Using the ClockLock & ClockBoost PLL Features

in APEX Devices

November 2003, ver. 2.6 Application Note 115

Introduction

APEX™ 20K devices have the ClockLock™ and ClockBoost™ features, which use phase-locked loops (PLLs) to increase performance and provide clock-frequency synthesis. The ClockLock feature minimizes clock delay and clock skew within the device, reducing clock-to-output and setup times while maintaining zero hold times. The ClockBoost feature allows designers to run the internal logic of the device at a faster or slower rate than the input clock frequency. This technique simplifies board design because the clock tree on the board does not have to distribute a high-speed signal. Through the use of time-domain multiplexing, the ClockBoost feature allows the designer to improve device area efficiency by sharing resources within the device.

APEX 20KE devices include PLLs with an enhanced ClockLock feature set, such as advanced ClockBoost capability for $m/(n \times k)$ multiplication, LVDS support, external clock outputs and feedback ability, and ClockShift™ circuitry for more complex clock-frequency synthesis applications. These enhanced features permit system-level clock management and skew control in APEX 20KE devices.

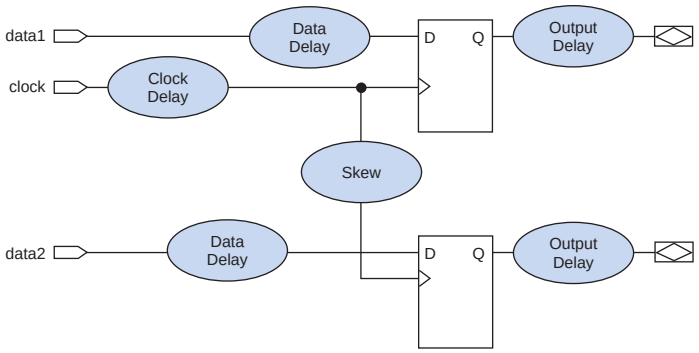
The ClockLock and ClockBoost features provide significant improvements in system performance, bandwidth, and system-on-a-programmable-chip (SOPC) integration. This application note explains the APEX 20K and APEX 20KE ClockLock and ClockBoost features and also describes common applications for these features.

Clock Delay & Skew

The delay from a clock pin to a register, especially for large devices, can be significant enough to degrade both on- and off-chip performance. Figure 1 shows the equation for the pin-to-pin clock-to-output delay (t_{CO}).

The clock delay (t_{CLOCK}) and clock skew (t_{SKEW}) parameters account for a significant portion of the total clock-to-output delay in larger devices. By reducing clock delay and clock skew, the ClockLock and ClockBoost circuitry improves the device's clock-to-output times.

Figure 1. APEX 20K & APEX 20KE Hold, Setup & Clock-to-Output Times



$t_H = t_{REG\_H} + t_{CLOCK} + t_{SKEW} - t_{DATA}$

$t_{SU} = t_{REG\_SU} + t_{DATA} - t_{CLOCK} - t_{SKEW}$

$t_{CO} = t_{CLOCK} + t_{SKEW} + t_{REG\_CO} + t_{OUTPUT}$

Clock skew—the difference between clock delays to different registers—also increases the setup time indirectly. To ensure a zero hold time (t_H), add a data delay to account for the longest clock delay to any register. This delay must be long enough to ensure a zero hold time under fast process, voltage, and temperature conditions. However, the added data delay also increases the register's setup time under slow process, voltage, and temperature conditions. When a ClockLock signal feeds a register, the signal bypasses the data delay element to the register, resulting in a decreased setup time. Because clock skew and delay are reduced, the register maintains a zero hold time.

As programmable logic devices (PLDs) become larger, clock delay and skew can become a problem. Clock skew can also affect a board's design. To address these issues, designers can use either PLLs or delay-locked loops (DLLs). Although both can reduce the skew within system clocks, PLLs are more flexible than DLLs for frequency synthesis of system clocks. In addition, DLLs are not capable of performing non-integer scaling, and they cannot attenuate input jitter. In DLLs, all input jitter propagates to the output and accumulates when the DLLs are cascaded. Because PLLs can perform non-integer scaling, they are ideal for clock multiplication and division applications.

Table 1 shows the ClockLock features for APEX 20K and APEX 20KE devices.

Table 1. APEX 20K & APEX 20KE ClockLock Features *Note (1)*

Device	Number of PLLs	ClockBoost Feature	Number of External Clock Outputs	Number of Feedback Inputs	ClockShift	T1/E1 Conversion	LVDS Clock	LVDS Data
EP20K100	1	1×, 2×, 4×	–	–	–	–	–	–
EP20K200	1	1×, 2×, 4×	–	–	–	–	–	–
EP20K400	1	1×, 2×, 4×	–	–	–	–	–	–
EP20K30E	2	$m/(n \times k)$ (2), (3)	1	1	✓	✓	✓	–
EP20K60E	2	$m/(n \times k)$ (2), (3)	1	1	✓	✓	✓	–
EP20K100E	2	$m/(n \times k)$ (2), (3)	1	1	✓	✓	✓	–
EP20K160E	2	$m/(n \times k)$ (2), (3)	1	1	✓	✓	✓	–
EP20K200E	2	$m/(n \times k)$ (2), (3)	1	1	✓	✓	✓	–
EP20K300E	4	$m/(n \times k)$ (2), (3)	2	2	✓	✓	✓	(4)
EP20K400E	4	$m/(n \times k)$ (2), (3)	2	2	✓	✓	✓	✓
EP20K600E	4	$m/(n \times k)$ (2), (3)	2	2	✓	✓	✓	✓
EP20K1000E	4	$m/(n \times k)$ (2), (3)	2	2	✓	✓	✓	✓
EP20K1500E	4	$m/(n \times k)$ (2), (3)	2	2	✓	✓	✓	✓

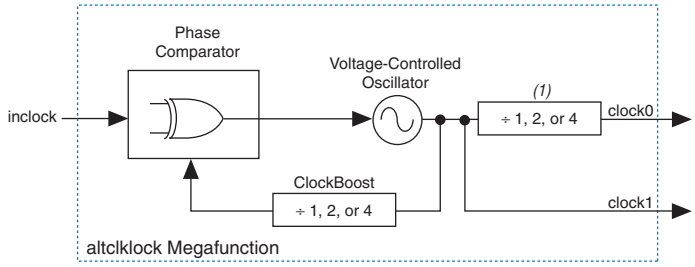
Notes:

(1) APEX 20K devices that support ClockLock and ClockBoost features are denoted by an "X" suffix in the ordering code (e.g., EP20K400FC672-1X).
(2) m is an integer that ranges from 1 to 160.
(3) The quantity $m/(n \times k)$ can range from 1 to 280. A special multiplication rate is also provided for T1 to E1 frequency conversion (i.e., 256/193), and E1 to T1 frequency conversion (i.e., 193/256).
(4) The EP20K300E device supports LVDS data transfer at up to 155 megabits per second (Mbps).

APEX 20K Devices

APEX 20K devices have one PLL that features ClockLock and ClockBoost circuitry. This PLL can be instantiated by using the `altclklock` megafunction. APEX 20K devices support ClockBoost multiplication circuitry, offering 1×, 2×, and 4× clock multiplication. Figure 2 shows the ClockLock and ClockBoost circuitry block diagrams within the `altclklock` megafunction and its ports.

Figure 2. ClockLock & ClockBoost Circuitry in APEX 20K Devices



Note:
(1) This division is used only for the purpose of dividing down a 2×/4× `clock1` to obtain a 1×/2×/4× on `clock0`.

You can use a single output clock of 1×, 2×, 4×, or any combination of output clocks. Table 2 describes the clock multiplication combinations that the `altclklock` megafunction supports for APEX 20K devices.

Table 2. APEX 20K Multiplication Factor Combinations

Clock 0	Clock 1	Input Frequency (MHz)	
		-1X Speed Grade	-2X Speed Grade
1×	2×	25 to 180	25 to 170
1×, 2×	4×	16 to 90	16 to 100
1×, 2×, 4×	4×	10 to 48	10 to 34

The dedicated clock pin (`CLK2`) supplies the clock to the PLL and the `altclklock` megafunction. Adhere to the following guidelines when using the `altclklock` megafunction in APEX 20K devices:

- The `inclock` port must only be fed directly by the dedicated clock input pin (`CLK2`) without inversion.
- `altclklock` can only be used to clock positive or negative edge-triggered registers in logic elements (LEs), I/O elements (IOEs), or embedded system blocks (ESBs). IOE registers can only have single clock polarity from a given `altclklock` output (i.e., IOE registers must be all positive-edge triggered or all negative-edge triggered). If both clock polarities are needed for IOE, use two PLL outputs.
- The `CLK2` pin that directly feeds the `inclock` port can also drive other registers without the PLL. However, doing so makes the `CLK1` pin and the `clock1` port unavailable.
- When two PLL outputs are used, the other clock pin (`CLK1`) cannot be used.

You should connect the board clock trace to only the CLK2 pin for designs that require two outputs from the altclklock megafunction. Figure 3 illustrates the valid clock connections for the PLL and global clock lines.

Figure 3. APEX 20K Dedicated Global Clock Pin Connections to PLL & Dedicated Clock Lines

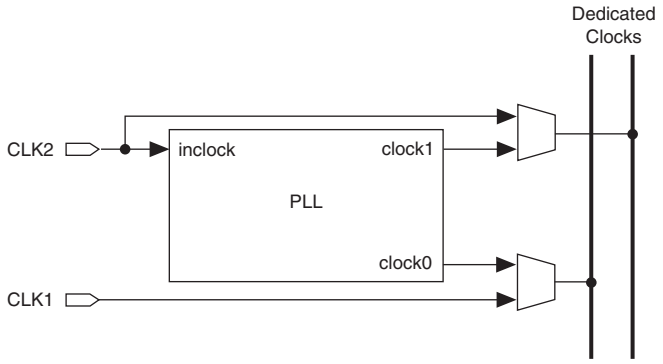


Table 3 shows the timing parameters for the APEX 20K ClockLock and ClockBoost features for -1 speed-grade devices and Table 4 shows the timing parameters for the APEX 20K ClockLock and ClockBoost features for -2 speed-grade devices.

Table 3. APEX 20K ClockLock & ClockBoost Parameters for -1X Speed-Grade Devices (Part 1 of 2)

Symbol	Parameter	Min	Max	Unit
f_{OUT}	Output frequency	25	180	MHz
f_{CLK1} *(1)*	Input clock frequency (ClockBoost clock multiplication factor equals 1)	25	180 *(1)*	MHz
f_{CLK2}	Input clock frequency (ClockBoost clock multiplication factor equals 2)	16	90	MHz
f_{CLK4}	Input clock frequency (ClockBoost clock multiplication factor equals 4)	10	48	MHz
t_{INDUTY}	Input duty cycle	40	60	%
$t_{OUTDUTY}$	Duty cycle for ClockLock/ClockBoost-generated clock	40	60	%

Table 3. APEX 20K ClockLock & ClockBoost Parameters for -1X Speed-Grade Devices (Part 2 of 2)

Symbol	Parameter	Min	Max	Unit
f_{CLKDEV}	Input deviation from user specification in the Quartus II software (ClockBoost clock multiplication factor equals 1) *(2)*		25,000 *(3)*	PPM
t_R	Input rise time		5	ns
t_F	Input fall time		5	ns
t_{LOCK}	Time required for ClockLock/ClockBoost to acquire lock *(4)*		10	µs
t_{SKEW}	Skew delay between related ClockLock/ClockBoost-generated clocks		500	ps
t_{JITTER}	Jitter on ClockLock/ClockBoost-generated clock *(5)*		200	ps
$t_{INCLKSTB}$	Input clock stability (measured between adjacent clocks)		50	ps

Notes:
(1) The PLL input frequency range for the EP20K100-1X device for 1× multiplication is 25 MHz to 175 MHz.
(2) All input clock specifications must be met. The PLL may not lock onto an incoming clock if the clock specifications are not met, creating an erroneous clock within the device.
(3) During device configuration, the ClockLock and ClockBoost circuitry is configured first. If the incoming clock is supplied during configuration, the ClockLock and ClockBoost circuitry locks during configuration, because the lock time is less than the configuration time.
(4) The jitter specification is measured under long-term observation.
(5) If the input clock stability is 100 ps, t_{JITTER} is 250 ps.

Table 4. APEX 20K ClockLock & ClockBoost Parameters for -2X Speed –Grade Devices (Part 1 of 2)

Symbol	Parameter	Min	Max	Unit
f_{OUT}	Output frequency	25	170	MHz
f_{CLK1}	Input clock frequency (ClockBoost clock multiplication factor equals 1)	25	170	MHz
f_{CLK2}	Input clock frequency (ClockBoost clock multiplication factor equals 2)	16	80	MHz
f_{CLK4}	Input clock frequency (ClockBoost clock multiplication factor equals 4)	10	34	MHz

Table 4. APEX 20K ClockLock & ClockBoost Parameters for -2X Speed –Grade Devices (Part 2 of 2)

Symbol	Parameter	Min	Max	Unit
$t_{OUTDUTY}$	Duty cycle for ClockLock/ClockBoost-generated clock	40	60	%
f_{CLKDEV}	Input deviation from user specification in the Quartus II software (ClockBoost clock multiplication factor equals one) *(1)*		25,000 *(2)*	PPM
t_R	Input rise time		5	ns
t_F	Input fall time		5	ns
t_{LOCK}	Time required for ClockLock/ ClockBoost to acquire lock *(3)*		10	µs
t_{SKEW}	Skew delay between related ClockLock/ ClockBoost-generated clock	500	500	ps
t_{JITTER}	Jitter on ClockLock/ ClockBoost-generated clock *(4)*		200	ps
$t_{INCLKSTB}$	Input clock stability (measured between adjacent clocks)		50	ps

Notes:

(1) To implement the ClockLock and ClockBoost circuitry with the Quartus II software, designers must specify the input frequency. The Quartus II software tunes the PLL in the ClockLock and ClockBoost circuitry to this frequency. The f_{CLKDEV} parameter specifies how much the incoming clock can differ from the specified frequency during device operation. Simulation does not reflect this parameter.

(2) 25,000 parts per million (PPM) equates to 2.5% of input clock period.

(3) During device configuration, the ClockLock and ClockBoost circuitry is configured before the rest of the device. If the incoming clock is supplied during configuration, the ClockLock and ClockBoost circuitry locks during configuration because the t_{LOCK} value is less than the time required for configuration.

(4) The t_{JITTER} specification is measured under long-term observation.

Table 5 lists the APEX 20K ClockLock pins and their functions.

Table 5. APEX 20K Device ClockLock Pins

Pin Name	Pin Type	Description	I/O Standards Supported *(1)*
CLK2	Input	Dedicated pin that drives the ClockLock and ClockBoost circuitry.	2.5-V I/O, LVCMOS, LVTTL, 3.3-V PCI, CMOS *(2)*, TTL *(2)*
LOCK	Output	Optional pin that shows the status of the ClockLock and ClockBoost circuitry. When the ClockLock and ClockBoost circuitry is locked to the incoming clock and generates an internal clock, LOCK is driven high. LOCK remains high as long as the clock input remains within specification.	

Notes:
(1) PCI: peripheral component interconnect.
(2) APEX 20K devices with a "V" suffix in their ordering code are 5.0-V tolerant.

APEX 20KE Devices

APEX 20KE devices incorporate multiple ClockLock circuits with advanced features. These features include ClockLock circuitry, advanced ClockBoost circuitry, LVDS support, ClockShift circuitry, and external clock outputs with optional external feedback inputs.

Advanced ClockBoost Multiplication & Division

Each APEX 20KE PLL includes circuitry that provides clock synthesis for two outputs using $m/(n \times k)$ and $m/(n \times v)$ scaling factors. When a PLL is locked, the locked output clock aligns to the rising edge of the input clock. The closed loop equation for Figure 3 gives an output frequency of $f_{CLOCK0} = (m/(n \times k)) f_{IN}$ and $f_{CLOCK1} = (m/(n \times v) f_{IN}$. This equation allows the multiplication or division of clocks by a programmable number. The Quartus II software automatically chooses the appropriate scaling factors (m, n, k, and v) according to the frequency, multiplication, and division values entered.

The scaling multiplication of APEX 20KE PLLs allows for a wide range of user-defined multiplication and division ratios that are not possible with DLLs. For example, if a frequency scaling factor of 3.75 is needed for a given input clock, you can enter a multiplication factor of 15 and a division factor of four. Because this advanced ClockBoost scaling can be performed with a single PLL, you do not need to cascade PLL outputs.

All APEX 20KE PLLs are capable of converting between T1 and E1 clock frequencies. The T1 telecommunications standard uses a 1.544-MHz clock, and the E1 telecommunications standard uses a 2.048-MHz clock. APEX 20KE PLLs can convert a T1 frequency to an E1 frequency and vice versa.

LVDS Interface

In EP20K400E and larger devices, two of the ClockLock PLLs (PLL3 and PLL4) can be configured for use in LVDS transmitter and receiver interfaces. When the PLL is configured to use LVDS, the I/O clock can be multiplied to support high-speed data transfer rates and to convert between LVDS and CMOS data. These PLLs interface with the APEX 20KE LVDS receiver and LVDS transmitter blocks.

When used for LVDS transmitter and receiver modes, PLLs 3 and 4 use connections shown at the top of the Figure 7. These modes require the use of the `ALTLVDS` megafuntion.

See *Application 120 (Using LVDS in APEX 20KE Devices)* for more information on LVDS.

External Clock Outputs

In EP20K300E and larger devices, low-jitter external clocks, `CLKLK_OUT1P` and `CLKLK_OUT2P`, are available for external clock sources. In EP20K30E, EP20K60E, EP20K100E, EP20K160E, and EP20K200E devices, one external clock, `CLKLK_OUT2P`, is available for an external clock source. The `CLKLK_OUT1P` signal originates from PLL 1; the `CLKLK_OUT2P` signal originates from PLL 2. Other devices on the board can use these outputs as clock sources.

External clock outputs are available in three modes:

- *Zero Delay Buffer*—The external clock output pin is phase aligned with the clock input pin for zero delay. Phase shift is not allowed in this configuration. Clock division is possible on the external clock output in this configuration. Multiplication is possible on the remaining PLL output that is not driven off-chip. The `altclklock` megafunction's MegaWizard® Plug-In should be used to verify possible clock settings.
- *External Feedback*—The external feedback input pin is phase aligned with the clock input pin. By aligning these clocks, you can actively remove clock delay and skew between devices. This mode has the same restrictions as the zero delay buffer.

- *Normal Mode*—The external clock output pin will have phase delay relative to the clock input pin. If an internal clock is used in this mode, it will be phase aligned to the input clock pin.

ClockShift Circuitry

APEX 20KE PLLs have ClockShift circuitry that provides programmable clock delay and phase shift. The user enters the desired phase shift. Phase shifts of 90°, 180°, and 270° can be implemented exactly. Other phase shift or delay shift values (in time units) are available with a 0.4 ns to 1.0 ns resolution range. This resolution varies with frequency input and the user-entered multiplication and division factors. The ClockShift circuitry can only be used on a multiplied or divided clock if the input and output frequency have an integer multiple relationship (i.e., f_{IN}/f_{OUT} or f_{OUT}/f_{IN} must be an integer). The PLL ClockShift circuitry is only available in Normal mode.

Clock Enable Signal

APEX 20KE PLLs have a `CLKLK_ENA` pin for enabling and disabling all of the device PLLs. When the `CLKLK_ENA` pin is high, the PLL drives the `clock0` and `clock1` ports. When the `CLKLK_ENA` pin is low, the `clock0` and `clock1` ports are driven by `GND` and all of the PLLs go out of lock. When the `CLKLK_ENA` pin goes high again, the PLL must relock.

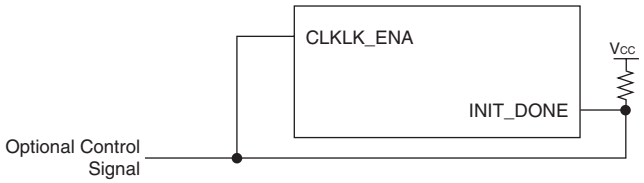
The individual enable port on the `altclklock` megafunction is an all-or-none control signal. If the enable port is used for any `altclklock` instance, all other instances must use the enable port and connect to the same pin. If the port is used, the pin connected to the enable ports must be the `CLKLK_ENA` dedicated pin.

The `inclocken` input port on the `altclklock` megafunction should be used for all designs. If the `CLKLK_ENA` control is needed by the designer, it must be brought to the `CLKLK_ENA` input pin. The Quartus II software automatically assigns the pin location to this dedicated function pin.

If the `CLKLK_ENA` control is not needed by the user in Zero Delay Buffer and Normal modes, the `inclocken` port should not be connected to an input pin in the design. On the board, this pin can be connected to V_{CC} or GND since it will be a floating input.

If the CLKLK_ENA control is not needed by the user in External Feedback mode, the inclocken port must be connected to an input pin. On the board, the user must connect this pin directly to the INIT_DONE pin. The INIT_DONE option must be set in the **Device & Pin Options** tab of the **Compiler Settings > Chips & Devices** dialog box (Processing menu). If system control over the CLKLK_ENA pin is needed in External Feedback mode, an open-drain control signal can be used in a wire-ANDed configuration with the open-drain INIT_DONE pin. Figure 4 demonstrates how this should be connected.

Figure 4. Wire-ANDing with the Open-Drain INIT_DONE Pin



Lock Signals

The APEX 20KE ClockLock PLL circuits support individual LOCK signals. The LOCK signal drives high when the PLL has locked onto the input clock. LOCK remains high as long as the input remains within specification. It will go low if the input is out of specification. When using two clock outputs from a PLL, the two outputs may become valid at different times. This difference is caused by the difference in divider ratios between the two outputs. A LOCK pin is optional for each PLL used in APEX 20KE devices; when not used, they are I/O pins. This signal is not available internally. If this signal is used in the design, it must be fed back in with an input pin.

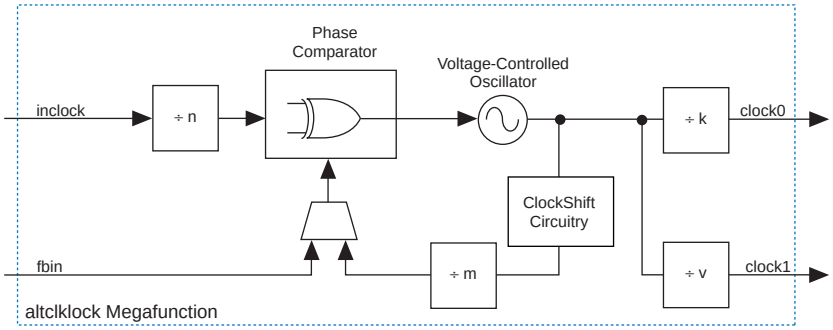
Before configuration, all PLL circuits are disabled and powered down. During configuration, the PLLs are disabled. When the device goes into user mode, the lock time is measured from the CLKLK_ENA rising edge to the LOCK rising edge.

During device configuration, the I/O standard of each I/O pin has not been set. For this reason, the PLL does not lock during configuration.

The LOCK signal indication also has separate programmable latency controls for LOCK assertion and deassertion. Users can select one of two multipliers, 1 or 5, to indicate a small or large latency. The number of half cycles for indication is dependent on the multiplication/division ratio and input frequency. Depending on these factors, the low end of latency ranges from 1 to 16 half-clock cycles; the high end of latency ranges from 5 to 80 half-clock cycles.

The APEX 20KE ClockLock circuits (general purpose PLLs) are instantiated using the altclklock megafunctions. Figure 5 shows the ClockLock and ClockBoost circuitry in APEX 20KE devices.

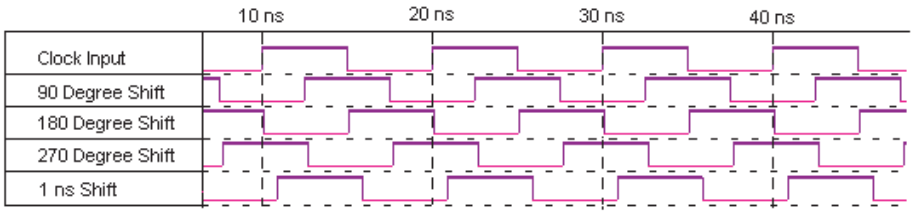
Figure 5. ClockLock & ClockBoost Circuitry in APEX 20KE Devices



Each of the dedicated global clock pins in EP20K300E, EP20K400E, EP20K600E, EP20K1000E, and EP20K1500E devices (`CLK1P`, `CLK2P`, `CLK3P`, `CLK4P`) supplies the clock to a PLL in general purpose mode. Each `altclklock` instance represents a single general-purpose PLL instantiation. The `altclklock` megafunction in APEX 20KE devices must adhere to the following guidelines:

- It can only be fed directly by a dedicated clock input pin without inversion.
- `altclklock` can only be used to clock positive or negative edge-triggered registers in LEs, IOEs, or ESBs. IOE registers can only have single clock polarity from a given `altclklock` output (i.e., IOE registers must be all positive-edge triggered or all negative-edge triggered). If both clock polarities are needed for IOE, use two PLL outputs.
- The allowable frequency input range is 1.5 to 420 MHz, depending on the I/O standard (see Table 7).
- The allowable frequency output range on `clock0` is 1.5 to 420 MHz, depending on the I/O standard for the external clock out and/or the internal max frequency shown in Table 7.
- The allowable frequency output range on `clock1` is 12.5 to 420 MHz, depending on the I/O standard for the external clock out and/or the internal max frequency shown in Table 7.
- Phase shifting is only possible on a multiplied clock if the input and output frequency have an integer-multiple relationship (i.e., f_{IN}/f_{OUT} or f_{OUT}/f_{IN} must be an integer).
- Phase shifting, using degree or time units, will delay or lag the output clock with respect to the input clock (see Figure 6).
- The ratio of `clock_boost` to `clock_divide` cannot be greater than 280. A special scaling ratio of 256/193 or 193/256 is allowed for T1/E1 or E1/T1 clock rate conversion, respectively.

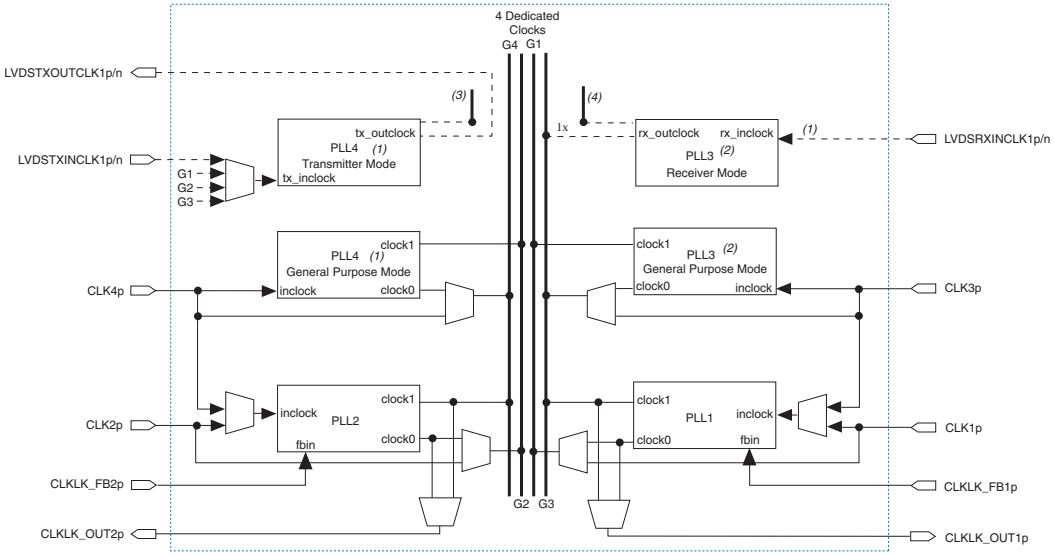
Figure 6. Phase & Delay Shifting Using APEX 20KE PLLs



Several conditions exist that govern the relationship between the input frequency, *m*, *n*, *k*, *v*, and phase shift values. The `altclklock` MegaWizard Plug-In automatically sets the *m*, *n*, *k*, and *v* dividers to satisfy these equalities and to accommodate the phase shift entered and the clock multiplication or division. The MegaWizard Plug-In verifies the validity of the settings and reports if a multiplication/division frequency ratio is not valid.

Each PLL can be driven by a dedicated clock pin and bypassed simultaneously. The `CLK3P` and `CLK4P` pins can feed two PLLs each, or two `altclklock` instances. This capability is useful for applications that need both phase-shifted and non-phase-shifted versions of the clock. Because the eight PLL outputs are shared among four possible dedicated global clock lines, certain combinations of multiple `altclklock` instances and their output connections are not possible. The Quartus II software uses different PLLs (i.e., 1, 2, 3, 4) based on pin assignments made to the dedicated global clock pin that feeds the `altclklock` megafunction. Figure 7 illustrates the valid clock connections for the PLL and the dedicated global clock lines in these devices. This figure should be used to determine whether a design clocking scheme is valid in terms of APEX 20KE PLL and clock connections. For example, in an EP20K400E device, `CLK4P` can feed PLL 4 and PLL 2 simultaneously; however, only a single internal output from each PLL can be used, because the four possible outputs feed two global clock lines.

Figure 7. Dedicated Global Clock Pin Connections to PLL & Dedicated Clock Lines for EP20K300E, EP20K400E, EP20K600E, EP20K1000E & EP20K1500E Devices



Notes:

(1) For EP20K400E, EP20K600E, EP20K1000E, and EP20K1500E devices, PLL 4 can be used in only one of two possible modes, general purpose mode or ALTLVDS_TX transmitter mode. Connections that apply to the chosen mode are shown above and are exclusive to that mode. Only one mode can be used at a time for PLL 4. PLL 4 in EP20K300E devices can only be used in general purpose mode.

(2) For EP20K400E, EP20K600E, EP20K1000E, and EP20K1500E devices, PLL 3 can be used in only one of two possible modes, general purpose mode or ALTLVDS_RX receiver mode. Connections that apply to the chosen mode are shown above and are exclusive to that mode. Only one mode can be used at a time for PLL 3. PLL 3 in EP20K300E devices can only be used in general purpose mode.

(3) This PLL output is a high-speed CMOS/LVDS interface clock that feeds the LVDS transmitter block.

(4) This PLL output is a high-speed LVDS/CMOS interface clock that feeds the LVDS receiver block.

For EP20K30E, EP20K60E, EP20K100E, EP20K160E, and EP20K200E devices, the CLK4P and CLK2P dedicated clock pins supply the clock to two PLLs. These PLLs have the same usage guidelines as the EP20K400E device and larger device's PLLs, with the exception of some PLL connections. You can use all four possible PLL output clocks that are shared among four dedicated clock lines. If you use all of the outputs, the CLK3P and CLK1P pins cannot be used.

Figure 8 illustrates the valid clock connections for the PLL and the global clock lines in these devices. Use Figure 8 to determine whether a design clocking scheme is valid in terms of PLL and clock connections. The altclklock megafunction port connections should follow the usage guidelines illustrated in Figures 7 and 8.

Figure 8. Dedicated Global Clock Pin Connections to PLL & Dedicated Clock Lines for EP20K30E, EP20K60E, EP20K100E, EP20K160E & EP20K200E Devices

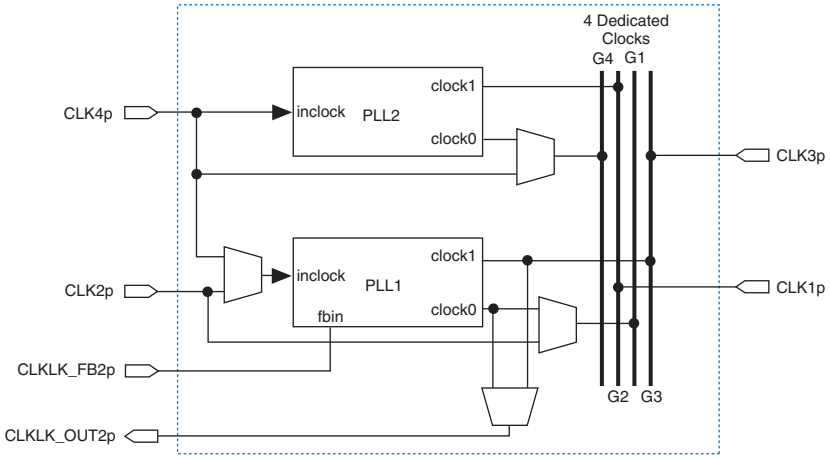


Table 6 shows the APEX 20KE ClockLock and ClockBoost parameters, and Table 7 shows the APEX 20KE Clock Input and Output parameters.

Table 6. APEX 20KE ClockLock & ClockBoost Parameters *Note (1)*

Symbol	Parameter	Condition	Min	Typ	Max	Unit
t_R	Input rise time				5	ns
t_F	Input fall time				5	ns
t_{INDUTY}	Input duty cycle		40		60	%
$t_{INJITTER}$	Input jitter peak-to-peak				2% of input period	peak-to-peak
$t_{OUTJITTER}$	Jitter on ClockLock or ClockBoost-generated clock				0.35% of output period	RMS
$t_{OUTDUTY}$	Duty cycle for ClockLock or ClockBoost-generated clock		45		55	%
t_{LOCK} *(2), (3)*	Time required for ClockLock or ClockBoost to acquire lock				40	µs

Table 7. APEX 20KE Clock Input and Output Parameters *Note (1)*

Symbol	Parameter	I/O Standard	-1X Speed Grade		-2X Speed Grade		Units
			Min	Max	Min	Max	
f_{IN} *(4), (5), (5)*	Input clock frequency	3.3-V LVTTL	1.5	290	1.5	257	MHz
		2.5-V LVTTL	1.5	281	1.5	250	MHz
		1.8-V LVTTL	1.5	272	1.5	243	MHz
		GTL+	1.5	303	1.5	261	MHz
		SSTL-2 Class I	1.5	291	1.5	253	MHz
		SSTL-2 Class II	1.5	291	1.5	253	MHz
		SSTL-3 Class I	1.5	300	1.5	260	MHz
		SSTL-3 Class II	1.5	300	1.5	260	MHz
		LVDS	1.5	420	1.5	350	MHz
f_{VCO} *(5)*	Voltage controlled oscillator operating range		200	500	200	500	MHz
f_{CLOCK0}	Clock0 PLL output frequency for internal use		1.25	335	1.25	200	MHz
f_{CLOCK1}	Clock1 PLL output frequency for internal use		12.5	335	20	200	MHz
f_{CLOCK0_EXT} *(4), (5), (5)*	Output clock frequency for external clock0 output	3.3-V LVTTL	1.25	245	1.25	226	MHz
		2.5-V LVTTL	1.25	234	1.25	221	MHz
		1.8-V LVTTL	1.25	223	1.25	216	MHz
		GTL+	1.25	205	1.25	193	MHz
		SSTL-2 Class I	1.25	158	1.25	157	MHz
		SSTL-2 Class II	1.25	142	1.25	142	MHz
		SSTL-3 Class I	1.25	166	1.25	162	MHz
		SSTL-3 Class II	1.25	149	1.25	146	MHz
		LVDS	1.25	420	1.25	350	MHz
f_{CLOCK1_EXT} *(4), (5), (5)*	Output clock frequency for external clock1 output	3.3-V LVTTL	12.5	245	20	226	MHz
		2.5-V LVTTL	12.5	234	20	221	MHz
		1.8-V LVTTL	12.5	223	20	216	MHz
		GTL+	12.5	205	20	193	MHz
		SSTL-2 Class I	12.5	158	20	157	MHz
		SSTL-2 Class II	12.5	142	20	142	MHz
		SSTL-3 Class I	12.5	166	20	162	MHz
		SSTL-3 Class II	12.5	149	20	146	MHz
		LVDS	12.5	420	20	350	MHz

Notes to tables:

(1) All input clock specifications must be met. The PLL may not lock onto an incoming clock if the clock specifications are not met, creating an erroneous clock within the device.
(2) The maximum lock time is 40 µs or 2000 input clock cycles, whichever occurs first.
(3) Before configuration, the PLL circuits are disabled and powered down. During configuration, the PLLs are still disabled. The PLLs begin to lock once the device is in the user mode. If the clock enable feature is used, lock begins once the CLKLK_ENA pin goes high in user mode.
(4) For the external feedback mode, the maximum clock frequency is either 165 MHz or the maximum clock output according to the I/O standard, whichever is smaller. For the EP20K400E device, the maximum external clock frequency for the external feedback mode is 124 MHz.
(5) For external feedback mode, the clock out, clock in, feedback input must have the same I/O standard.
(6) For zero delay buffer mode, the clock out and clock in must have the same I/O standard.
(7) The PLL voltage-controlled oscillator (VCO) operating range is wide enough to support the device's maximum LVDS data rate when in LVDS mode.b

Table 8 shows the APEX 20KE device LVDS mode PLL Parameters for EP20K400E and EP20K600E devices and Table 9 shows the APEX 20KE LVDS Mode PLL Parameters for EP20K1000E and EP20K1500E devices.

Table 8. EP20K400E & EP20K600E LVDS Mode PLL Parameters *Note (1)*

Symbol	Parameter	Mode	-1X Speed Grade			-2X Speed Grade			Units
			Min	Typ	Max	Min	Typ	Max	
t_C	LVDS receiver/transmitter input clock period	×8	9.52		33	11.43		33	ns
		×7	9.52		33	11.43		33	ns
		×4	5.71		20	6.88		20	ns
f_{INLVDS}	LVDS receiver/transmitter input clock frequency	×8	30		105	30		87.5	MHz
		×7	30		105	30		87.5	MHz
		×4	50		175	50		145.25	MHz
f_{LVDSDR}	Maximum LVDS data transfer rate	×8	240		840	240		700	Mbps
		×7	210		735	210		612.5	Mbps
		×4	200		700	200		581	Mbps
Input Jitter (peak-to-peak)	Peak-to-peak input Jitter on input clock	All			2% of t_C			2% of t_C	ns
Output Jitter (RMS)	RMS output jitter in LVDS mode *(2)*, *(3)*	All			0.25% of t_C			0.25% of t_C	ns
t_{DUTY}	Duty cycle on LVDS transmitter output clock	All	49	50	51	49	50	51	%
t_{LOCK}	Lock time for LVDS transmitter and receiver PLLs	All			5			5	µs

Table 9. EP20K1000E and EP20K1500E LVDS Mode PLL Parameters *Note (1)*

Symbol	Parameter	Mode	-1X Speed Grade			-2X Speed Grade			Units
			Min	Typ	Max	Min	Typ	Max	
t_C	LVDS receiver/transmitter Input clock period	×8	10.67		33	12.80		33	ns
		×7	9.52		33	11.43		33	ns
		×4	5.71		20	6.88		20	ns
f_{INLVDS}	LVDS receiver/transmitter Input clock frequency	×8	30		93.7	30		78.125	MHz
		×7	30		105	30		87.5	MHz
		×4	50		175	50		145.25	MHz
f_{LVDSDR}	Maximum LVDS data transfer rate	×8	240		750	240		625	Mbps
		×7	210		735	210		612.5	Mbps
		×4	200		700	200		581	Mbps
Input Jitter (peak-to-peak)	Peak-to-peak Input Jitter on input clock	All			2% of t_C			2% of t_C	ns
Output Jitter (RMS)	RMS Output Jitter in LVDS mode *(2)*, *(3)*	All			0.25% of t_C			0.25% of t_C	ns
t_{DUTY}	Duty cycle on LVDS transmitter output clock	All	49	50	51	49	50	51	%
t_{LOCK}	Lock time for LVDS transmitter and receiver PLLs	All			5			5	µs

Notes to tables:

(1) The PLL voltage-controlled oscillator (VCO) operating range is wide enough to support the device's maximum LVDS data rate when in LVDS mode.

(2) This assumes an input clock with 5 ps of input jitter

(3) This jitter is for both the receiver and the transmitter LVDS PLLs cascaded together. The jitter for the receiver or transmitter PLL will be less.

Table 10 lists the APEX 20KE ClockLock pins and their function.

Table 10. APEX 20KE Device ClockLock Pins

Pin Name	Pin Type	Description	I/O Standards Supported
CLKp[1..4]	Input	Dedicated pins that drive the PLL clock inputs. In EP20K400E and larger devices, CLK3p and CLK4p drive PLL3 and PLL4, respectively, for general-purpose mode only.	1.8-V I/O, 2.5-V I/O, AGP, CTT, HSTL, LVCMOS, LVDS, LVPECL, LVTTL, GTL+, 3.3-V PCI, 3.3-V PCI-X, SSTL-2, SSTL-3
CLKLK_FB	Input	Dedicated pins that allow external feedback to the PLLs.	
CLKLK_OUT	Output	Dedicated clock output that allows the PLL output to be driven off-chip.	
LVDSTXINCLK1p/n LVDSRXINCLK1p/n *(1)*	Input	Dual-purpose I/O pins that drive the PLL clock input in LVDS mode for LVDS/CMOS data conversion.	LVDS
LVDSTXOUTCLK1p/n *(1)*	Output	LVDS clock output that allows the PLL to drive the LVDS 1× clock off-chip in LVDS mode. LVDS output data is synchronized to this clock. These pins are dual-purpose I/O pins.	
LOCK	Output	Optional pin that shows the status of the ClockLock and ClockBoost circuitry. When the ClockLock and ClockBoost circuitry is locked to the incoming clock and generates an internal clock, LOCK is driven high. LOCK remains high as long as the clock input remains within specification.	1.8-V I/O, 2.5-V I/O, LVCMOS, LVTTL, 3.3-V PCI

Note:
(1) This pin is available in EP20K400E, EP20K600E, EP20K1000E, and EP20K1500E devices only.

Board Layout

Each PLL uses its own VCC and GND pins. APEX 20K devices have one pair of VCC and GND pins for the ClockLock and ClockBoost circuitry. APEX 20KE devices use a VCC and GND pair for each PLL and each clock output pin. Separate VCC and GND pins and external clock outputs reduce jitter by isolating the output pin from adjacent switching pins. It also minimizes any ground bounce or V_{CC} sag effects from nearby switching outputs.

Table 11 shows the power pins required for APEX 20K and APEX 20KE devices.

Table 11. Power Pin Requirements for APEX 20K & APEX 20KE ClockLock PLL Features

Device	Pin Name	Description
EP20K100 EP20K200 EP20K400	VCC_CKLK GND_CKLK	Power and ground pins for the ClockLock and ClockBoost PLL circuitry. To ensure noise resistance, the power and ground supply to the PLL should be isolated from the power and ground to the rest of the device. VCC_CKLK pins have the same specifications as VCCINT and should be connected to a 2.5 V supply, even if the PLL is not used.
EP20K60E EP20K100E EP20K160E EP20K200E	VCC_CKLK2 VCC_CKLK4 GND_CKLK2 GND_CKLK4	Power and ground pins for the ClockLock and ClockBoost PLL circuitry. To ensure noise resistance, the power and ground supply to the PLL should be isolated from the power and ground to the rest of the device. VCC_CKLK pins have the same specifications as VCCINT and should be connected to a 1.8 V supply, even if the PLL is not used.
	VCC_CKOUT2 GND_CKOUT2	Power and ground pins for the external clock output pins (CLKLK_OUT). These pins supply the VCCIO and GNDIO for the CLKLK_OUT pins. VCC_CKOUT pins have the same specifications as VCCIO and should be connected to a VCCIO supply even if the external clock is not used.
EP20K300E EP20K400E EP20K600E EP20K1000E EP20K1500E	VCC_CKLK[4..1] GND_CKLK[4..1]	Power and ground pins for the ClockLock and ClockBoost PLL circuitry. To ensure noise resistance, the power and ground supply to the PLL should be isolated from the power and ground to the rest of the device. VCC_CKLK pins have the same specifications as VCCINT and should be connected to a 1.8 V supply even if the PLL is not used.
	VCC_CKOUT1 VCC_CKOUT2 GND_CKOUT1 GND_CKOUT2	Power and ground pins for the external clock output pins (CLKLK_OUT). These pins supply the VCCIO and GNDIO for the CLKLK_OUT pins. VCC_CKOUT pins have the same specifications as VCCIO and should be connected to a VCCIO level even if the external clock is not used.

The ClockLock circuits contain analog components, which may be sensitive to noise generated by digital components. Voltage fluctuations on the power and ground planes on the board, such as ground bounce and VCC sag, directly affects clock jitter. To avoid excessive jitter, the designer must use proper supply decoupling.

All devices with ClockLock circuitry have special VCC and GND pins, which provide power to the PLL and its dedicated output pin. The power and ground connected to these pins must be isolated from the power and ground to the rest of the APEX device or to any other digital devices. These pins are named VCC_CKLK, GND_CKLK, VCC_CKOUT and GND_CKOUT. The pin file generated by the Quartus II software and the *APEX 20K Device Family Data Sheet* pin tables identify these pins.

One of following recommended board layout techniques can be used:

- Use separate VCC_CKLK, GND_CKLK, VCC_CKOUT and GND_CKOUT power planes in board layout.

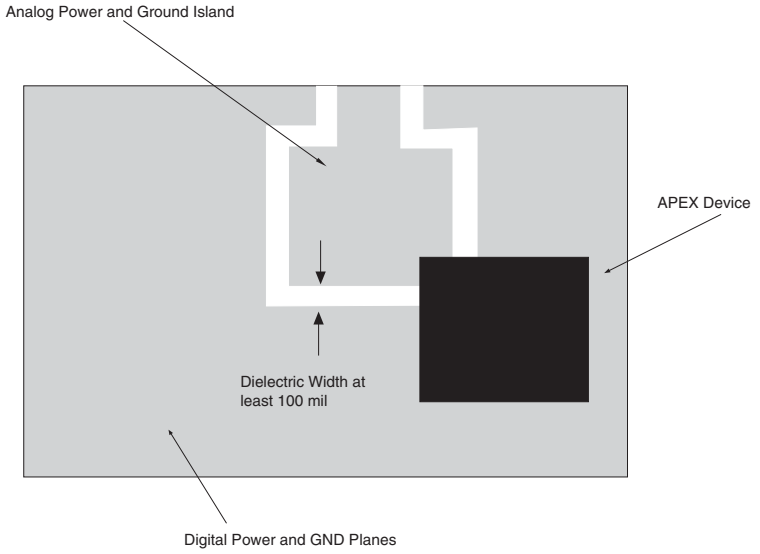
The designer of a mixed-signal system will have already partitioned the system into analog and digital sections, each with its own power and ground planes on the board. In this case, the VCC_CKLK and GND_CKLK pins can be connected to the analog power and ground planes. VCC_CKOUT requires a digital power plane connection.

- Partition the VCCINT, GNDINT, VCCIO, and GNDIO planes to include islands for VCC_CKLK, GND_CKLK, VCC_CKOUT, and GND_CKOUT respectively.

For fully digital systems that do not already have separate analog power and ground planes on the board, adding two or four new planes to the board may be prohibitively expensive. Instead, the board designer can create islands for the VCC_CKLK/GND_CKLK and VCC_CKOUT/GND_CKOUT. Figure 9 shows an example board layout with an analog power island. This would need to be done for VCC_CKLK/GND_CKLK and VCC_CKOUT/GND_CKOUT.

If the CLKLK_OUT pins are not used, isolation for VCC_CKOUT is not necessary.

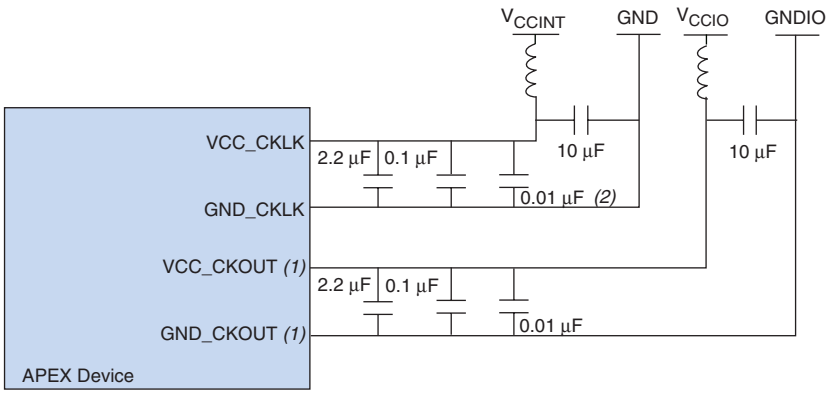
Figure 9. Board Layout for PLL V_{CC} & GND Islands



Each of the VCC_CKLK/GND_CKLK and VCC_CKOUT/GND_CKOUT pairs should be decoupled with a 2.2-µF, 0.1-µF and 0.01-µF parallel combination of ceramic capacitors located as close as possible to the APEX 20K or APEX 20KE device. Place a 10-µF tantalum capacitor immediately adjacent to the location where the power-supply lines for the PLLs come into the PCB along with a ferrite bead. Values depend on the frequency of the application. Refer to the ferrite bead manufacturer frequency specifications. See Figure 10. If PLLs are used in LVDS mode, the VCC_CKLK4/GND_CKLK4 and VCC_CKLK3/GND_CKLK3 pins should be decoupled with 2.2-µF, 0.1-µF, 0.01-µF, and 0.001-µF parallel capacitors.

Figure 10 shows the general purpose PLL power supply decoupling.

Figure 10. General Purpose PLL Power Supply Decoupling



Notes:
(1) The VCC_CKOUT and GND_CKOUT pins are only available on APEX 20KE devices.
(2) If PLLs are used in LVDS mode, the VCC_CKLK4/GND_CKLK4 and VCC_CKLK3/GND_CKLK3 pins should be decoupled with an additional 0.001-µF parallel capacitor.

Software Support

The ClockShift and ClockBoost features, as well as other PLL feature settings, are controlled by the altclklock parameters. This section describes the ports and parameters for the altclklock megafunction and shows the function prototype and component declarations.

altclklock Ports & Parameters

Tables 12 through 14 list the altclklock input port, output port, and parameter descriptions.

Table 12. altclklock Input Port Descriptions

Name	Required	Description	Comments
inclock	Yes	Clock port that drives the ClockLock PLL	
inclocken	No	PLL enable signal	When the inclocken port is high, the PLL drives the clock0 and clock1 ports. When the inclocken port is low, the clock0 and clock1 ports drive out GND and the PLL goes out of lock. When the inclocken port goes high again, the PLL must relock. This port is not available for APEX 20K devices.
fbin	No	External feedback input for PLL	To complete the feedback loop, there must be a board-level connection between the fbin pin and the PLL's external clock output pin. This port is not available for APEX 20K devices.

Table 13. altclklock Output Port Descriptions

Name	Required	Description	Comments
clock0	No	First output clock of the PLL	In APEX 20K devices, if the pin driving the PLL's inclock port is used elsewhere in the design, you can use only the PLL's clock0 output port. No fit is possible if you simultaneously use the clock0 port, the clock1 port, and the pin driving the PLL's inclock port. In APEX 20KE devices, you can use the clock0 port, the clock1 port, and the pin driving the PLL's inclock port. If, however, you are using the PLL to generate only one clock signal, you should use the clock1 port to give the Compiler added flexibility when fitting the PLL.
clock1	No	Second output clock of the PLL	In APEX 20K devices, if the pin driving the PLL's inclock port is used elsewhere in the design, you can use only the PLL's clock0 output port. No fit is possible if you simultaneously use the clock0 port, the clock1 port, and the pin driving the PLL's inclock port. In APEX 20KE devices, you can use the clock0 port, the clock1 port, and the pin driving the PLL's inclock port. If, however, you are using the PLL to generate only one clock signal, you should use the clock1 port to give the Compiler added flexibility when fitting the PLL.
locked	No	Status of the PLL	When the PLL is locked, this signal is VCC. When the PLL is out of lock, this signal is GND. The locked port may pulse high and low while the PLL is in the process of achieving lock.

Table 14. altclklock Parameter Descriptions (Part 1 of 6)

Name	Type	Required	Description
INCLOCK_PERIOD	Integer	Yes	Specifies the period of the inclock port in ps. This parameter is not required if a clock setting is specified for the inclock port.
INCLOCK_SETTINGS	String	No	Specifies the clock setting assignment to be used with the inclock port. If the INCLOCK_SETTINGS parameter is specified, the INCLOCK_PERIOD parameter is not required and is ignored. If omitted, the default is "UNUSED".
VALID_LOCK_CYCLES	Integer	No	Specifies the number of half-clock cycles for which the clock0 and clock1 ports must be locked before the locked pin goes high. This parameter is used only for third-party and functional simulations. To compute the actual number of half-clock cycles for which the clock0 and clock1 ports must be locked before the locked pin goes high, the Compiler uses the value of the VALID_LOCK_MULTIPLIER parameter. The computed value replaces any manually-specified values for the VALID_LOCK_CYCLES parameter. Altera recommends creating the PLL with the MegaWizard Plug-In Manager to obtain and select, based on the design, a close approximation of the value of the VALID_LOCK_CYCLES parameter. The MegaWizard Plug-In Manager automatically specifies values for both the VALID_LOCK_CYCLES and VALID_LOCK_MULTIPLIER parameters. If omitted, the default is five. This parameter is available only in APEX 20KE devices.

Table 14. altclklock Parameter Descriptions (Part 2 of 6)

Name	Type	Required	Description
INVALID_LOCK_CYCLES	Integer	No	Specifies the number of half-clock cycles for which the clock0 and clock1 ports must be out of lock before the locked pin goes low. This parameter is used only for third-party and functional simulations. To compute the actual number of half-clock cycles for which the clock0 and clock1 ports must be out of lock before the locked pin goes low, the Compiler uses the value of the INVALID_LOCK_MULTIPLIER parameter. The computed value replaces any manually specified values for the INVALID_LOCK_CYCLES parameter. Altera recommends creating the PLL with the MegaWizard Plug-In Manager to obtain and select, based on the design, a close approximation of the value of the INVALID_LOCK_CYCLES parameter. The MegaWizard Plug-In Manager automatically specifies values for both the INVALID_LOCK_CYCLES and INVALID_LOCK_MULTIPLIER parameters. If omitted, the default is five. This parameter is available only in APEX 20KE devices.
VALID_LOCK_MULTIPLIER	Integer	No	Specifies the scaling factor, in half-clock cycles, for which the clock0 and clock1 ports must be locked before the locked pin goes high. The Compiler uses the value of the VALID_LOCK_MULTIPLIER parameter to compute the value of the VALID_LOCK_CYCLES parameter. Altera recommends creating the PLL with the MegaWizard Plug-In Manager to obtain and select, based on the design, a close approximation of the scaling factor. The MegaWizard Plug-In Manager automatically specifies values for both the VALID_LOCK_CYCLES and VALID_LOCK_MULTIPLIER parameters. This parameter is required if the locked port is connected. Values are one and five. If omitted, the default is five. This parameter is available only in APEX 20KE devices.

Table 14. altclklock Parameter Descriptions (Part 3 of 6)

Name	Type	Required	Description
INVALID_LOCK_MULTIPLIER	Integer	No	Specifies the scaling factor, in half-clock cycles, for which the clock0 and clock1 ports must be out of lock before the locked pin goes low. The Compiler uses the value of the INVALID_LOCK_MULTIPLIER parameter to compute the value of the INVALID_LOCK_CYCLES parameter. Altera recommends creating the PLL with the MegaWizard Plug-In Manager to obtain and select, based on the design, a close approximation of the scaling factor. The MegaWizard Plug-In Manager automatically specifies values for both the INVALID_LOCK_CYCLES and INVALID_LOCK_MULTIPLIER parameters. This parameter is required if the locked port is connected. Values are one and five. If omitted, the default is five. This parameter is available only in APEX 20KE devices.

Table 14. altclklock Parameter Descriptions (Part 4 of 6)

Name	Type	Required	Description
OPERATION_MODE	String	No	In Normal mode, the phase shift is measured between the internal clock network and the dedicated inclock pin. If the PLL also feeds an external CLKLK_OUT pin, a phase difference results at the output of the external CLKLK_OUT pin due to the time delay it introduces. In Zero Delay Buffer mode, the PLL behaves as a zero-delay buffer of the input clock. The PLL must be connected to an external CLKLK_OUT pin, and the output of the external CLKLK_OUT pin is in phase with the dedicated inclock pin. If the clock0 port is used to drive the external CLKLK_OUT pin, the CLOCK0_BOOST parameter must be unused or set to one; if the clock1 port is used to drive the external CLKLK_OUT pin, the CLOCK1_BOOST parameter must be unused or set to one. If the PLL is also used to drive the internal clock network, a corresponding phase shift of that network results. The programmable phase shift feature is not available in this mode. Thus, the OUTCLOCK_PHASE_SHIFT parameter must be unused or set to zero. In External Feedback mode, the fbin port must be used, and a board-level connection between the external CLKLK_OUT pin and the CLKLK_FB pin must exist. In addition, the PLL adjusts its output to cause the signal observed at the CLKLK_FB pin to be synchronized with the input clock. If the PLL is also used to drive the internal clock network, a corresponding phase shift on that network results. Values are NORMAL, ZERO_DELAY_BUFFER, and EXTERNAL_FEEDBACK. If omitted, the default is NORMAL. This parameter is available only in APEX 20KE devices.
CLOCK0_BOOST	Integer	No	Specifies the integer multiplication factor, which must be greater than zero, for the clock0 port with respect to the input clock frequency. This parameter can be specified only if the clock0 port is used; however, it is not required if a clock setting is specified for the clock0 port. The value for this parameter must be one, two, or four for APEX 20K devices. Create the PLL with the MegaWizard Plug-In Manager to calculate the value for this parameter for APEX 20KE devices. If omitted, the default is one.

Table 14. altclklock Parameter Descriptions (Part 5 of 6)

Name	Type	Required	Description
CLOCK0_DIVIDE	Integer	No	Specifies the integer division factor, which must be greater than zero, for the clock0 port with respect to the input clock frequency. This parameter can be specified only if the clock0 port is used; however, it is not required if a clock setting is specified for the clock0 port. The setting for this parameter must be one for APEX 20K devices. Create the PLL with the MegaWizard Plug-In Manager to calculate the value for this parameter for APEX 20KE devices. If omitted, the default is one.
CLOCK0_SETTINGS	String	No	Specifies the clock setting assignment to be used with the clock0 port. If this parameter is specified, the CLOCK0_BOOST, CLOCK0_DIVIDE, and OUTCLOCK_PHASE_SHIFT parameters are not required and are ignored. If both CLOCK0_SETTINGS and CLOCK1_SETTINGS are specified, both must have the same phase shift. If omitted, the default is "UNUSED".
CLOCK1_BOOST	Integer	No	Specifies the integer multiplication factor, which must be greater than zero, for the clock1 port with respect to the input clock frequency. This parameter can only be specified if the clock1 port is used; however, it is not required if a clock setting is specified for the clock1 port. The setting for this parameter must be one, two, or four for APEX 20K devices. Create the PLL with the MegaWizard Plug-In Manager to calculate the value for this parameter for APEX 20KE devices. If omitted, the default is one.
CLOCK1_DIVIDE	Integer	No	Specifies the integer division factor, which must be greater than zero, for the clock1 port with respect to the input clock frequency. The parameter can only be specified if the clock1 port is used; however, it is not required if a clock setting is specified for the clock1 port. If omitted, the default is one.
CLOCK1_SETTINGS	String	No	Specifies the clock setting assignment to be used with the clock1 port. If this parameter is specified, the CLOCK1_BOOST, CLOCK1_DIVIDE, and OUTCLOCK_PHASE_SHIFT parameters are not required and are ignored. If both CLOCK0_SETTINGS and CLOCK1_SETTINGS are specified, they both must have the same phase shift. If omitted, the default is "UNUSED".

Table 14. altclklock Parameter Descriptions (Part 6 of 6)

Name	Type	Required	Description
OUTCLOCK_PHASE_SHIFT	Integer	No	Specifies, in ps, the output clock phase shift relative to the input clock. Phase shifts of 0.0 0.25, 0.5, or 0.75 times the input period (0°, 90°, or 270°) are implemented precisely. The allowable range for the phase shift is between 0 ps and one input clock period. If the phase shift is outside this range, the Compiler adjusts it to fall within this range. For other phase shifts, the Compiler chooses the closest allowed value. If the fbin port is used, the programmable phase shift is not available. This parameter is not required if clock settings are used for the clock0 and clock1 ports. If omitted, the default is zero. ClockShift circuitry allows you to adjust the clock delay or phase for precise timing. This parameter is available only if the OPERATION_MODE parameter is set to NORMAL. This parameter is available only in APEX 20KE devices.

Function Prototype

The following sample code shows an AHDL function prototype (port name and order also apply to Verilog HDL).

```
FUNCTION altclklock (inclock, inclocken, fbin)
   WITH (INCLOCK_PERIOD, INCLOCK_SETTINGS,
      VALID_LOCK_CYCLES, INVALID_LOCK_CYCLES,
      VALID_LOCK_MULTIPLIER,
      INVALID_LOCK_MULTIPLIER, OPERATION_MODE,
      CLOCK0_BOOST, CLOCK0_DIVIDE, CLOCK0_SETTINGS,
      CLOCK1_BOOST, CLOCK1_DIVIDE, CLOCK1_SETTINGS,
      OUTCLOCK_PHASE_SHIFT)
   RETURNS (clock0, clock1, locked);
```

VHDL Component Declaration

The following sample code shows a VHDL component declaration for the altclklock megafunction.

```
COMPONENT altclklock
   GENERIC (INCLOCK_PERIOD: NATURAL;
      INCLOCK_SETTINGS: STRING := "UNUSED";
      VALID_LOCK_CYCLES: NATURAL := 3;
      INVALID_LOCK_CYCLES: NATURAL := 3;
      VALID_LOCK_MULTIPLIER: NATURAL := 1;
      INVALID_LOCK_MULTIPLIER: NATURAL := 1;
      OPERATION_MODE: STRING := "NORMAL";
      CLOCK0_BOOST: NATURAL := 1;
      CLOCK0_DIVIDE: NATURAL := 1;
      CLOCK1_BOOST: NATURAL := 1;
      CLOCK1_DIVIDE: NATURAL := 1;
      CLOCK0_SETTINGS: STRING := "UNUSED";
      CLOCK1_SETTINGS: STRING := "UNUSED";
      OUTCLOCK_PHASE_SHIFT: NATURAL := 0);

   PORT (inclock, inclocken: IN STD_LOGIC;
         fbin : IN STD_LOGIC := '0';
         clock0, clock1, locked : OUT STD_LOGIC);
END COMPONENT;
```

MegaWizard Interface

The MegaWizard Plug-In Manager automatically sets the appropriate parameters. On the first page, the MegaWizard manager allows you to select between entering a new instance or editing an existing instantiation. On the second page of the MegaWizard Plug-In, choose altclklock from the **gates** directory. On this same page, you can choose the filename and an AHDL, VHDL, or Verilog HDL file type. The options on page three of the MegaWizard window only apply to APEX 20KE device PLLs, as shown in Figure 11. Table 15 lists the options available on page three of the altclklock MegaWizard Plug-In Manager.

Figure 11. Page 3 of the altclklock MegaWizard Plug-In Manager

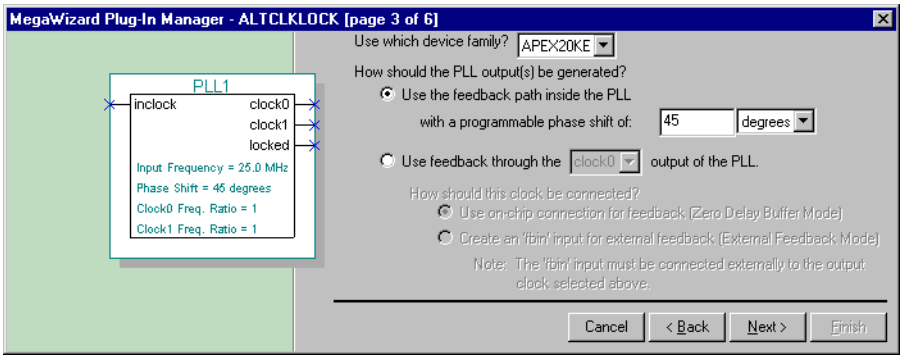
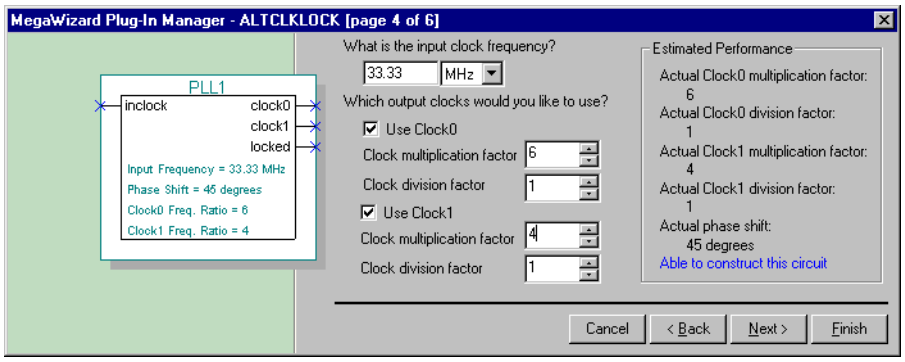


Table 15. altclklock MegaWizard Plug-In Options

Option	Description
Use the feedback path inside the PLL	This option sets the `OPERATION_MODE` parameter to `NORMAL`. In this mode, the PLL feedback path is internal to the PLL, minimizing the clock delay to the registers. This mode allows for programmable phase shift. The phase shift can be entered as degrees, ps, or ns using the drop-down list box. The smallest resolution that can be implemented is between 500 ps and 1 ns, depending on the other PLL settings. The `clock0` or `clock1` signals can be driven off-chip in this mode; however, they will not be phase aligned with the clock input pin.
Use feedback through the clock0/clock1 output of the PLL	This option sets which PLL output, `clock0` or `clock1`, will be driven off-chip in Zero Delay Buffer or External Feedback mode.
Use on-chip connection for the feedback (zero delay buffer mode)	This option sets the `OPERATION_MODE` parameter to `ZERO_DELAY_BUFFER`. In this mode, the clock port driven off-chip is phase aligned with the clock input pin for a minimized clock input to external clock output pin delay. Phase shifting is not possible in this mode. The clock output selected cannot be multiplied, but can be divided. The remaining output port that is not driven off-chip can be multiplied.
Use off-chip connection for the feedback (external feedback mode)	This option sets the `OPERATION_MODE` parameter to `EXTERNAL_FEEDBACK`. In this mode, the external feedback input pin is phase aligned with the clock input pin. The external clock output pin must feed the external feedback input pin on the board. This has the same restrictions as Zero Delay Buffer mode.

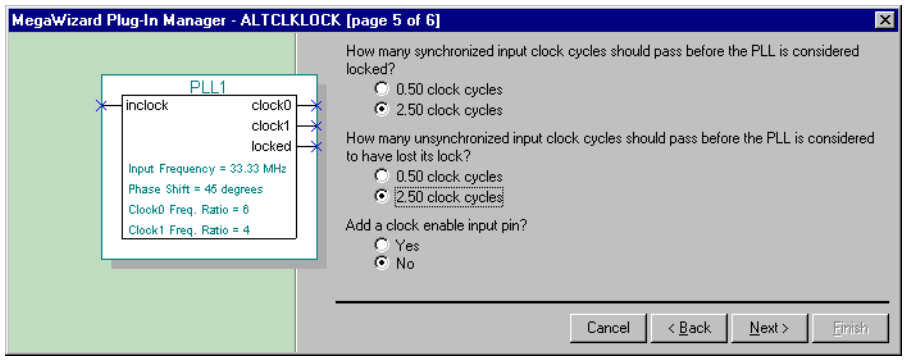
Page four of the MegaWizard Plug-In Manager (see Figure 12) sets the megafunction's input frequency, clock multiplication, and clock division. The **Estimated Performance** box displays the actual multiplication, division, and phase shift. For circuits that can be constructed, the actual multiplication and division factors may differ from the values you enter, but the ratio of multiplication/division for a given clock output will be the same. For circuits that cannot be constructed, the closest achievable multiplication and division factors are displayed. The closest possible phase shift for the estimated performance ratios is also given. The inability to achieve the desired phase shift does not prevent circuit construction; the compiler achieves the closest possible shift, shown under **Actual phase shift** in the **Estimated Performance** box.

Figure 12. Page 4 of the altclklock MegaWizard Plug-In Manager



Page five of the MegaWizard Plug-In Manager (see Figure 13) provides lock indication latency and clock enable port options. The lock indication options are determined by the internal PLL configuration parameters that are affected by the user-desired multiplication, division, and frequency. These options automatically set the `VALID_LOCK_CYCLES`, `INVALID_LOCK_CYCLE`, `VALID_LOCK_MULTIPLIER`, and `INVALID_LOCK_MULTIPLIER` parameters.

Figure 13. Page 5 of the altclklock MegaWizard Plug-In Manager



Figures 14 and 15 are examples of PLL instantiations and configurations.

Figure 14. APEX 20K & APEX 20KE altclklock Instantiation with 2× & 4× Clocks & Clock Inversion

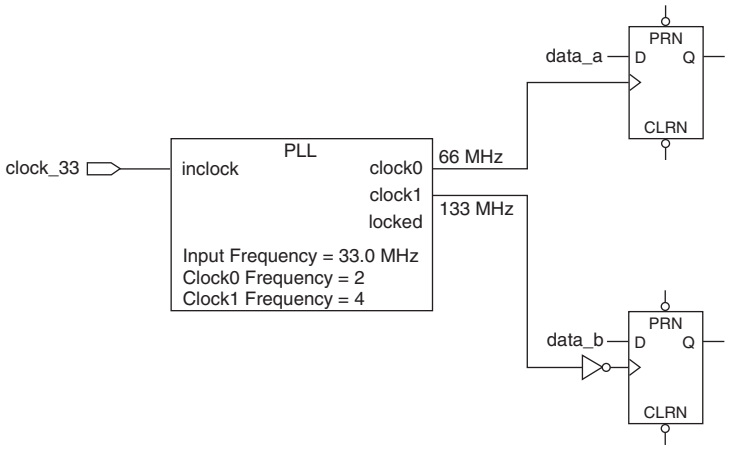
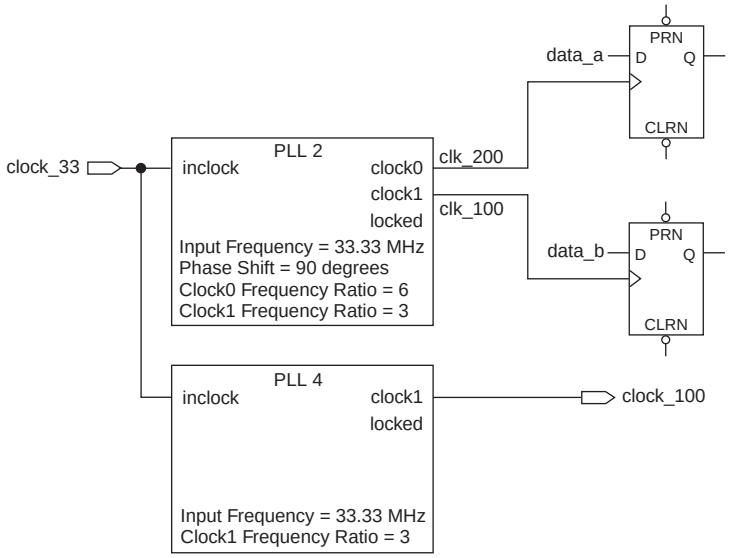


Figure 15. APEX 20KE altclklock Instantiation with Clock Multiplication, Phase Shift & External Clock Output



Reporting

The ClockLock section of the compilation report displays information regarding device PLL usage (i.e., `altclklock` megafunction usage). A compilation information message displays whether the requested `clock_boost` and `clock_divide` factors and/or the requested phase shift can be achieved. This information is useful if you do not use the MegaWizard Plug-In Manager to verify if a PLL configuration can be constructed. For unachievable `clock_boost` and `clock_divide` factors, compilation will provide an error message displaying the closest achievable factors. For unachievable phase shift, the compilation displays the closest-achievable and implemented phase shift. Actual valid or invalid lock cycle indication is also displayed.

The ClockLock section is omitted from the compilation report if the design does not include PLLs. For more information on the ClockLock section, see Quartus II software Help.

Timing Analysis

Multi-clock timing analysis causes the timing analyzer to report results using slack. The PLL input clocks and output clocks are different clocks that require multi-clock analysis. This condition is true even for the 1× case, because the clock coming out of the PLL is generated from the PLL VCO (not the clock pin), and has a reduced clock delay on the PLL output clock.

Another important fact is that the PLL is tuned to run at the frequency you specify. It will not function reliably when above or below the specified frequency (except for a 2.5% frequency tolerance). The PLL runs according to your specified settings and may not run at the maximum clock frequency ($\mathbf{f_{MAX}}$). Because of this and the multi-clock analysis, $\mathbf{f_{MAX}}$ is not reported.

If an $\mathbf{f_{MAX}}$ calculation is necessary, you can derive it from the reported slack. The micro $\mathbf{t_{CO}}$, $\mathbf{t_{SU}}$, and path delay are given for a list path command in the Slack Report window. These delays can be added and inverted to find the $\mathbf{f_{MAX}}$ of that path.

When using an external feedback input, the **External Input Delay** option can be used to specify the amount of board delay from the external clock output pin back to the external feedback input. This pin assignment can be made in the Quartus II software through the **Timing** dialog box (Tools menu – **> Assignment Organizer**).

Clock Domain Transfers

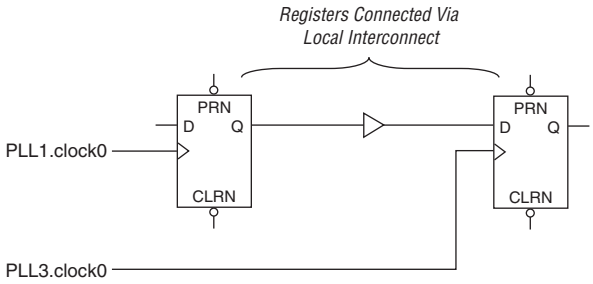
For data transfer across clock domains, specific design considerations should be made when using PLL clocks with synchronous and asynchronous transfers. The next two sections describe these considerations.

Synchronous Transfers

If the two clocks for domain transfer come from a single PLL, all synchronous register-to-register transfers (i.e. 50 MHz to 50 MHz or 50 MHz to 100 MHz) work across all conditions and no special design considerations need to be made.

If the two clocks come from two different PLLs (i.e., fed by the same clock with no ClockShift), you must insert at least one LE in the data path to guarantee data transfer between two registers that are connected via local interconnect. All other register-to-register transfers (e.g., across MegaLAB™ interconnects) work without special design considerations. Figure 16 shows the `LCELL` insertion for multiple clock source register-to-register transfer via local interconnect.

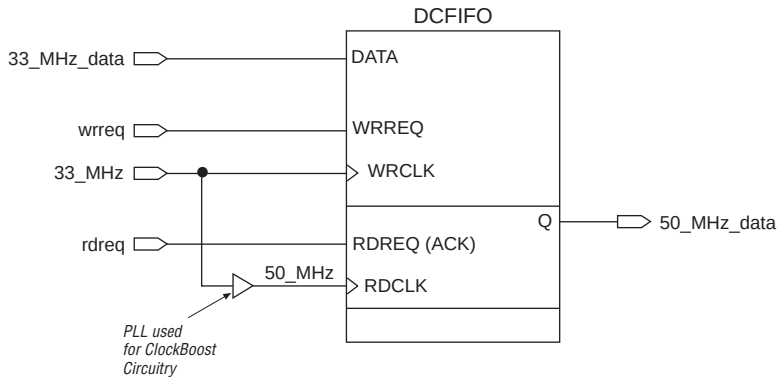
Figure 16. LCELL Insertion for Multiple PLL Clock Source Register-to-Register Transfer via Local Interconnect



Asynchronous Transfers

For asynchronous register-to-register transfer (i.e., 50 MHz to 33 MHz), use the appropriate asynchronous design techniques to transfer data from one clock domain to the other. For example, the `DCFIFO` first-in first-out (FIFO) function can be used to buffer the data transfer. Figure 17 shows a `DCFIFO` function that can be used to buffer the data transfer.

Figure 17. Using DCFIFO to Interface between Asynchronous Clock Domains



If ClockShift feature is used only on some clocks in a register-to-register transfer, the f_{MAX} may be reduced or a hold time violation may occur, depending on the direction, magnitude of the shift (any positive shift past 180 degrees can be considered negative shift), and whether the destination or source register's clock is shifted.

Simulation

The `altclklock` behavioral model can be used to simulate both the APEX 20K PLL and the APEX 20KE PLL by generating a clock signal based upon a reference clock. The APEX 20K and APEX 20KE behavioral model's instantiation should follow the same guidelines and restrictions as the design entry. The `altclklock` behavioral and timing models do not simulate jitter or lock acquisition time. At zero time, the simulation assumes lock time has already occurred. The latency on lock indication is modeled.

To simulate the External Feedback Input pin, you must set the **External Input Delay** option on the external feedback input pin, as shown in the steps below.

1. In the Quartus II software, choose **Assignment Organizer** (Tools menu).
2. Choose the **By Node** tab. In the **Mode** box, select **Edit Specific Entity & Node Settings for:**.
3. Select **Browse (...)**. Use the Node finder to search for the external feedback input pin and select **OK**.

4. In the **Assignment Organizer**, select the **Timing** assignment categories list. In the **Name** list, select **External Input Delay**. In the **Setting** box, type the amount of board fly time between the external clock output pin and external feedback input pin. The delay should not exceed 5 ns or 50% of the input clock period, whichever is less. select **OK**.

The simulator can now model the external clock output timing with external feedback.

The behavioral models for the `altclklock` megafunction reside in the **\quartus\eda\sim_lib** directory. The **apex20ke_mf.vhd** file contains the VHDL behavioral models and can be used for `altclklock` in both APEX 20K and APEX 20KE devices. The **apex20ke_mf.v** file contains the Verilog HDL behavioral models and can be used for `altclklock` in both APEX 20K and APEX 20KE devices.

The behavioral model does not perform parameter error checking, and the user must specify only valid values for the `altclklock` parameters. When targeting APEX 20K devices, only use APEX 20K-applicable parameters with appropriate values.

To simulate the model successfully, the VHDL simulator's resolution must be set to ps. A larger resolution will result in calculation rounding and thus create incorrect multiplication or division.

Sample VHDL Instantiation of the altclklock Model in a Design

The following shows a sample VHDL instantiation of the `altclklock` model in a design.

```
library ieee;
use ieee.std_logic_1164.all;
entity pll_design is
      port (inclock  : in std_logic;
            inclocken: in std_logic;
            data_in1 : in std_logic_vector(7 downto
0);
            clock0   : out std_logic;
            r_out    : out std_logic_vector(7 downto
0);
            locked   : out std_logic);
end pll_design;

architecture apex of pll_design is

component my_dff
      port (clock    : in STD_LOGIC;
            data     : in STD_LOGIC_VECTOR(7 DOWNTO
0);
            q        : out STD_LOGIC_VECTOR(7 DOWNTO
0));
end component;

component altclklock
      generic (
            inclock_period        : natural;
            inclock_settings      : string :=
"UNUSED";
            valid_lock_cycles     : natural := 5;
            invalid_lock_cycles   : natural := 5;
            valid_lock_multiplier  : natural := 5;
            invalid_lock_multiplier: natural := 5;
            operation_mode        : string :=
"NORMAL";
            clock0_boost          : natural := 1;
            clock0_divide         : natural := 1;
            clock1_boost          : natural := 1;
            clock1_divide         : natural := 1;
            clock0_settings       : string :=
"UNUSED";
            clock1_settings       : string :=
"UNUSED";
```

```
            outclock_phase_shift   : natural := 0 );

port        (inclock  : in std_logic;
            inclocken: in std_logic;
            fbin      : in std_logic := '0';
            clock0    : out std_logic;
            clock1    : out std_logic;
            locked    : out std_logic);
end component;

signal      clock1_sig : std_logic;
begin
U0: altclklock
generic map
(
            inclock_period => 40000,
            clock1_boost => 4,
            clock1_divide => 1,
            clock0_boost => 2,
            clock0_divide => 1,
            operation_mode => "NORMAL",
            valid_lock_cycles => 5,
            invalid_lock_cycles => 5,
            valid_lock_multiplier => 5,
            invalid_lock_multiplier => 5,
            outclock_phase_shift => 10000
)

      port map
            (inclock => inclock,
            inclocken => inclocken,
            clock0 => clock0,
            clock1 => clock1_sig,
            locked => locked);
process(clock1_sig)

begin
            if clock1_sig'event and clock1_sig = '1'
then
                        r_out <= data_in1;
            end if;

end process;
end apex;
```

Sample Testbench for the VHDL Design

The following shows a sample testbench for the VHDL design.

```
library ieee;
use ieee.std_logic_1164.all;

entity plltest2 is
end plltest2;

architecture behave2 of plltest2 is

       signal inclock : std_logic := '0';
       signal inclocken : std_logic;
       signal data_in1 : std_logic_vector(7 downto 0)
:= "10101010";
       signal clock0 : std_logic;
       signal locked : std_logic;
       signal r_out : std_logic_vector(7 downto 0);

       component pll_design
       port (
         inclock : in std_logic;
         inclocken : in std_logic;
         data_in1 : std_logic_vector(7 downto 0);
         clock0 : out std_logic;
         r_out : out std_logic_vector(7 downto 0);
         locked : out std_logic) ;
       end component;
begin

inclocken <= '1' after 5 ns;

U0 : pll_design port map (
         inclock => inclock,
         inclocken => inclocken,
         data_in1 => data_in1,
         clock0 => clock0,
         r_out => r_out,
         locked => locked);

process(inclock)
begin
   for i in 1 to 100 loop
      inclock <= not inclock after 20 ns;
   end loop;
end process;
```

```
end behave2;

configuration pllconfig of plltest2 is
   for behave2
      for U0: pll_design use entity
work.pll_design(apex);
      end for;
   end for;
end pllconfig;
```

Example Verilog HDL Instantiation of the altclklock Model in a Design

The following shows an example Verilog HDL instantiation of the `alkclklock` model in a design.

```
module pllsource (inclock, inclocken, data_in1,
clock0, r_out, locked);
       input inclock, inclocken;
       input [7:0] data_in1;
       output clock0, locked;
       output [7:0] r_out;

wire clock1_sig;
reg [7:0] r_out;

altclklock PLL_1

       ( .inclock(inclock), .inclocken(inclocken),
.clock0(clock0),
         .clock1(clock1_sig), .locked(locked));

defparam
              PLL_1.inclock_period = 50000,
              PLL_1.inclock_settings = "UNUSED",
              PLL_1.clock0_settings = "UNUSED",
              PLL_1.clock1_settings = "UNUSED",
              PLL_1.valid_lock_cycles = 5,
              PLL_1.invalid_lock_cycles = 5,
              PLL_1.valid_lock_multiplier = 5,
              PLL_1.invalid_lock_multiplier = 5,
              PLL_1.clock0_boost = 4,
              PLL_1.clock1_boost = 2,
              PLL_1.clock0_divide = 1,
              PLL_1.clock1_divide = 1,
              PLL_1.outclock_phase_shift = 0,
              PLL_1.operation_mode = "NORMAL";
```

```
always @(posedge clock1_sig)
begin
       r_out = data_in1;
end
```

Sample Testbench for Verilog HDL Design

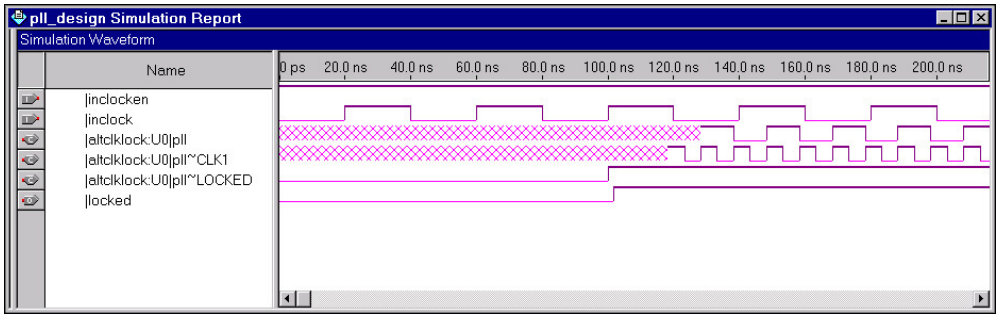
The following shows a sample testbench for a Verilog HDL design.

```
timescale 1 ns/100ps

module plltest;

parameter tmp = 8'b 10101010;
reg inclock, inclocken;
reg [7:0] data_in1;
wire clock0, locked;
wire [7:0] r_out;

pllsource U1

       ( .inclock(inclock), .inclocken(inclocken),
.data_in1(data_in1),
         .clock0(clock0), .r_out(r_out),
.locked(locked));

initial
       data_in1 = tmp;

initial
       inclock = 0;
always #25 inclock = ~inclock;

initial
       begin
          #0 inclocken = 0;
          #5 inclocken = 1;
       end

initial
       begin
          #100 data_in1 = 8'b 11110000;
          #200 data_in1 = 8'b 00110011;
       end

endmodule
```

Sample Waveform

Figure 18 shows an example waveform for dual-clock outputs of the APEX 20KE PLL. In this example, `clock0` is a 2× clock and `clock1` is a 4× clock; both are shifted/lag by 90°. For simulation, `|altclklock|`<*instance*>`|pll` is the `clock0` output of the PLL, `|altclklock|`<*instance*>`|pll~CLK1` is the `clock1` output of the PLL, and `|altclklock|`<*instance*>`|pll~LOCKED` is the locked output indication. In timing simulation, output clocks have a slight negative shift because they are at the output of the PLL and not at the flip-flop clock ports. A positive delay is added as they reach the clock ports of RAMs or flip-flops.

Figure 18. Timing Simulation Output Waveform for Dual-Output Clocks with 90° Shift



Applications

This section describes some applications of the APEX 20KE device's ClockLock, ClockBoost, and ClockShift features.

Clock Multiplication & Division

The ClockBoost feature allows designers to use low-speed clocks on the board, reducing the effects of high-speed clocks; designers can use a low-speed clock on the board and then use the ClockBoost feature to increase clock speed within the device.

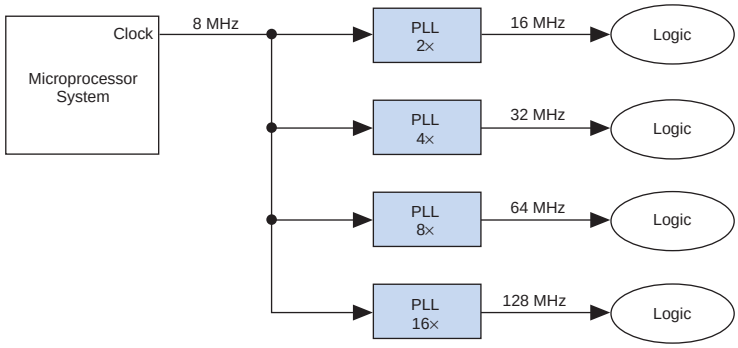
Using a lower speed clock can help reduce transmission line effects and allow the designer to simplify board layout. In APEX 20K devices, the board clock can be multiplied by 2 or 4 within the device. More complex ratios are possible with APEX 20KE devices. See Tables 12 and 13 on page 23 for more information.

Clock multiplication and division are useful for communications applications. The ClockBoost feature can be used when transfer rates must be multiplied or divided. The multiplication and division of clocks is also needed to maintain bit rates when converting between parallel data streams and serial data streams.

In microprocessor-based systems, a system clock may run at a lower rate than other system components. For example, an embedded processor or its peripheral circuits may run at a faster rate than the system I/O bus clock. Embedded applications also require faster internal rates for operations such as synchronization and counting. The ClockBoost feature can be used to multiply a slower system bus clock for an embedded application in an APEX 20K device. The multiplication and division capabilities of an APEX 20K device give designers the ability to develop SOPC

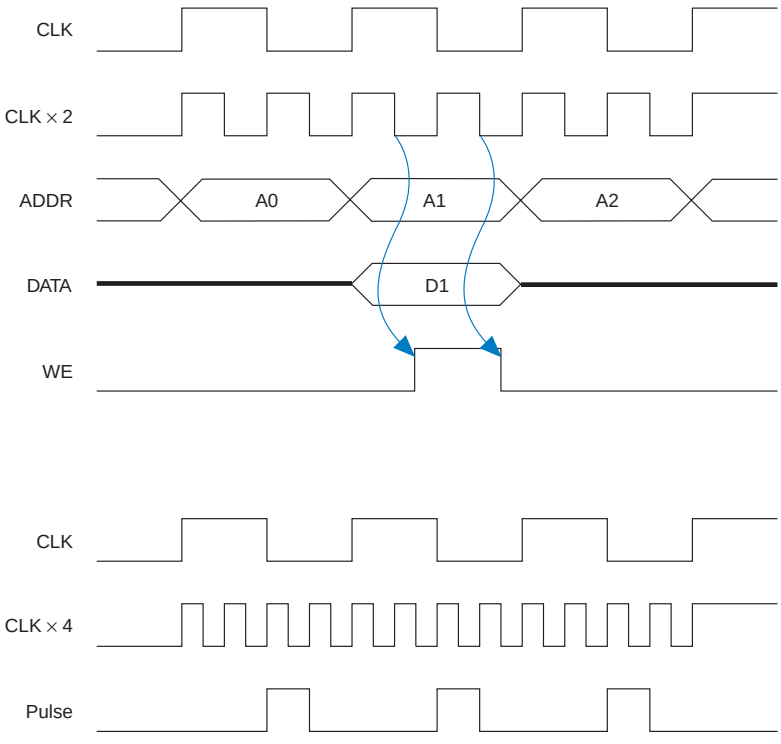
designs. Figure 19 shows clock synthesis in an embedded application.

Figure 19. Embedded Application Using Clock Synthesis



The ClockBoost feature can be used to create variable-size pulse widths. By using a multiplied frequency and a counter, you can create pulse widths of various sizes depending on the multiplied frequency driving the counter. These pulses can be used to interface with external SRAM or DRAM. For example, write enable (`WE`), row address strobe (`RAS`), and column address strobe (`CAS`) signals can be generated for a DRAM interface, meeting appropriate address and data setup times. See Figure 20.

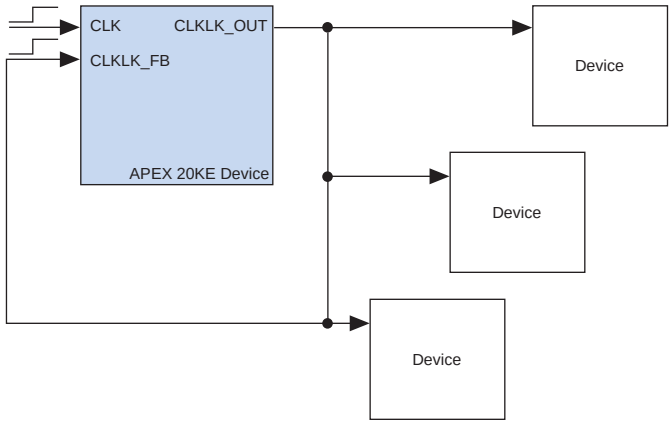
Figure 20. Using Multiplication to Generate Pulses



Removing Board Delay

APEX 20KE device feedback pins allow the designer to reduce the clock skew between several devices on a board. The PLL actively aligns the feedback input to the CLK input clock. The PLL dynamically adjusts the output during operation to account for delay changes that occur due to temperature or voltage. While designing the board, you should match the return delay containing the feedback input with the delay to each device involved. Similar delays ensure that the aligned feedback input edge is also aligned at the destination devices, eliminating delay. Figure 21 illustrates how board delay is reduced using an APEX 20KE device.

Figure 21. Reducing Board Delay Using an APEX 20KE Device *Note (1)*



Note:
(1) For board design, the route delay from CLKLK_OUT1 to each device and the return route delay to CLKLK_FB1 should be equal.

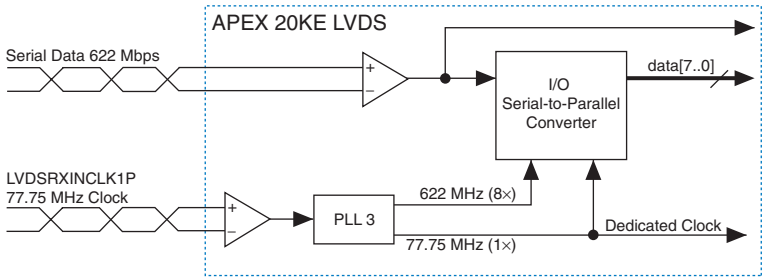
Minimize the delays between the CLKLK_OUT and CLKLK_FB signals in APEX 20KE devices. Be sure the board fly time is less than 5 ns or 50% of the input clock period, whichever is less.

LVDS

In EP20K400E and larger devices, two of the general-purpose PLLs can be configured for use in LVDS interfaces. These PLLs interface with the APEX 20KE LVDS differential input and output blocks. You can multiply the clock input by 4, 7, or 8 for LVDS/CMOS data conversion using dedicated, built-in parallel-to-serial and serial-to-parallel converters.

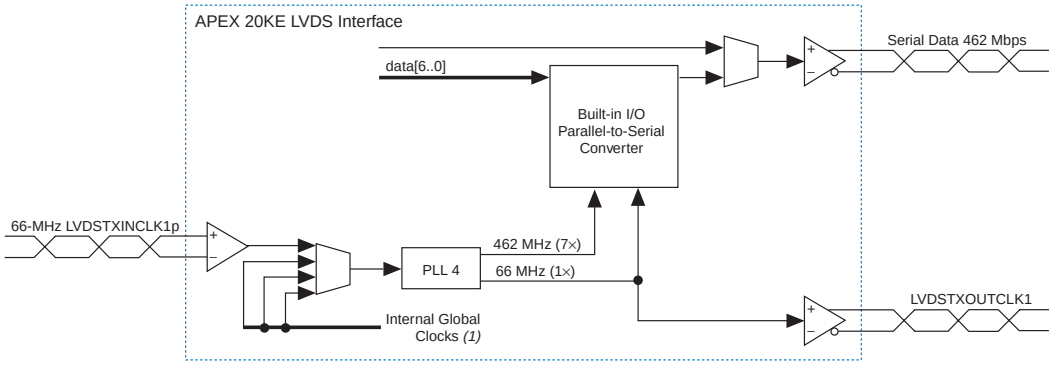
When the APEX 20KE device is configured to use LVDS, it uses PLL 3 to multiply the LVDSRXINCLK1p input. Serial-to-parallel conversion circuitry uses the multiplied clock to convert the high-speed serial LVDS data to low-speed parallel CMOS data. The multiplication factor used should match the multiplexer/de-multiplexer ratio desired. For example, if a conversion of 1-to-8 is required for a 622-Mbps LVDS channel, the multiplication factor needed is 8 with a clock input of 77.75 MHz. If needed, the serial-to-parallel converter and PLL can be bypassed for low-speed LVDS data inputs. Figure 22 shows the built-in LVDS input interface with the LVDS serial input at 622 Mbps and the multiplexer/de-multiplexer ratio at 1-to-8.

Figure 22. APEX 20KE LVDS Receiver Interface



When the APEX 20KE device is configured to use LVDS, it uses PLL 4 to multiply the LVDSTXINCCLK1p input. Parallel-to-serial conversion circuitry uses the multiplied clock to convert the low-speed parallel CMOS data to high-speed serial LVDS output data. The multiplication factor should match the multiplexer/de-multiplexer ratio desired. For example, if a conversion of 7-to-1 is needed for a 462-Mbps LVDS output, the multiplication factor needed is seven with an input clock of 66 MHz. If needed, the parallel-to-serial converter and PLL can be bypassed for a low-speed LVDS serial output. Figure 23 shows the built-in LVDS output interface that converts internal parallel data into LVDS serial data with a 7-to-1 ratio at 462 Mbps.

Figure 23. APEX 20KE Transmitter Interface



Note:

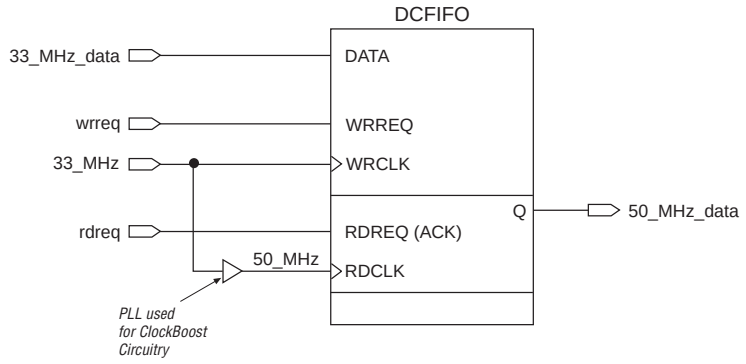
(1) In LVDS mode, PLL 4 is fed by LVDSTXINCLK1p or one of the three remaining internal global clocks (G1, G2, G3). These remaining global clocks cannot be used if fed by a general-purpose PLL (receiver PLL is allowed).

T1 & E1 Clock Domain Conversion

In APEX 20KE devices, the ClockBoost circuitry can be used to convert a T1 clock frequency (1.544 MHz) to an E1 clock frequency (2.048 MHz) and vice versa. The ClockLock circuit has a special mode to perform T1/E1 conversions; this multiplication is accomplished by setting the CLOCK0_BOOST and CLOCK0_DIVIDE parameters to 256/193 or 193/256.

As with any type of clock domain data transfer, use the appropriate asynchronous design techniques to transfer data from one clock domain to the other. For example, the DCFIFO FIFO function can be used to buffer the data transfer. Figure 24 shows a DCFIFO that interfaces between clock domains. For example, a DCFIFO is fed with input data clocked by the PLL input T1 clock. The output of the DCFIFO should be clocked with the converted E1 clock from the PLL output. Output data is synchronized to the PLL output E1 clock. This same practice is used for synchronization across other clock domains.

Figure 24. Using DCFIFO to Interface between Clock Domains

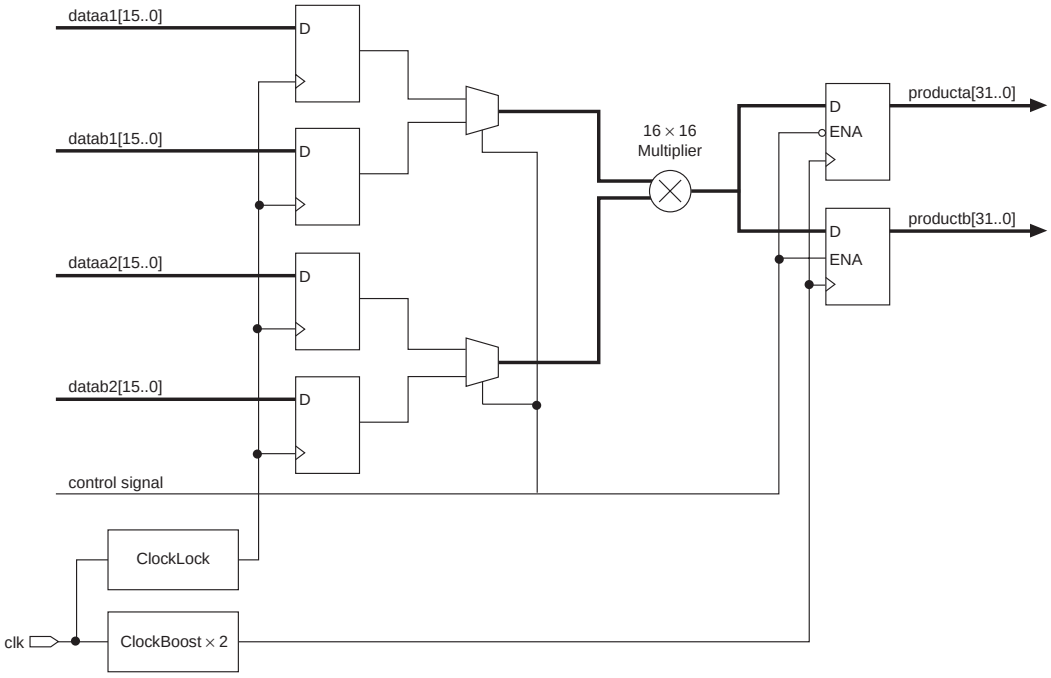


Time-Domain Multiplexing

The ClockBoost feature allows designers to implement time-domain multiplexed applications in which a given circuit is used more than once per clock cycle. Depending on whether the circuit is clocked by the 2× or 4× ClockBoost circuitry in the APEX 20K device, it can operate two or four times, respectively, per system cycle. With time-domain multiplexing, a given function can be implemented with fewer LEs or ESBs.

For example, in a circuit using two 16 × 16 multipliers, each multiplier uses 447 LEs for a total of 894 LEs. Alternatively, you could implement the circuit as one multiplier that is used twice per clock cycle by using a clock that is 2× the system clock. The input of the multiplier is multiplexed so it can switch between two sets of inputs; the output is de-multiplexed so that it can drive out the two multiplication results. While some LEs are needed to accomplish the multiplexing, the cost is outweighed by the LEs saved by using one multiplier. Figure 25 shows a schematic of the time-domain multiplexed circuit.

Figure 25. Time-Domain Multiplexed Circuit



The same example can be applied to a circuit requiring four multipliers; the circuit would use a 4× clock and 4-to-1 multiplexers instead of the 2-to-1 multiplexers shown in Figure 25. A control line bus can be created using a one-hot counter or state machine to enable one output bus register per 4× clock cycle, permitting one multiplier to be used four times in a single system clock cycle. Table 16 shows the reduction in resource requirements.

Table 16. Resources Required for 16 × 16 Multipliers

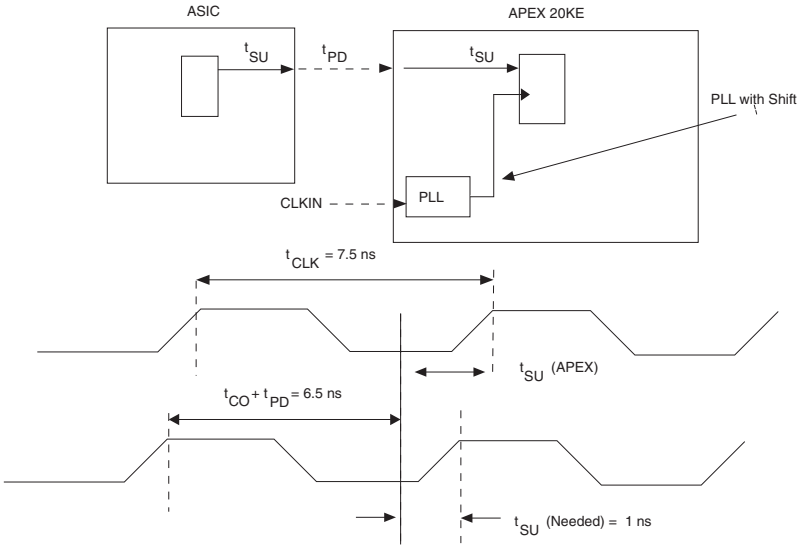
Design	LEs Required
Two 16 × 16 multipliers	894
Two 16 × 16 multipliers, time-domain multiplexed with 2× ClockBoost	547
Four 16 × 16 multipliers	1,788
Four 16 × 16 multipliers, time-domain multiplexed with 4× ClockBoost	741

ClockShift Applications

Phase and time delay adjustment of PLL clock outputs have many interface applications. Delay adjustment allows the designer to overcome strict timing margins that would not be possible to overcome without clock adjustment. By adjusting the clock output's lead or lag, you can use a known clock delay between various clocks to improve the perceived clock-to-output timing for the PLD or external devices.

Some high-speed devices, such as SDRAMs, have access times that require fast setup times on the interface device for a given critical path. To meet device-to-device timing requirements, the internal clock to the destination chip's input register can be adjusted to lag a specified amount of time from the input clock. By adding clock lag, you can obtain a faster setup time on the destination device input register. Figure 26 shows an APEX 20KE and ASIC device-to-device interface with timing. For example, the APEX 20KE device may be receiving data from an ASIC with a 5.5 ns t_{CO}. Assuming a 133-MHz system speed and a board propagation delay of 1 ns (t_{PD}) between the ASIC and APEX 20KE device, the $t_{CO} + t_{DELAY} = 6.5$ ns. Only 1.0 ns is left for setup time into the APEX 20KE device (7.5 ns period). Timing can be met by adjusting the APEX 20KE internal clock to lag by an amount of t_{SU} (PLD) – t_{SU} (needed).

Figure 26. APEX 20KE and ASIC Device-to-Device Interface with Timing



In cases where feedback is not used, clock delay control can be used to adjust clock delay to other devices based on their distance from the clock source. Designers can manually adjust the external clock output of the APEX 20KE device to compensate for board delay.

Phase adjustment is also useful for interfaces to an external device. An input clock can be phase shifted with two separate `altclklock` circuits and then output to the two external outputs. This along with the input could be used for a three-phase DC motor control.

Conclusion

The advanced APEX 20K ClockLock and ClockBoost features use PLLs to provide significant improvements in system performance and design versatility. The reduction in clock delay and the elimination of clock skew within the device improves design speed, and time-domain multiplexing improves area usage. The ClockBoost feature simplifies board design by running the internal logic of the device at a faster rate than the input clock frequency. The advanced APEX 20KE ClockLock and ClockBoost feature set is further enhanced with $m/(n \times k)$ multiplication, LVDS I/O interfaces, and phase adjustment for more complex clock synthesis applications.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
lit_req@altera.com

Revision History

The information contained in *AN 115: Using the ClockLock & ClockBoost PLL Features in APEX Devices* version 2.5 supersedes information published in previous versions.

Version 2.5 Changes

The following changes were made to *AN 115: Using the ClockLock & ClockBoost PLL Features in APEX Devices* version 2.5: updated Removing Board Delay section.

Version 2.4 Changes

The following changes were made to *AN 115: Using the ClockLock & ClockBoost PLL Features in APEX Devices* version 2.4:

- Updated *Note (3)* of Table 1.
- Updated Figure 15.