divider that is larger than 128. CLKOUT6 feeds the input of the CLKOUT4 divider. There is a static phase offset between the output of the cascaded divider and all other output dividers.

MMCM Programming

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

- Reference clock period
- Output clock frequencies (up to seven maximum)
- Output clock duty cycle (default is 50%)
- Output clock phase shift in number of degrees relative to the original 0 phase of the clock
- Desired bandwidth of the MMCM (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. There can be a very large number of frequencies. When using integer divides, in the worst case there are $106 \times 64 \times 136 = 868,363$ possible combinations. In reality, the total number of different frequencies is less because 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 10 MHz, D can only go from 1 to 10:

- D = 1, M can only have values from four to 16
- D = 2, M can only have values from eight to 32
- D = 4, M can only have values from 16 to 64

In addition, D = 1 M = 4 is a subset of D = 2 M = 8, D = 4 M = 16, and D = 8 M = 32 allowing these cases to be dropped. For this case, only D = 1, 3, 5, 6, 7, and 9 are considered because all other D values are subsets of these cases. 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. After 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. This process is continued until all the output frequencies are selected.

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

$$D_{MIN} = \text{roundup} \frac{f_{IN}}{f_{PFD MAX}} \quad \text{Equation 3-6}$$

$$D_{MAX} = \text{rounddown} \frac{f_{IN}}{f_{PFD MIN}} \quad \text{Equation 3-7}$$

$$M_{MIN} = \text{roundup} \left(\frac{f_{VCOMIN}}{f_{IN}} \times D_{MIN} \right) \quad \text{Equation 3-8}$$

$$M_{MAX} = \text{rounddown} \left(\frac{f_{VCOMAX}}{f_{IN}} \times D_{MAX} \right) \quad \text{Equation 3-9}$$

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.

$$M_{IDEAL} = \frac{D_{MIN} \times f_{VCOMAX}}{f_{IN}} \quad \text{Equation 3-10}$$

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.

MMCM Ports

Table 3-3 summarizes the MMCM ports.

Table 3-3: MMCM Ports⁽¹⁾

Pin Name	I/O	Pin Description
CLKIN1	Input	General clock input. See CLKIN1 – Primary Reference Clock Input .
CLKIN2	Input	Secondary clock input for the MMCM reference clock. See CLKIN2 – Secondary Clock Input .
CLKFBIN	Input	Feedback clock input. See CLKFBIN – Feedback Clock Input .
CLKINSEL	Input	This signal controls the state of the clock input MUX, High = CLKIN1, Low = CLKIN2. CLKINSEL dynamically switches the MMCM reference clock. See CLKINSEL – Clock Input Select .
RST	Input	Asynchronous reset signal. The RST signal is an asynchronous reset for the MMCM. The MMCM synchronously re-enables itself when this signal is released (i.e., MMCM re-enabled). A reset is required when the input clock conditions change (e.g., frequency). See RST – Asynchronous Reset Signal .
PWRDWN	Input	Powers down instantiated but unused MMCMs. See PWRDWN – Power Down .
DADDR[6:0]	Input	The dynamic reconfiguration address (DADDR) input bus provides a reconfiguration address for the dynamic reconfiguration. When not used, all bits must be assigned zeros. See DADDR[6:0] – Dynamic Reconfiguration Address .
DI[15:0]	Input	The dynamic reconfiguration data input (DI) bus provides reconfiguration data. When not used, all bits must be set to zero. See DI[15:0] – Dynamic Reconfiguration Data Input .
DWE	Input	The dynamic reconfiguration write enable (DWE) input pin provides the write enable control signal to write the DI data into the DADDR address. When not used, it must be tied Low. See DWE – Dynamic Reconfiguration Write Enable .

Table 3-3: MMCM Ports⁽¹⁾ (Cont'd)

Pin Name	I/O	Pin Description
DEN	Input	The dynamic reconfiguration enable (DEN) provides the enable control signal to access the dynamic reconfiguration feature. When the dynamic reconfiguration feature is not used, DEN must be tied Low. See DEN – Dynamic Reconfiguration Enable Strobe .
DCLK	Input	The DCLK signal is the reference clock for the dynamic reconfiguration port. See DCLK – Dynamic Reconfiguration Reference Clock .
PSCLK	Input	Phase shift clock. See PSCLK – Phase Shift Clock .
PSEN	Input	Phase shift enable. See PSEN – Phase Shift Enable .
PSINCDEC	Input	Phase shift increment/decrement control. See PSINCDEC – Phase Shift Increment/Decrement Control .
CLKOUT[0:6]	Output	User configurable clock outputs (0 through 6) that can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 128. The output clocks are phase aligned to each other (unless phase shifted) and aligned to the input clock with a proper feedback configuration.
CLKOUT[0:3]B	Output	Inverted CLKOUT[0:3]. See CLKOUT[0:3]B – Inverted Output Clocks .
CLKFBOUT	Output	Dedicated MMCM feedback output. See CLKFBOUT – Dedicated MMCM Feedback Output .
CLKFBOUTB	Output	Inverted CLKFBOUT. See CLKFBOUTB – Inverted CLKFBOUT .
CLKINSTOPPED	Output	Status pin indicating that the input clock has stopped. See CLKINSTOPPED – Input Clock Status .
CLKFBSTOPPED	Output	Status pin indicating that the feedback clock has stopped. See CLKFBSTOPPED – Feedback Clock Status .
LOCKED	Output	An output from the MMCM that indicates when the MMCM has achieved phase alignment within a predefined window and frequency matching within a predefined PPM range. The MMCM automatically locks after power on. No extra reset is required. LOCKED is deasserted if the input clock stops or the phase alignment is violated (e.g., input clock phase shift). The MMCM must be reset after LOCKED is deasserted.
DO[15:0]	Output	The dynamic reconfiguration output bus provides MMCM data output when using dynamic reconfiguration. See DO[15:0] – Dynamic Reconfiguration Output Bus .
DRDY	Output	The dynamic reconfiguration ready output (DRDY) provides the response to the DEN signal for the MMCM's dynamic reconfiguration feature. See DRDY – Dynamic Reconfiguration Ready .
PSDONE	Output	Phase shift done. See PSDONE – Phase Shift Done .
CDDCREQ	Input	Requests a dynamic frequency change for selected clock outputs. See Clock Divide Dynamic Change, page 38 .
CDDCDONE	Output	Signals that the dynamic frequency change is completed. See Clock Divide Dynamic Change, page 38 .

Notes:

1. All control and status signals except PSINCDEC are active High.

MMCM Port Descriptions

CLKIN1 – Primary Reference Clock Input

CLKIN1 can be driven by a global clock I/O directly when in the same bank adjacent to the PHY tile.

CLKIN2 – Secondary Clock Input

CLKIN1 can be driven by a global clock I/O directly when in the same bank adjacent to the PHY tile.

CLKFBIN – Feedback Clock Input

CLKFBIN must be connected either directly to the CLKFBOUT for internal feedback or IBUFG (through a global clock pin for external deskew), B, or interconnect (not recommended). For external clock alignment, the feedback path clock buffer type should match the forward clock buffer type.

CLKFBOUT – Dedicated MMCM Feedback Output

For possible configuration of CLKFBOUT, see [MMCM Use Models, page 52](#). CLKFBOUT can also drive logic if the feedback path contains a clock buffer.

CLKFBOUTB – Inverted CLKFBOUT

This signal should not be used for feedback. It provides an additional, inverted CLKFBOUT output clock. CLKFBOUTB can drive logic if the feedback path contains a clock buffer.

CLKINSEL – Clock Input Select

The CLKINSEL signal controls the state of the clock input multiplexers. High = CLKIN1, Low = CLKIN2 (see [Reference Clock Switching, page 52](#)). The MMCM must be held in RESET during clock switchover.

RST – Asynchronous Reset Signal

The RST signal is an asynchronous reset for the MMCM. The MMCM is synchronously re-enabled when this signal is deasserted.

PWRDWN – Power Down

This signal powers down instantiated but currently unused MMCMs. This mode can be used to save power for temporarily inactive portions of the design and/or MMCMs that are not active in certain system configurations. No MMCM power is consumed in this mode.

DADDR[6:0] – Dynamic Reconfiguration Address

The dynamic reconfiguration address (DADDR) input bus provides a reconfiguration address for dynamic reconfiguration. The address value on this bus specifies the 16 configuration bits that are written or read with the next DCLK cycle. When not used, all bits must be assigned zeros.

DI[15:0] – Dynamic Reconfiguration Data Input

The dynamic reconfiguration data input (DI) bus provides reconfiguration data. The value of this bus is written to the configuration cells. The data is presented in the cycle that DEN and DWE are active. The data is captured in a shadow register and written at a later time. DRDY indicates when the DRP port is ready to accept another write. When not used, all bits must be set to zero.

DWE – Dynamic Reconfiguration Write Enable

The dynamic reconfiguration write enable (DWE) input pin provides the write/read enable control signal to write the DI data into or read the DO data from the DADDR address. When not used, it must be tied Low.

DEN – Dynamic Reconfiguration Enable Strobe

The dynamic reconfiguration enable strobe (DEN) provides the enable control signal to access the dynamic reconfiguration feature and enables all DRP port operations. When the dynamic reconfiguration feature is not used, DEN must be tied Low.

DCLK – Dynamic Reconfiguration Reference Clock

The DCLK signal is the reference clock for the dynamic reconfiguration port. The rising edge of this signal is the timing reference for all other port signals. The setup time is specified in the UltraScale device data sheets [Ref 4]. There is no hold time requirement for the other input signals relative to the rising edge of the DCLK. The pin can be driven by an IBUF, IBUG, BUFGCE, or BUFGCTRL. There are no dedicated connections to this clock input.

PSCLK – Phase Shift Clock

This input pin provides the source clock for the dynamic phase shift interface. All other inputs are synchronous to the positive edge of this clock. The pin can be driven by an IBUF, IBUG, BUFG, or BUFGCE. There are no dedicated connections to this clock input.

PSEN – Phase Shift Enable

A dynamic (variable) phase shift operation is initiated by synchronously asserting this signal. PSEN must be activated for one cycle of PSCLK. After initiating a phase shift, the phase is gradually shifted until a High pulse on PSDONE indicates that the operation is complete. There are no glitches or sporadic changes during the operation. From the start to the end of the operation, the phase is shifted in a continuous analog manner.

PSINCDEC – Phase Shift Increment/Decrement Control

This input signal synchronously indicates if the dynamic phase shift is an increment or decrement operation (positive or negative phase shift). PSENCDEC is asserted High for increment and Low for decrement. There is no phase shift overflow associated with the dynamic phase shift operation. If 360° or more are shifted, the phase wraps around, starting at the original phase.

CLKOUT[0:6] – Output Clocks

These user-configurable clock outputs (CLKOUT0 through CLKOUT6) can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 128. The input clock and output clocks can be phase aligned.

For possible configurations, see [MMCM Use Models, page 52](#). In the MMCM, CLKOUT0 and CLKFBOUT can be used in fractional divide mode. All CLKOUT outputs can be used in non-fractional mode to provide a static or dynamic phase shift. In fractional mode, only fixed phase shift is allowed. See [Static Phase Shift Mode, page 35](#) for more information.

CLKOUT[0:3]B – Inverted Output Clocks

This is the inverted (180° phase shift) CLKOUT[0:3].

CLKINSTOPPED – Input Clock Status

This is a status pin indicating that the input clock has stopped. This signal is asserted within one clock cycle of clock stoppage. The signal is deasserted after the clock has restarted and LOCKED is achieved, or the clock is switched to the alternate clock input and the MMCM has re-locked.

CLKFBSTOPPED – Feedback Clock Status

This is a status pin indicating that the feedback clock has stopped. CLKFBSTOPPED is asserted within one clock cycle of clock stoppage. The signal is deasserted after the feedback clock has restarted and the MMCM has re-locked.

LOCKED

This is an output from the MMCM used to indicate when the MMCM has achieved phase and frequency alignment of the reference clock and the feedback clock at the input pins. Phase alignment is within a predefined window and frequency matching within a predefined PPM range. The MMCM automatically locks after power on; no extra reset is required. LOCKED is deasserted within one PFD clock cycle if the input clock stops, the phase alignment is violated (e.g., input clock phase shift), or the frequency has changed. The MMCM must be reset when LOCKED is deasserted. The clock outputs should not be used prior to the assertion of LOCKED.

DO[15:0] – Dynamic Reconfiguration Output Bus

The dynamic reconfiguration output bus provides MMCM data output when using dynamic reconfiguration. If DWE is inactive while DEN is active at the rising edge of DCLK, this bus holds the content of the configuration cells addressed by DADDR. The DO bus must be captured on the rising edge of DCLK when DRDY is active. The DO bus value is held until the next DRP operation.

DRDY – Dynamic Reconfiguration Ready

The dynamic reconfiguration ready output (DRDY) provides the response to the DEN signal for the MMCM’s dynamic reconfiguration feature. This signal indicates that a DEN/ DCLK operation has completed.

PSDONE – Phase Shift Done

The phase shift done output signal is synchronous to the PSCLK. When the current phase shift operation is completed, the PSDONE signal is asserted for one clock cycle indicating that a new phase shift cycle can be initiated.

CDDCREQ – Request a Clock Output Divide Change

This is a request signal for dynamically changing the output clock divide value and therefore the frequency. When asserted High, a request is sent to all affected counters and must stay asserted until the last change via the DRP has been completed.

CDDCDONE – Clock Output Change Done

This is an acknowledge signal from the MMCM that the output clock divide change is complete and the output is valid.

MMCM Attributes

Table 3-4 lists the attributes for the MMCME3_BASE and MMCME3_ADV primitives.

Table 3-4: MMCM Attributes

Attribute	Type	Allowed Values	Default	Description
BANDWIDTH	String	HIGH LOW OPTIMIZED	OPTIMIZED	Specifies the MMCM programming algorithm affecting the jitter, phase margin, and other characteristics of the MMCM.

Table 3-4: MMCM Attributes (Cont'd)

Attribute	Type	Allowed Values	Default	Description
CLKOUT[1:6]_DIVIDE	Integer	1 to 128	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_F and DIVCLK_DIVIDE values, determines the output frequency.
CLKOUT[0]_DIVIDE_F ⁽¹⁾	Integer or Real	1 to 128 or 2.000 to 128.000 in increments of 0.125	1	
CLKOUT[0:6]_PHASE	Real	-360.000 to 360.000 in increments of 1/56 the F _{VCO} and/or increments depending on CLKOUT_DIVIDE.	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).
CLKOUT[0:6]_DUTY_CYCLE	Real	0.01 to 0.99	0.50	Specifies the duty cycle of the associated CLKOUT clock output as a percentage (i.e., 0.50 generates a 50% duty cycle).
CLKFBOUT_MULT_F ⁽¹⁾	Integer or Real	2 to 64 or 2.000 to 64.000 in increments of 0.125	5	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, determines the output frequency.
DIVCLK_DIVIDE	Integer	1 to 106	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD.
CLKFBOUT_PHASE	Real	-360.000 to 360.000	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 MMCM.
REF_JITTER1 REF_JITTER2	Real	0.000 to 0.999	0.010	Allows specification of the expected jitter on the reference clock to better optimize MMCM performance. A bandwidth setting of OPTIMIZED attempts to choose the best parameter for input clocking when unknown. If known, the value provided should be specified in terms of the unit interval (UI) (the maximum peak-to-peak value) of the expected jitter on the input clock.

Table 3-4: MMCM Attributes (Cont'd)

Attribute	Type	Allowed Values	Default	Description
CLKIN1_PERIOD	Real	0.938 to 100.000	0.000	Specifies the input period in ns to the MMCM CLKIN1 input. Resolution is down to the ps. This information is mandatory and must be supplied.
CLKIN2_PERIOD	Real	0.938 to 100.000	0.000	Specifies the input period in ns to the MMCM CLKIN2 input. Resolution is down to the ps. This information is mandatory and must be supplied.
CLKFBOUT_USE_FINE_PS ⁽²⁾	Boolean	FALSE, TRUE	FALSE	CLKFBOUT counter variable fine phase shift enable.
CLKOUT0_USE_FINE_PS ⁽²⁾	Boolean	FALSE, TRUE	FALSE	CLKOUT0 counter variable fine phase shift enable. CLKOUT0_DIVIDE must be an integer. Therefore, fractional divide is not allowed.
CLKOUT[1:6]_USE_FINE_PS ⁽²⁾	Boolean	FALSE, TRUE	FALSE	CLKOUT[1:6] variable fine phase shift enable.
STARTUP_WAIT	Boolean	FALSE, TRUE	FALSE	Wait during the configuration startup cycle for the MMCM to lock.
CLKOUT4_CASCADE	Boolean	FALSE, TRUE	FALSE	Cascades the output divider (counter) CLKOUT6 into the input of the CLKOUT4 divider for an output clock divider that is greater than 128, effectively providing a total divide value of 16,384.
COMPENSATION	String	AUTO ⁽³⁾ , ZHOLD, EXTERNAL, INTERNAL, BUF_IN	ZHOLD	<p>Clock input compensation. Must be set to AUTO. Defines how the MMCM feedback is configured.</p> <p>ZHOLD: Indicates the MMCM is configured to provide a negative hold time at the I/O registers.</p> <p>EXTERNAL: Indicates a network external to the FPGA is being compensated.</p> <p>INTERNAL: Indicates the MMCM is using its own internal feedback path so no delay is being compensated.</p> <p>BUF_IN: Indicates that the configuration does not match with the other compensation modes and no delay is compensated.</p>

Table 3-4: MMCM Attributes (Cont'd)

Attribute	Type	Allowed Values	Default	Description
SS_EN	Boolean	FALSE, TRUE	FALSE	Enables spread spectrum generation.
SS_MODE	String	DOWN_LOW, DOWN_HIGH, CENTER_LOW, CENTER_HIGH	CENTER_HIGH	Controls the spread spectrum frequency deviation and the spread type.
SS_MOD_PERIOD	Integer	4000–40000	10000	Specifies the spread spectrum modulation period (ns).

Notes:

1. The Vivado tools round up or down to the nearest multiple of 0.125 if the value is not specified as an exact 1/8th fraction.
2. When using the variable fine phase shift, the initial phase shift value is always zero and cannot be preset to a static, initial phase.
3. The COMPENSATION attribute values are documented for informational purpose only. The Vivado® tools automatically select the appropriate compensation based on circuit topology. Do not manually select a compensation value, leave the attribute at the default value.

MMCM Clock Input Signals

The possible clock sources for the MMCM include:

- IBUF – Global clock input buffer. The MMCM compensates the delay of this path. IBUF represents a global clock pin in the same region. The IBUF must be located at a global clock pin location.
- BUFGCTRL – Internal global clock buffer. The MMCM does not compensate the delay of this path.
- BUFGCE – Global clock buffer. The MMCM does not compensate the delay of this path.

Counter Control

The MMCM output counters provide a wide variety of synthesized clocks using a combination of DIVIDE, DUTY_CYCLE, and PHASE. Figure 3-6 illustrates how the counter settings impact the counter output.

The top waveform represents the output from the VCO.

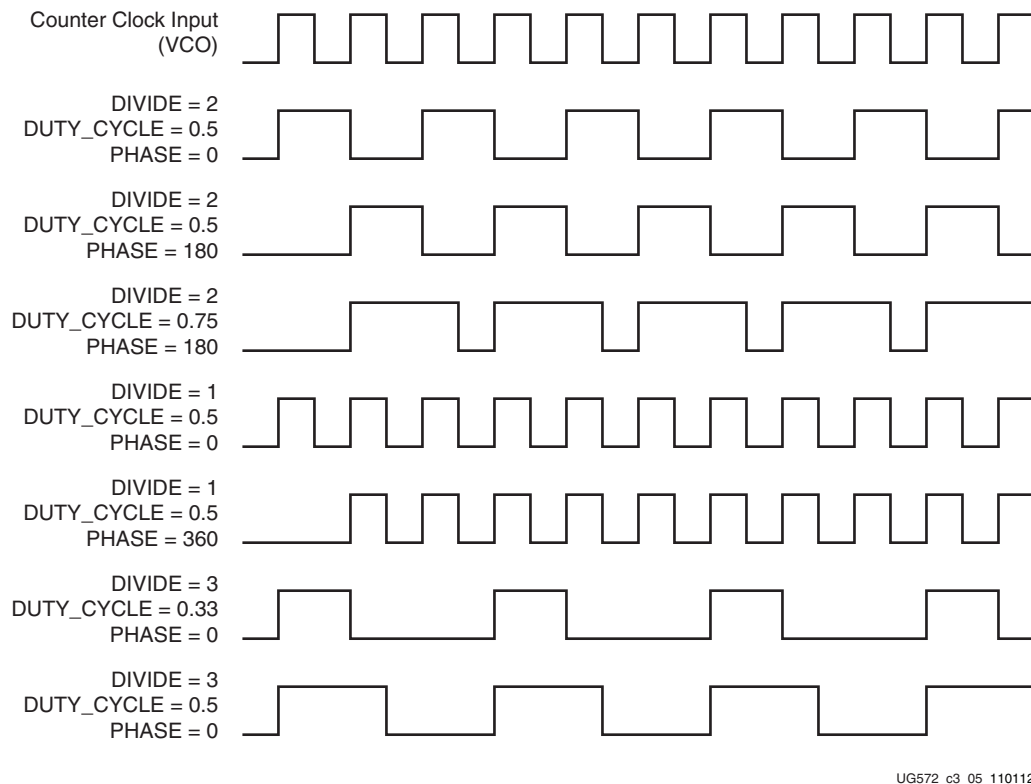


Figure 3-6: Output Counter Clock Synthesis Examples

Detailed VCO and Output Counter Waveforms

Figure 3-7 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 ensure the correct phase is maintained. This means the rising edge of the 0° phase happens 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. This counter setting can 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. Phase shifts greater than one VCO period are possible.

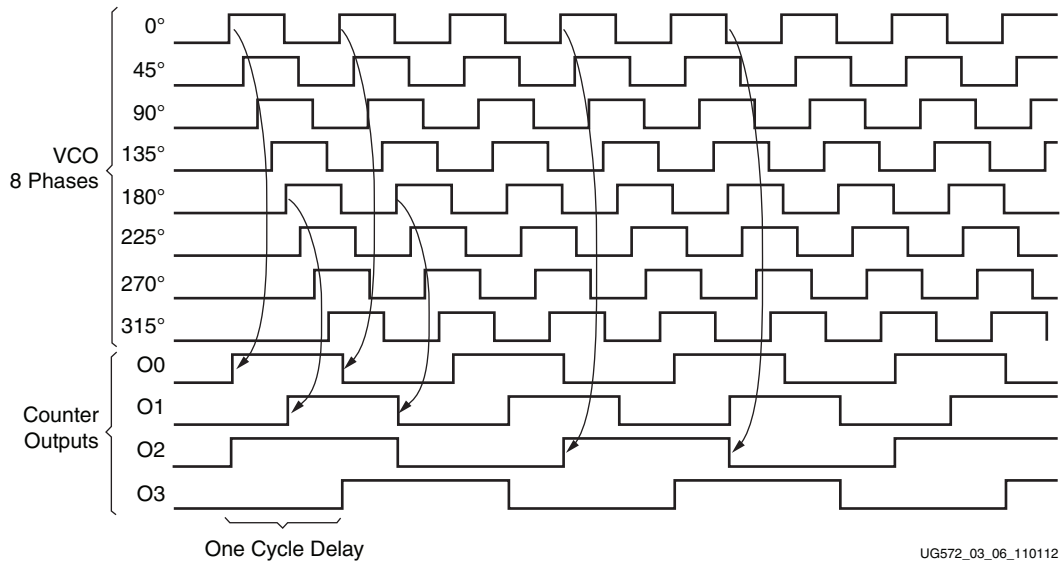


Figure 3-7: Selecting VCO Phases

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

All "O" counters can be equivalent; anything O0 can do, O1 can do. The O0 counter has the additional capability to be used in fractional divide mode. The MMCM outputs are flexible when connecting to the global clock network because they are identical. In most cases, this level of detail is imperceptible to the designer because the software and Clocking Wizard determine the proper settings through the MMCM attributes and Wizard inputs.

Reference Clock Switching

The MMCM reference clock can be dynamically switched by using the CLKINSEL pin. The switching is done asynchronously. After the clock switches, the MMCM is likely to lose LOCKED and automatically lock onto the new clock. Therefore, after the clock switches, the MMCM must be reset. The MMCM clock MUX switching is shown in Figure 3-8. The CLKINSEL signal directly controls the MUX. No synchronization logic is present.

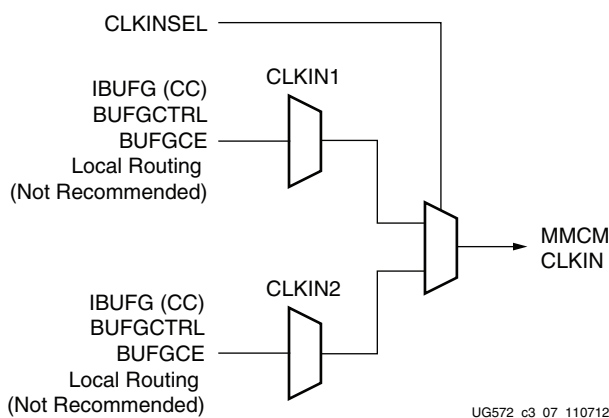


Figure 3-8: Input Clock Switching

Missing Input Clock or Feedback Clock

When the input clock or feedback clock is lost, the CLKINSTOPPED or CLKFBSTOPPED status signal is asserted. The MMCM deasserts the LOCKED signal. After the clock returns, the CLKINSTOPPED signal is deasserted and a RESET must be applied.

MMCM Use Models

The examples in this section show the MMCM. There are several methods to design with the MMCM. The Clocking Wizard in the Vivado tools can assist with generating the various MMCM parameters. Additionally, the MMCM can be manually instantiated as a component. It is also possible for the MMCM to be merged with an IP core. The IP core would contain and manage the MMCM.

Clock Network Deskew

One of the predominant uses of the MMCM is for clock network deskew. Figure 3-9 shows the MMCM in this mode. The clock output from one of the CLKOUT 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. The configuration in Figure 3-9 is the most flexible, but it does require two global clock networks.

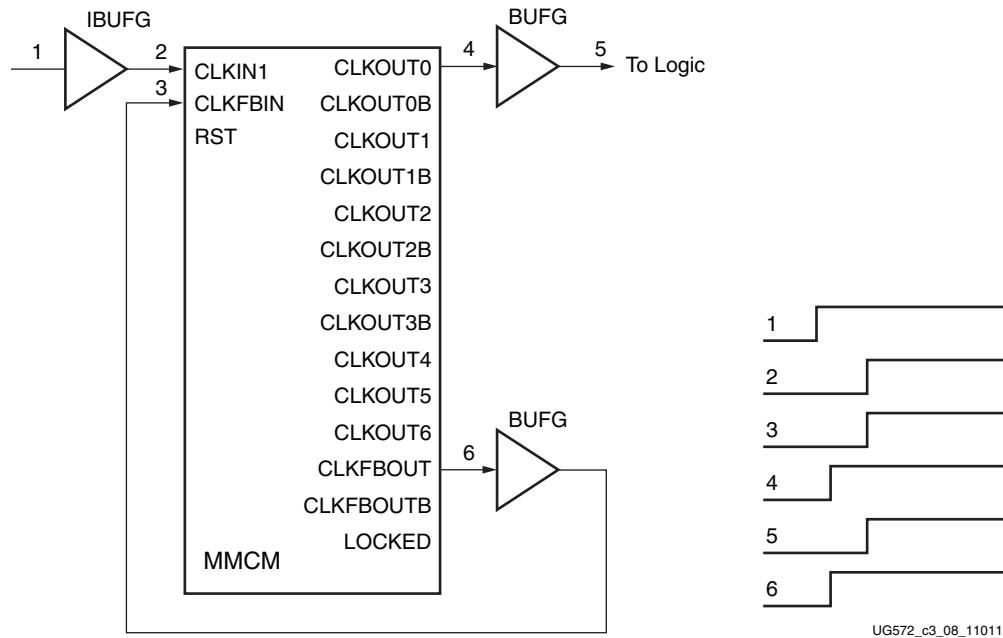


Figure 3-9: Global Clock Network Deskew Using Two BUFs

There are certain restrictions on implementing the feedback. The CLKFBOUT output can be used to provide the feedback clock signal. When an MMCM is driving both BUFs and BUFCTRL, only one of the clock buffers that is also used in the feedback path is deskewed. The fundamental restriction is that both input frequencies to the PFD must be identical. Therefore, this relationship must be met:

$$\frac{f_{IN}}{D} = f_{FB} = \frac{f_{VCO}}{M} \quad \text{Equation 3-11}$$

As an example, if f_{IN} is 166 MHz, $D = 1$, $M = 6$, and $O = 2$, then VCO is 996 MHz and the clock output frequency is 498 MHz. Because the M value in the feedback path is 6, both input frequencies at the PFD are 166 MHz.

Another more complex scenario has an input frequency of 66.66 MHz and $D = 2$, $M = 30$, and $O = 4$. The VCO frequency in this case is 1000 MHz and the CLKOUT output frequency is 250 MHz. Therefore, the feedback frequency at the PFD is 1000/30 or 33.33 MHz, matching the 66.66 MHz/2 input clock frequency at the PFD.

MMCM with Internal Feedback

The MMCM feedback can be internal to the MMCM when the MMCM is used as a synthesizer or jitter filter, and there is no required phase relationship between the MMCM input clock and the MMCM output clock. The MMCM performance increases because the feedback clock is not subjected to noise on the core supply since it never passes through a block powered by this supply. However, noise introduced on the CLKIN signal and the BUF are still present (Figure 3-10).

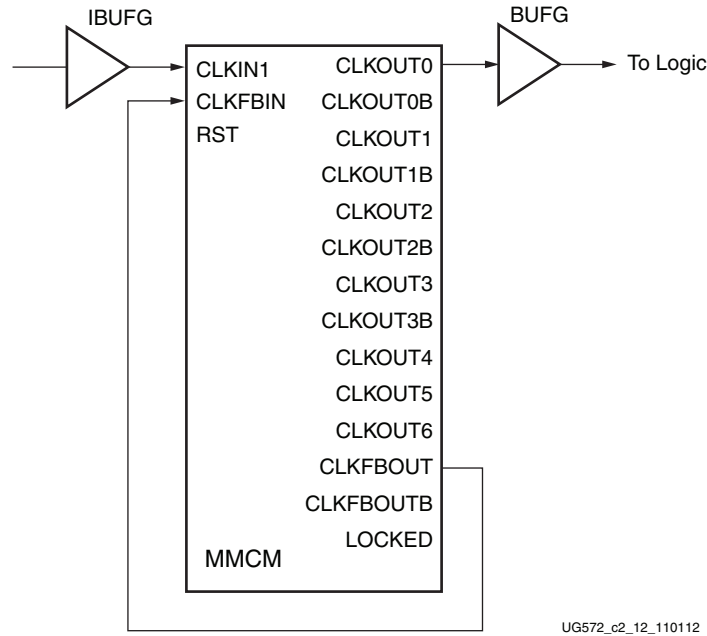
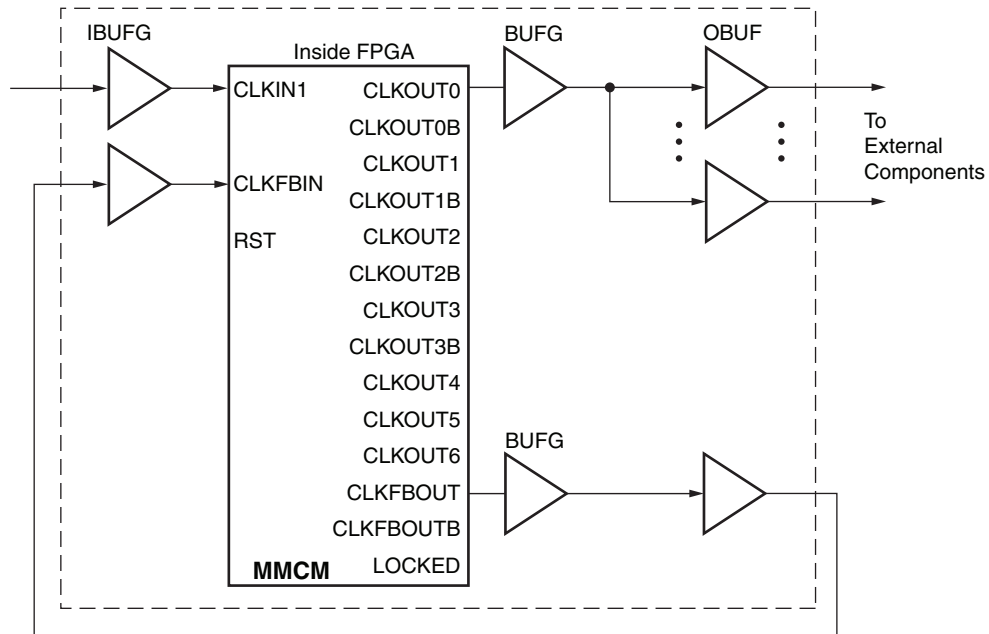


Figure 3-10: MMCM with Internal Feedback

Zero Delay Buffer

The MMCM 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 fanout to multiple destinations with a low skew between them. This configuration is shown in Figure 3-11. Here, the feedback signal drives off chip 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. The input clock buffers for CLKIN and CLKFBIN must be in the same bank.



UG572_c2_10_110112

Figure 3-11: Zero Delay Buffer

In some cases, precise alignment cannot occur because of the difference in loading between the input capacitance of the external component and the feedback path capacitance of the FPGA. For example, the external components can have an input capacitance of 1 pF to 4 pF while the FPGA has an input capacitance as specified in the UltraScale device data sheets [Ref 4]. There is a difference in the signal slope, which is basically skew. Designers should be aware of this effect to ensure timing.

CMT to CMT Connection

The MMCM can be cascaded through the routing resources only. There is no compensation for routing delays.

Spread-Spectrum Clock Generation

Spread-spectrum clock generation (SSCG) is widely used by manufacturers of electronic devices to reduce the spectral density of the electromagnetic interference (EMI) generated by these devices. Manufacturers must ensure that levels of electromagnetic energy emitted do not interfere with the operation of other nearby electronic devices. For example, the clarity of a phone call should not degrade when the phone is next to a video display. In the same way, the display should not be affected when the phone is used.

Electromagnetic compatibility (EMC) regulations are used to control the noise or EMI that causes these disturbances. Typical solutions for meeting EMC requirements involve adding expensive shielding, ferrite beads, or chokes. These solutions can adversely impact the cost

of the final product by complicating PCB routing and forcing longer product development cycles.

SSCG spreads the electromagnetic energy over a large frequency band to effectively reduce the electrical and magnetic field strengths measured within a narrow window of frequencies. The peak electromagnetic energy at any one frequency is reduced by modulating the SSCG output.

The MMCME3 can generate a spread-spectrum clock from a standard fixed frequency oscillator when SS_EN is set to TRUE (see Figure 3-12). Within the MMCME3, the VCO frequency is modulated along with CLKFBOUT and CLKOUT[6:4,1,0]. Clock outputs CLKOUT[3:2] are used to control the modulation period and are not available for general use. As long as the clock frequency is adjusted slowly, the spread spectrum does not affect the period jitter of the MMCME3.

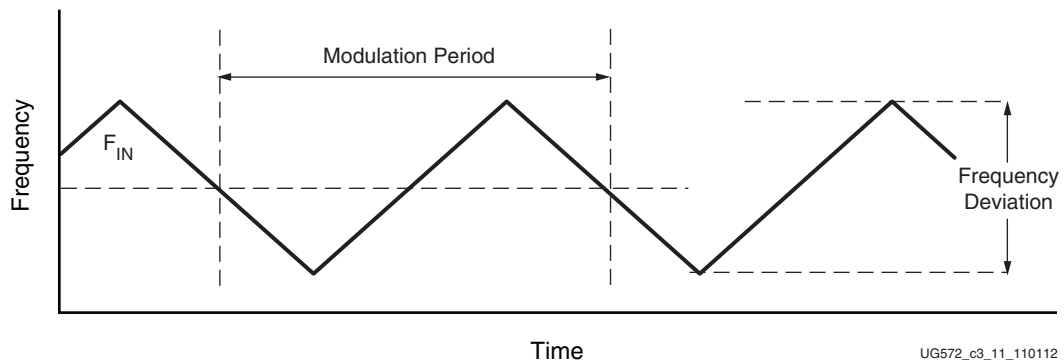


Figure 3-12: Center-Spread Modulation

Adjusting the modulation period SS_MOD_PERIOD allows the FPGA designer to direct the tools to select the closest modulation period based on the MMCME3 settings. The spread-spectrum modulation reduces EMI as long as the modulation frequencies are higher than the audible frequency range of 30 KHz. Typically, lower modulation frequencies are preferred by designers to minimize the impact of the introduction of spread spectrum.

Increasing the frequency deviation with SS_MODE (CENTER_HIGH or DOWN_HIGH) increases the overall EMI reduction, but care must be taken to ensure that the increased range of frequencies does not affect the overall system operation (see Figure 3-13). Because the spread-spectrum clock and the input clock are operating at different frequencies, any data being transferred between the clock domains should use an asynchronous FIFO to ensure that data is not lost. Increasing the frequency deviation requires a larger FIFO.

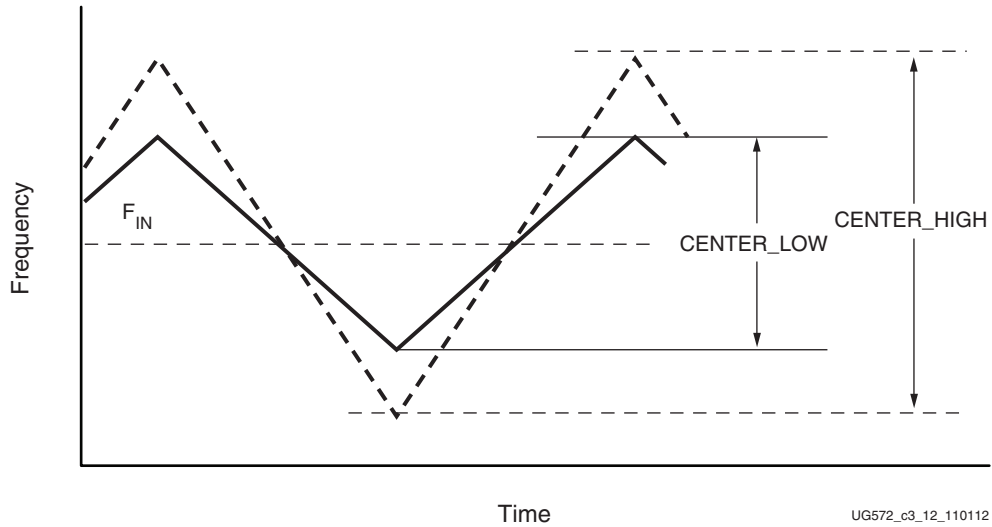


Figure 3-13: Center-Spread Modulation (CENTER_LOW vs. CENTER_HIGH)

Another design trade-off is the decision to use a center spread or down spread. Selecting SS_MODE ($DOWN_HIGH$, $DOWN_LOW$) spreads the frequencies to lower frequencies as shown in Figure 3-14. $DOWN_HIGH$ has similar frequency deviation to $CENTER_LOW$.

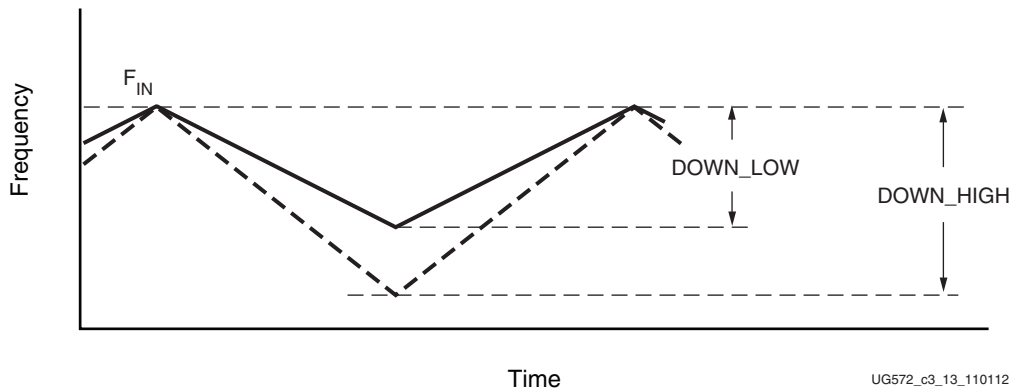


Figure 3-14: Down-Spread Modulation

The decision to use down spread is often the result of considering the timing analysis impact of spread spectrum. When using a spread-spectrum clock, the design must meet timing at the highest frequency in the frequency deviation. Therefore, if a 100 MHz clock with SS_MODE ($CENTER_LOW$) produces a 3% ($\pm 1.5\%$) center spread, the 100 MHz clock with 3% center spread must pass timing analysis as a 101.5 MHz clock. However, if SS_MODE ($DOWN_HIGH$) produces a 3% down spread, the input frequency is the highest frequency within the frequency deviation. Consequently, for a 100 MHz clock with 3% down spread, the down-spread clock would continue to be analyzed by timing analysis as a 100 MHz clock.

As an example of adjusting timing constraints to allow for the range of frequencies within the spread, the input frequency can be adjusted to manually account for the increased range of frequencies, as shown in Table 3-5.

Table 3-5: Manual Spread Spectrum Timing Adjustment Using Input Frequency

Parameter	Input Frequency (MHz)	M	Input Frequency Adjustment (FIN_SS)
SS_MODE(CENTER_HIGH)	$25 < F_{IN} < 35$	M = 28	$FIN_SS = F_{IN} \times 56/55$
	$35 < F_{IN} < 50$	M = 21	$FIN_SS = F_{IN} \times 42/41$
		M = 22	$FIN_SS = F_{IN} \times 44/43$
	$50 < F_{IN} < 75$	M = 28	$FIN_SS = F_{IN} \times 56/55$
	$75 < F_{IN} < 150$	M = 21	$FIN_SS = F_{IN} \times 42/41$
		M = 22	$FIN_SS = F_{IN} \times 44/43$
SS_MODE (CENTER_LOW)	$25 < F_{IN} < 35$	M = 56	$FIN_SS = F_{IN} \times 112/111$
	$35 < F_{IN} < 50$	M = 42	$FIN_SS = F_{IN} \times 84/83$
		M = 44	$FIN_SS = F_{IN} \times 88/87$
	$50 < F_{IN} < 75$	M = 56	$FIN_SS = F_{IN} \times 112/111$
	$75 < F_{IN} < 150$	M = 42	$FIN_SS = F_{IN} \times 84/83$
		M = 44	$FIN_SS = F_{IN} \times 88/87$
SS_MODE (DOWN_HIGH)	$25 < F_{IN} < 35$	M = 28	$FIN_SS = F_{IN}$
	$35 < F_{IN} < 50$	M = 21, 22	$FIN_SS = F_{IN}$
	$50 < F_{IN} < 75$	M = 28	$FIN_SS = F_{IN}$
	$75 < F_{IN} < 100$	M = 21, 22	$FIN_SS = F_{IN}$
	$100 < F_{IN} < 150$	M = 21, 22	$FIN_SS = F_{IN}$
SS_MODE (DOWN_LOW)	$25 < F_{IN} < 35$	M = 56	$FIN_SS = F_{IN}$
	$35 < F_{IN} < 50$	M = 42, 44	$FIN_SS = F_{IN}$
	$50 < F_{IN} < 75$	M = 56	$FIN_SS = F_{IN}$
	$75 < F_{IN} < 100$	M = 42, 44	$FIN_SS = F_{IN}$
	$100 < F_{IN} < 150$	M = 42, 44	$FIN_SS = F_{IN}$

For a 25 MHz input clock, the new timing constraints would be:

- SS_MODE(CENTER_HIGH) = $25 \times 56/55 = 25.45$ MHz
- SS_MODE (CENTER_LOW) = $25 \times 112/111 = 25.23$ MHz
- SS_MODE (DOWN_HIGH) = 25 MHz
- SS_MODE (DOWN_LOW) = 25 MHz

For an 80 MHz input clock, the new timing constraints would be:

- SS_MODE(CENTER_HIGH) = $80 \times 44/43 = 81.86$ MHz

- SS_MODE (CENTER_LOW) = $80 \times 88/87 = 80.92$ MHz
- SS_MODE (DOWN_HIGH) = 80 MHz
- SS_MODE (DOWN_LOW) = 80 MHz

Because the average output frequency when using down spread is lower than the input frequency, an asynchronous FIFO must be used for transferring data between the input and output clock domains. Logic within the MMCME3 controls the spread-spectrum modulation based on a given input frequency and SS_MOD_PERIOD. The restrictions shown in Table 3-6 apply when using spread-spectrum generation.

Table 3-6: Spread-Spectrum Generation Restrictions for MMCME3

Parameter	Value	
F _{MODULATION}	Minimum	25 [KHz]
	Maximum	250 [KHz]
Input Clock Frequency	Minimum	25 [MHz]
	Maximum	150 [MHz]
SS_MODE(CENTER_HIGH)	$25 \text{ MHz} < F_{IN} < 35 \text{ MHz}$	M = 28 D = 1
	$35 \text{ MHz} < F_{IN} < 50 \text{ MHz}$	M = 21, 22 D = 1
	$50 \text{ MHz} < F_{IN} < 75 \text{ MHz}$	M = 28 D = 2
	$75 \text{ MHz} < F_{IN} < 100 \text{ MHz}$	M = 21, 22 D = 2
	$100 \text{ MHz} < F_{IN} < 150 \text{ MHz}$	M = 21, 22 D = 3
SS_MODE (CENTER_LOW)	$25 \text{ MHz} < F_{IN} < 35 \text{ MHz}$	M = 56 D = 2
	$35 \text{ MHz} < F_{IN} < 50 \text{ MHz}$	M = 42, 44 D = 2
	$50 \text{ MHz} < F_{IN} < 75 \text{ MHz}$	M = 56 D = 4
	$75 \text{ MHz} < F_{IN} < 100 \text{ MHz}$	M = 42, 44 D = 4
	$100 \text{ MHz} < F_{IN} < 150 \text{ MHz}$	M = 42, 44 D = 6

Table 3-6: Spread-Spectrum Generation Restrictions for MMCME3 (Cont'd)

Parameter	Value	
SS_MODE (DOWN_HIGH)	$25 \text{ MHz} < F_{\text{IN}} < 35 \text{ MHz}$	M = 28 D = 1
	$35 < F_{\text{IN}} < 50 \text{ MHz}$	M = 21, 22 D = 1
	$50 \text{ MHz} < F_{\text{IN}} < 75 \text{ MHz}$	M = 28 D = 2
	$75 \text{ MHz} < F_{\text{IN}} < 100 \text{ MHz}$	M = 21, 22 D = 2
	$100 \text{ MHz} < F_{\text{IN}} < 150 \text{ MHz}$	M = 21, 22 D = 3
SS_MODE (DOWN_LOW)	$25 \text{ MHz} < F_{\text{IN}} < 35 \text{ MHz}$	M = 56 D = 2
	$35 \text{ MHz} < F_{\text{IN}} < 50 \text{ MHz}$	M = 42, 44 D = 2
	$50 \text{ MHz} < F_{\text{IN}} < 75 \text{ MHz}$	M = 56 D = 4
	$75 \text{ MHz} < F_{\text{IN}} < 100 \text{ MHz}$	M = 42, 44 D = 4
	$100 \text{ MHz} < F_{\text{IN}} < 150 \text{ MHz}$	M = 42, 44 D = 6
CLKOUT[3:2]_DIVIDE	N/A	
CLKOUT[6:4,1,0]_DIVIDE	1 to 128	
Bandwidth	Low	

When using spread-spectrum generation, the VCO frequency is set by the clocking wizard based on the input frequency and SS_MODE. As a result, the clocking wizard is recommended to set the output frequencies for CLKOUT[6:4,1,0].

Based on the VCO frequency and SS_MOD_PERIOD, the clocking wizard also determines the correct modulation settings to set the modulation frequency within 10% of SS_MOD_PERIOD. Because the modulation frequency is dependent on the VCO frequency, the modulation frequency scales as the input frequency changes for a given compilation.

MMCM Application Example

These MMCM attribute settings result in a wide variety of synthesized clocks:

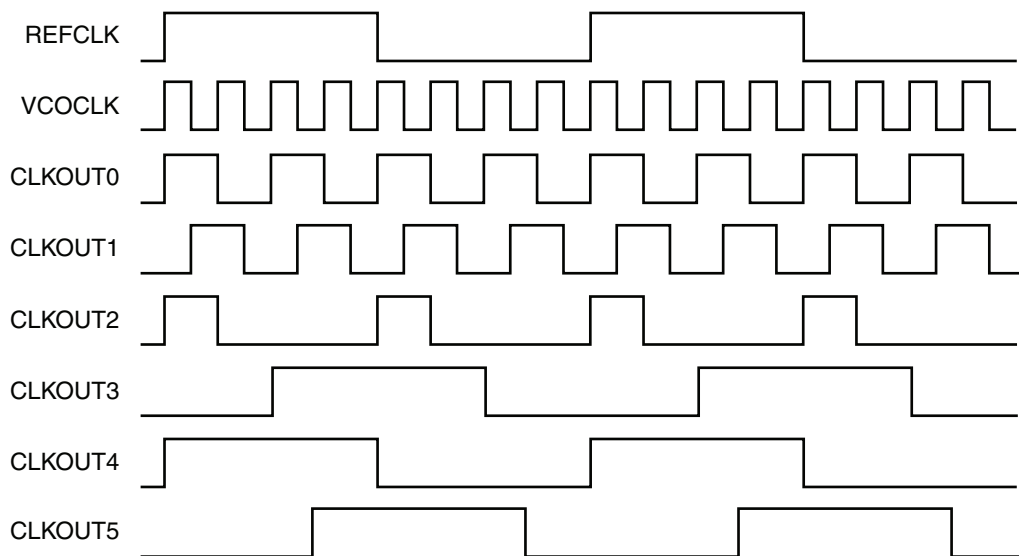
```

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_F = 8;
DIVCLK_DIVIDE = 1;
CLKIN1_PERIOD = 10.0;
    
```

Figure 3-15 displays the resulting waveforms.



UG572_c2_14_110112

Figure 3-15: Example Waveform

PLLs

There are two PLLs per CMT that provide clocking to the PHY logic and I/Os. In addition, they can be used as frequency synthesizers for a wide range of frequencies, serve as jitter filters, and provide basic phase shift capabilities and duty cycle programming. The PLLs differ from the MMCM in number of outputs, cannot deskew clock nets, and do not have advanced phase shift capabilities. Multipliers and input dividers have a smaller value range and do not have many of the other advanced features of the MMCM.

General Usage Description

PLL Primitives

The two PLL primitives, PLLE3_BASE and PLLE3_ADV, are shown in [Figure 3-16](#).

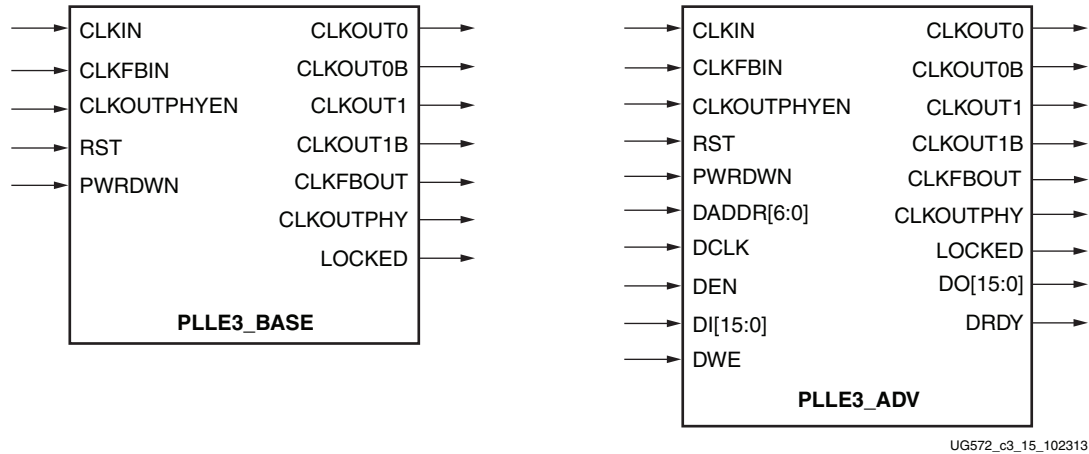


Figure 3-16: PLLE3_BASE and PLLE3_ADV

PLLE3_BASE Primitive

The PLLE3_BASE primitive provides access to the most frequently used features of a stand-alone PLL. Clock deskew, frequency synthesis, and duty cycle programming are available to use with the PLLE3_BASE. The ports are listed in [Table 3-7](#).

Table 3-7: PLLE3_BASE Ports

Description	Ports
Clock Input	CLKIN, CLKFBIN
Control Inputs	RST, CLKOUTPHYEN
Clock Output	CLKOUT0, CLKOUT1, CLKOUT0B, CLKOUT1B, CLKOUTPHY, CLKFBOUT
Status and Data Outputs	LOCKED
Power Control	PWRDWN

PLLE3_ADV Primitive

The PLLE3_ADV primitive provides access to all PLLE3_BASE features plus additional ports for access to the DRP. The ports are listed in [Table 3-8](#).

Table 3-8: PLLE3_ADV Ports

Description	Ports
Clock Input	CLKIN, DCLK, CLKFBIN
Control and Data Input	RST, CLKOUTPHYEN, DWE, DEN, DADDR, DI

Table 3-8: PLLE3_ADV Ports (Cont'd)

Description	Ports
Clock Output	CLKOUT0, CLKOUT1, CLKOUT0B, CLKOUT1B, CLKOUTPHY, CLKFBOUT
Status and Data Output	LOCKED, DO, DRDY
Power Control	PWRDWN

PLL Ports

Table 3-9 summarizes the PLL ports.

Table 3-9: PLL Ports

Pin Name	I/O	Pin Description
CLKIN	Input	General clock input.
RST	Input	Asynchronous reset signal. The RST signal is an asynchronous reset for the PLL. The PLL synchronously re-enables itself when this signal is released (i.e., PLL re-enabled). A reset is required when the input clock conditions change (e.g., frequency).
PWRDWN	Input	Powers down instantiated but unused PLLs. See PWRDWN – Power Down, page 64 .
CLKOUT[0:1] CLKOUT[0:1]B	Output	User configurable clock outputs 0 and 1 and their inverted versions. The CLKOUT can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 128.
CLKOUTPHYEN	Input	Enable PHY clocking.
CLKOUTPHY	Output	Dedicated PHY clock.
LOCKED	Output	An 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 is deasserted 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.
DADDR[6:0]	Input	The dynamic reconfiguration address (DADDR) input bus provides a reconfiguration address for the dynamic reconfiguration. When not used, all bits must be assigned zeros.
DI[15:0]	Input	The dynamic reconfiguration data input (DI) bus provides reconfiguration data. When not used, all bits must be set to zero.
DWE	Input	The dynamic reconfiguration write enable (DWE) input pin provides the write enable control signal to write the DI data into the DADDR address. When not used, DWE must be tied Low.
DEN	Input	The dynamic reconfiguration enable (DEN) provides the enable control signal to access the dynamic reconfiguration feature. When the dynamic reconfiguration feature is not used, DEN must be tied Low.
DCLK	Input	The DCLK signal is the reference clock for the dynamic reconfiguration port.

Table 3-9: PLL Ports (Cont'd)

Pin Name	I/O	Pin Description
DO[15:0]	Output	The dynamic reconfiguration output bus provides PLL data output when using dynamic reconfiguration.
DRDY	Output	The dynamic reconfiguration ready output (DRDY) provides the response to the DEN signal for the PLL's dynamic reconfiguration feature.

PLL Port Descriptions

CLKIN – Reference Clock Input

This is a general clock input to the PLL.

RST – Asynchronous Reset Signal

The RST signal is an asynchronous reset for the MMCM. The MMCM is synchronously re-enabled when this signal is deasserted.

PWRDWN – Power Down

This signal powers down instantiated but currently unused MMCMs. This mode can be used to save power for temporarily inactive portions of the design and/or MMCMs that are not active in certain system configurations. No MMCM power is consumed in this mode.

CLKOUT[0:1] – Output Clocks

These are user-configurable clock outputs and can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 128. The input clock and output clocks can be phase aligned.

CLKOUTPHY – PHY Clock Output

This is a dedicated clock output for use by the PHY byte logic and I/O. It can be 2X, 1X, or 0.5X of the VCO frequency.

CLKOUTPHYEN – PHY Clock Enable

CLKOUTPHYEN enables the CLKOUTPHY_P and CLKOUTPHY_N clock outputs. All CLKOUT outputs can be used to provide a static phase shift.

CLKOUT[0:1]B – Inverted Output Clocks

This is the inverted (180° phase shift) CLKOUT[0:1].

LOCKED

This output from the PLL is used to indicate when the PLLs have achieved frequency alignment of the reference clock and the internal feedback. Frequency alignment is within a predefined window of frequency matching within a predefined PPM range. The PLL automatically locks after power on; no extra reset is required. LOCKED is deasserted within one PFD clock cycle if the input clock stops or the frequency has changed. The PLL must be reset when LOCKED is deasserted. The clock outputs should not be used prior to the assertion of LOCKED.

DADDR[6:0] – Dynamic Reconfiguration Address

The dynamic reconfiguration address (DADDR) input bus provides a reconfiguration address for the dynamic reconfiguration. The address value on this bus specifies the 16 configuration bits that are written or read with the next DCLK cycle. When not used, all bits must be assigned zeros.

DI[15:0] – Dynamic Reconfiguration Data Input

The dynamic reconfiguration data input (DI) bus provides reconfiguration data. The value of this bus is written to the configuration cells. The data is presented in the cycle that DEN and DWE are active. The data is captured in a shadow register and written at a later time. DRDY indicates when the DRP port is ready to accept another write. When not used, all bits must be set to zero.

DWE – Dynamic Reconfiguration Write Enable

The dynamic reconfiguration write enable (DWE) input pin provides the write/read enable control signal to write the DI data into or read the DO data from the DADDR address. When not used, DWE must be tied Low.

DEN – Dynamic Reconfiguration Enable Strobe

The dynamic reconfiguration enable strobe (DEN) provides the enable control signal to access the dynamic reconfiguration feature and enable all DRP port operations. When the dynamic reconfiguration feature is not used, DEN must be tied Low.

DCLK – Dynamic Reconfiguration Reference Clock

DCLK is the reference clock for the dynamic reconfiguration port. The rising edge of this signal is the timing reference for all other port signals. The setup time is specified in the UltraScale device data sheets [Ref 4]. There is no hold time requirement for the other input signals relative to the rising edge of DCLK. This signal can be driven by an IBUF, IBUFG, BUFGCE, or BUFGCTRL. There are no dedicated connections to this clock input.

DO[15:0] – Dynamic Reconfiguration Output Bus

The dynamic reconfiguration output bus provides MMCM data output when using dynamic reconfiguration. If DWE is inactive while DEN is active at the rising edge of DCLK, this bus holds the content of the configuration cells addressed by DADDR. The DO bus must be captured on the rising edge of DCLK when DRDY is active. The DO bus value is held until the next DRP operation.

DRDY – Dynamic Reconfiguration Ready

The dynamic reconfiguration ready output (DRDY) provides the response to the DEN signal for the MMCM's dynamic reconfiguration feature. This signal indicates that a DEN/ DCLK operation has completed.

PLL Attributes

Table 3-10 lists the attributes for the PLLE3_BASE and PLLE3_ADV primitives.

Table 3-10: PLL Attributes

Attribute	Type	Allowed Values	Default	Description
CLKOUT[0:1]_DIVIDE	Integer	1 to 128	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 values, determines the output frequency.
CLKOUT[0:1]_DUTY_CYCLE	Real	0.01 to 0.99	0.50	Specifies the duty cycle of the associated CLKOUT clock output in percentages (i.e., 0.50 generates a 50% duty cycle).
CLKFBOUT_MULT	Integer	1 to 19	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, determines the output frequency.
DIVCLK_DIVIDE	Integer	1 to 16	1	Specifies the division ratio for all output clocks with respect to the input clock.

Table 3-10: PLL Attributes (Cont'd)

Attribute	Type	Allowed Values	Default	Description
REF_JITTER	Real	0.000 to 0.999	0.010	Allows specification of the expected jitter on the reference clock to better optimize PLL performance. A bandwidth setting of OPTIMIZED attempts to choose the best parameter for input clocking when unknown. If known, the value provided should be specified in terms of the unit interval (UI) (the maximum peak-to-peak value) of the expected jitter on the input clock.
CLKIN_PERIOD	Real	0.938 to 52.631	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.
STARTUP_WAIT	Boolean	FALSE, TRUE	FALSE	Wait during the configuration startup cycle for the PLL to lock.
CLKOUT[0:1]_PHASE	Real	-360.000 to 360.000	0.000	Allows specification of the output phase relationship of the associated CLKOUT clock output in number of degrees offset (i.e., 90 indicates a 90° offset or ¼ cycle phase offset while 180 indicates a 180° offset or ½ cycle phase offset).
CLKFBOUT_PHASE	Real	-360.000 to 360.000		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.

Table 3-10: PLL Attributes (Cont'd)

Attribute	Type	Allowed Values	Default	Description
COMPENSATION	String	AUTO ⁽¹⁾ , BUF_IN, INTERNAL	AUTO	Clock input compensation. Must be set to AUTO. Defines how the PLL feedback is configured. INTERNAL: Indicates that the PLL is using its own internal feedback path so no delay is being compensated. BUF_IN: Indicates that the configuration does not match with the other compensation modes and no delay is compensated.
CLKOUTPHY_MODE	String	VCO_2X, VCO, VCO_HALF	VCO_2X	Determines the clock output frequency based on the VCO frequency for the BITSlice_CONTROL block.

Notes:

1. The COMPENSATION attribute values are documented for informational purpose only. The Vivado tools automatically select the appropriate compensation based on circuit topology. Do not manually select a compensation value, leave the attribute at the default value.

VHDL and Verilog Templates and the Clocking Wizard

The VHDL and Verilog code for all clocking resource primitives and Vivado tools language templates are available in the Libraries Guide.

The Clocking Wizard helps to correctly set up the MMCM and PLL resources. Additionally, the Clocking Wizard reports the jitter and supports phase and frequency synthesis.

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see [Xilinx Support](#).

For a glossary of technical terms used in Xilinx documentation, see the [Xilinx Glossary](#).

Solution Centers

See the [Xilinx Solution Centers](#) for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

1. *UltraScale Architecture Packaging and Pinout User Guide* ([UG575](#))
2. *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#))
3. *Vivado Design Suite User Guide: Using Constraints* ([UG903](#))
4. UltraScale device data sheets:
 - *Kintex UltraScale Architecture Data Sheet: DC and AC Switching Characteristics* ([DS892](#))
5. *UltraScale Architecture and Product Overview* ([DS890](#))