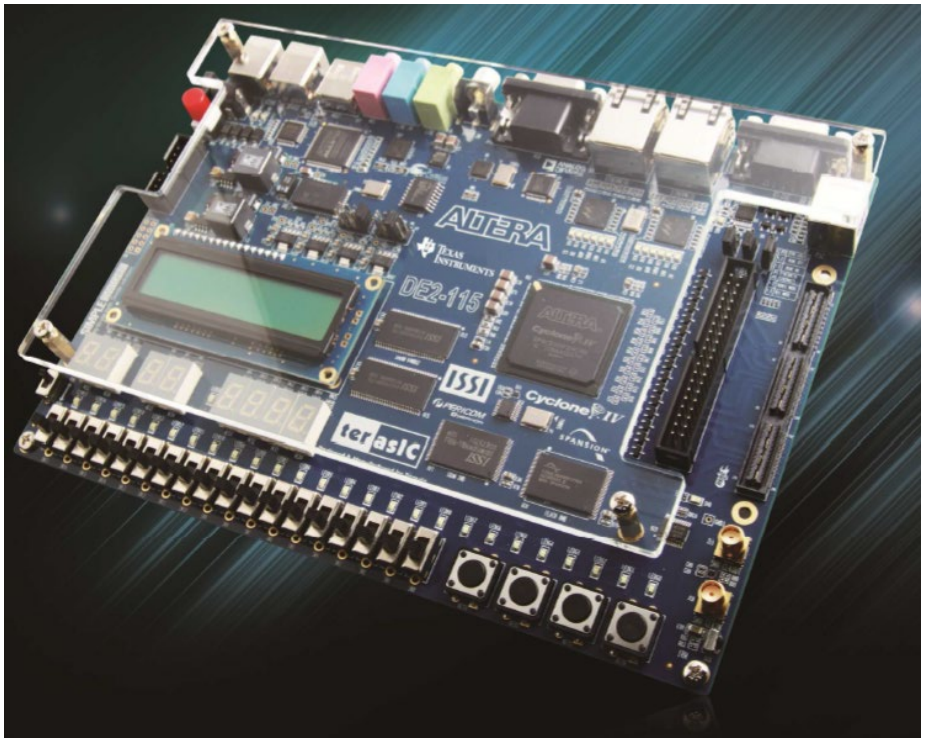
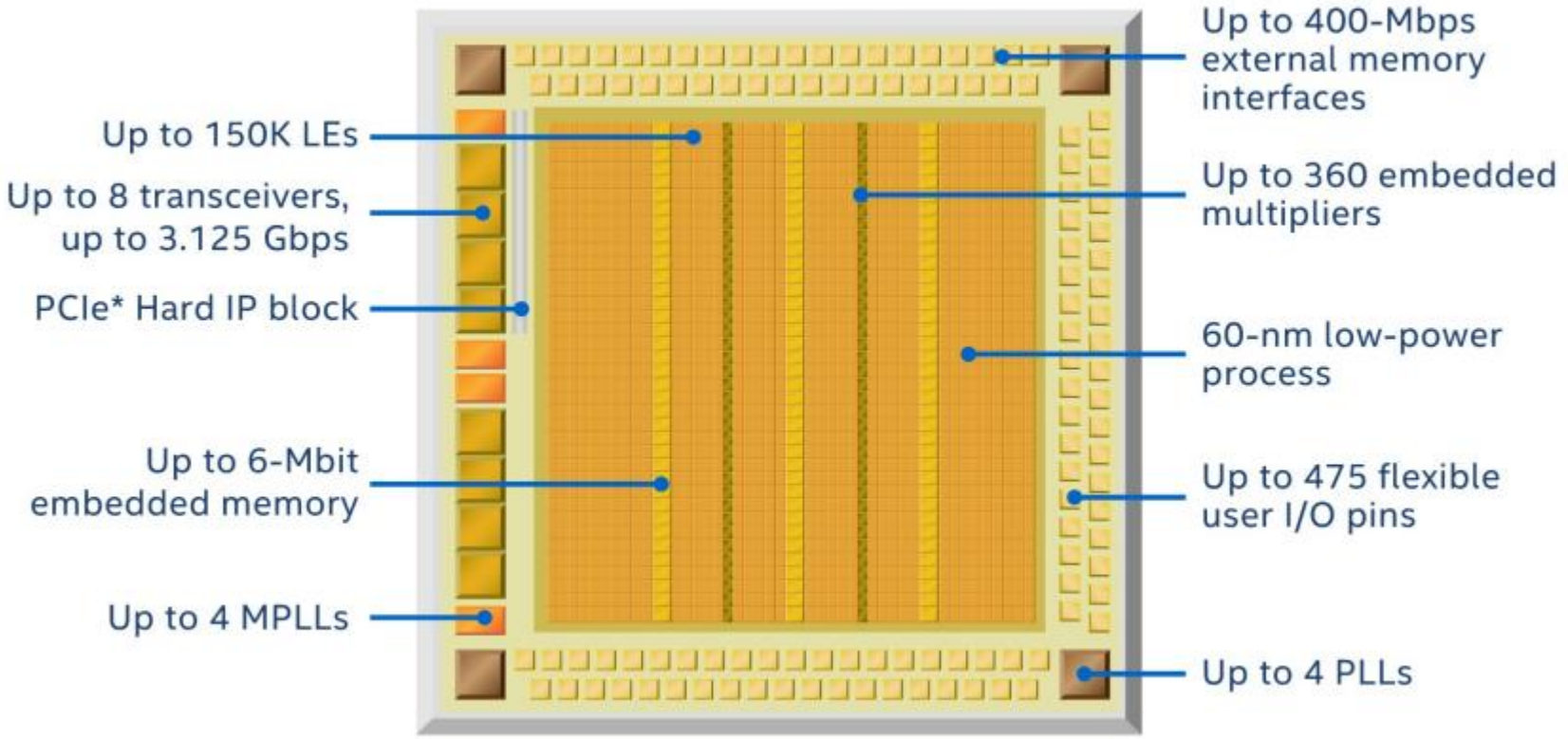
CPE 322
Digital Hardware Design Fundamentals

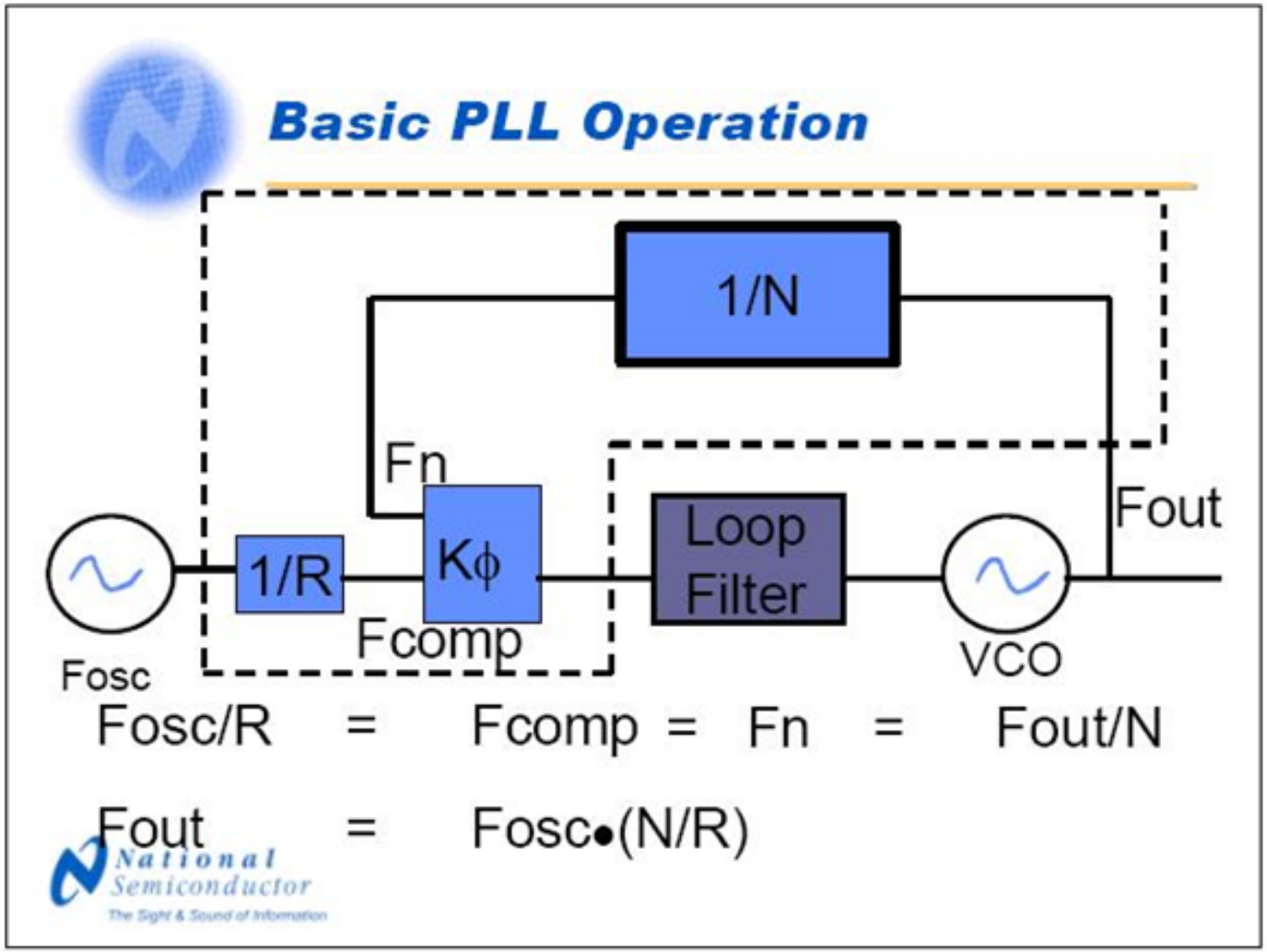
Non CLB FPGA Building Blocks:
Embedded Phase-Locked-Loops



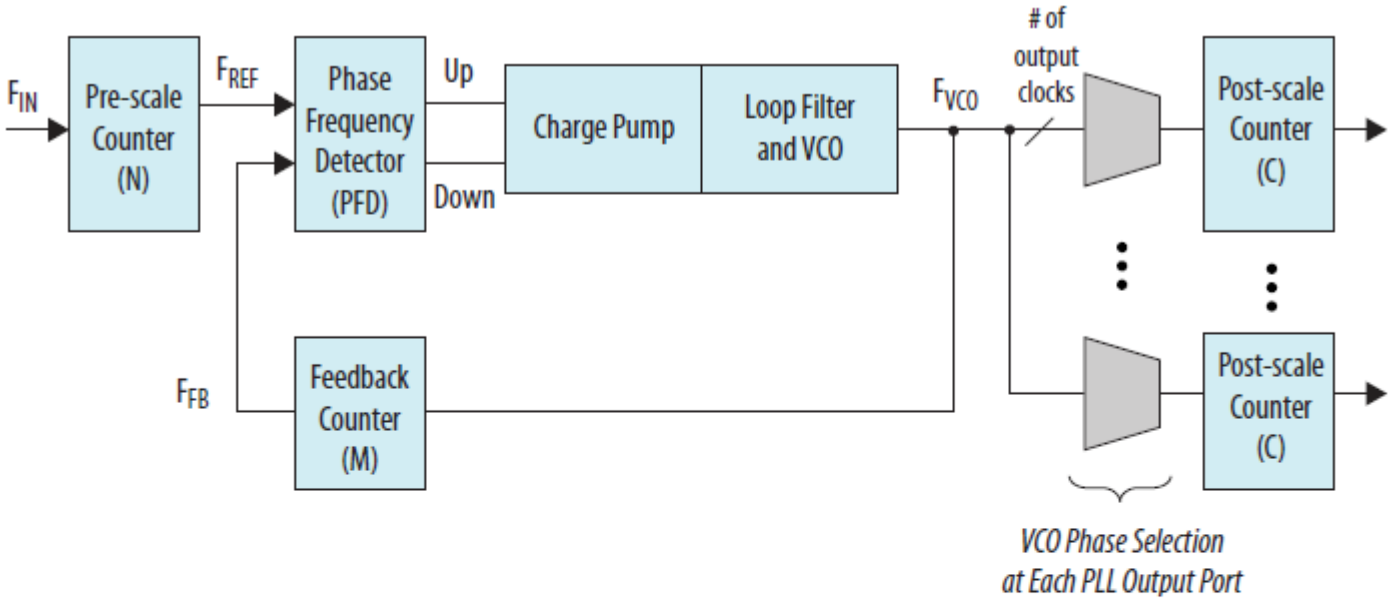
Cyclone IV E Family FPGA Layout



Basic PLL Operation



PLL in IntelFPGAs



The following terms are commonly used to describe the behavior of a PLL:

- PLL lock time—also known as the PLL acquisition time. PLL lock time is the time for the PLL to attain the target frequency and phase relationship after power-up, after a programmed output frequency change, or after a PLL reset.

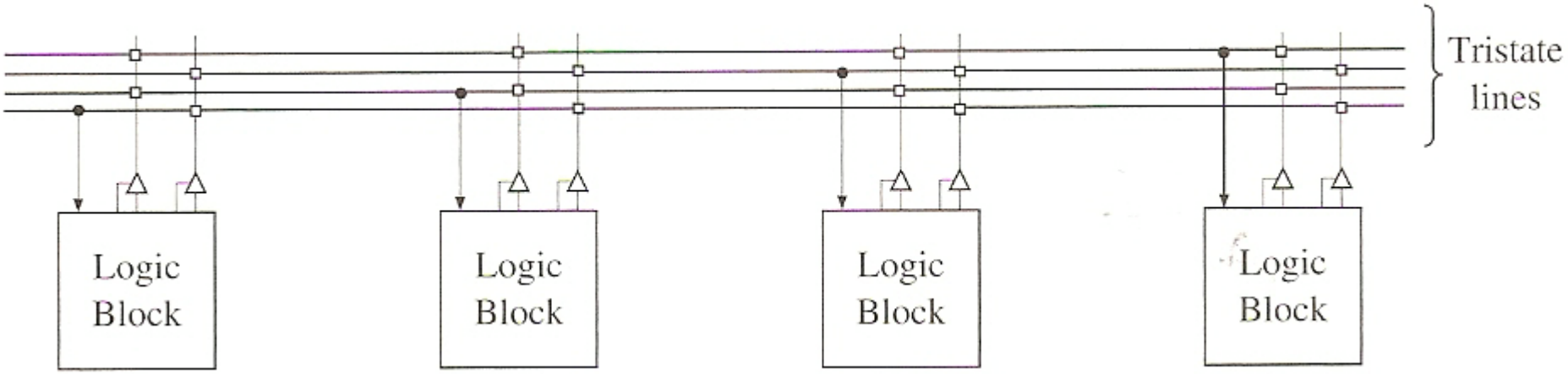
  Note: Simulation software does not model a realistic PLL lock time. Simulation shows an unrealistically fast lock time. For the actual lock time specification, refer to the device datasheet.
- PLL resolution—the minimum frequency increment value of a PLL VCO. The number of bits in the M and N counters determine the PLL resolution value.
- PLL sample rate—the F_{REF} sampling frequency required to perform the phase and frequency correction in the PLL. The PLL sample rate is f_{REF} /N.

Clock Skew and Clock Distribution

Clock Skew

There are several million gates in modern FPGA chips. When a clock is distributed to various parts of such a large chip, the delays in the wire carrying the clock can result in the clock edge arriving at different times at different parts. This difference in the actual edge of the clock as it arrives at different flip-flops or other devices is called clock skew. Clock skew is a problem in large systems, including modern microprocessors. Carefully planned clock distribution circuits are implemented in most systems in order to minimize the effect of clock skew. Modern FPGAs provide specialized clock distribution circuitry in order to create a clock of sufficient strength and low skew.

Global Lines



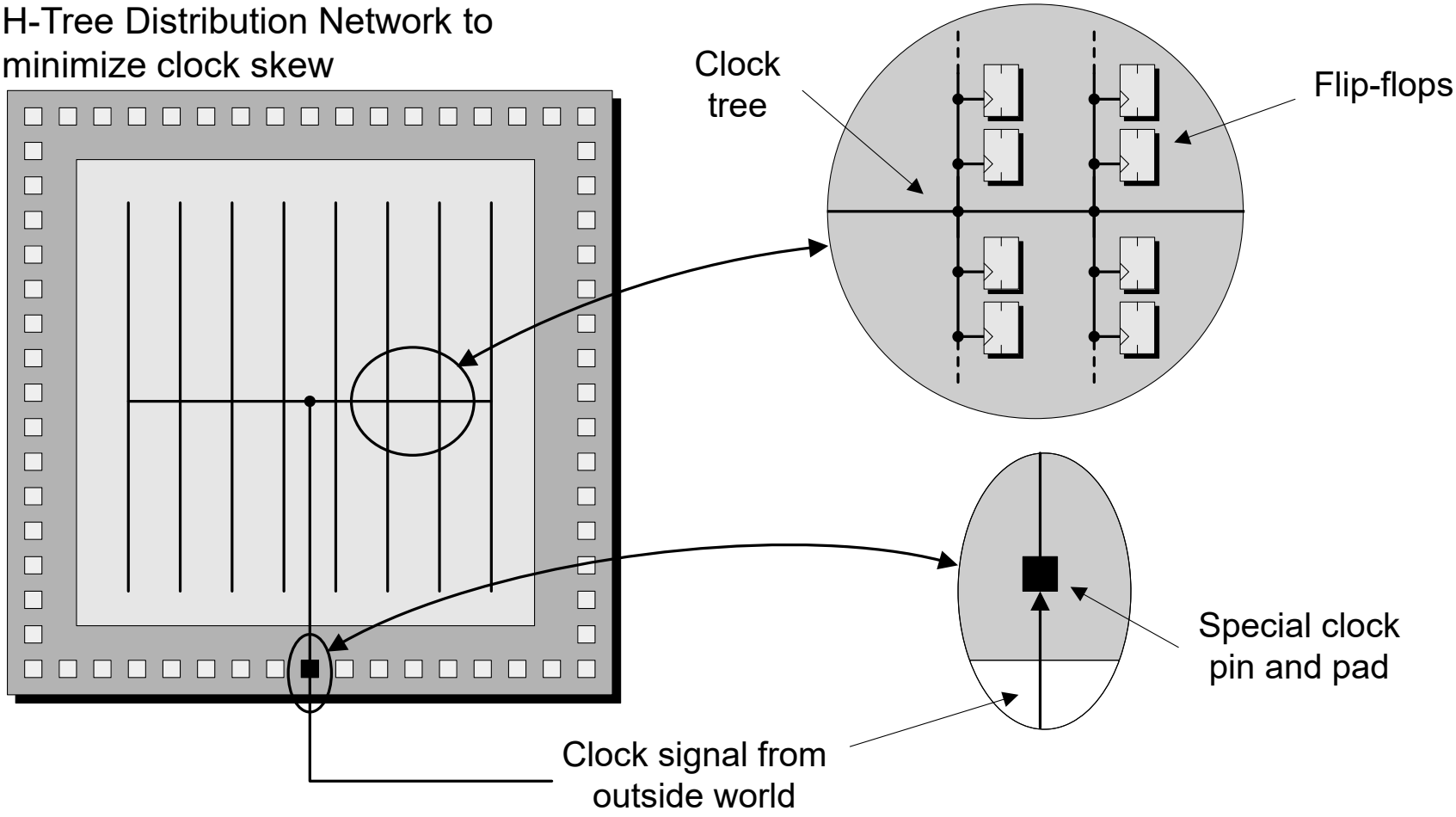
Lines that span the entire width/height of the device.
Supports High Fan-out and Low-skew clock distribution.

Internal Tri-state buffers often connect the logic blocks to these lines.

A limited number of these lines are provided.

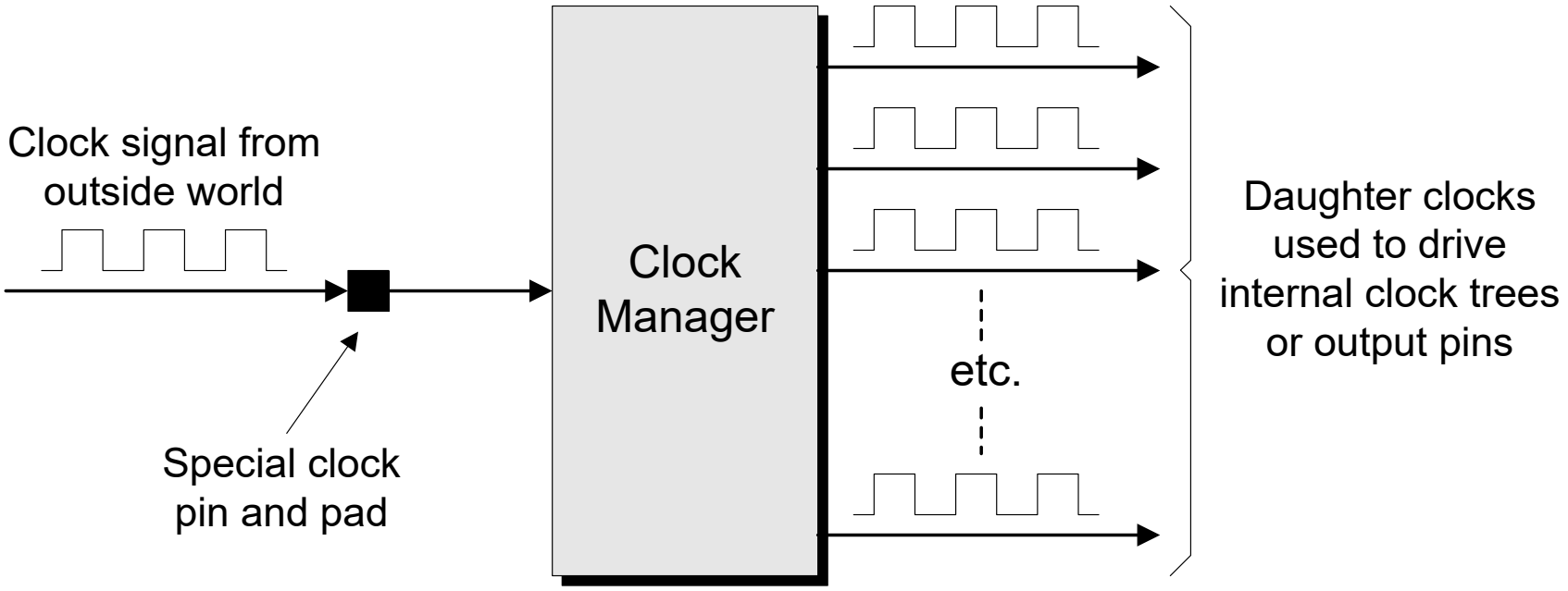
Clock Trees

- H-Tree Distribution Network to minimize clock skew



The Design Warrior's Guide to FPGAs
Devices, Tools, and Flows. ISBN 0750676043

Digital Clock Manager

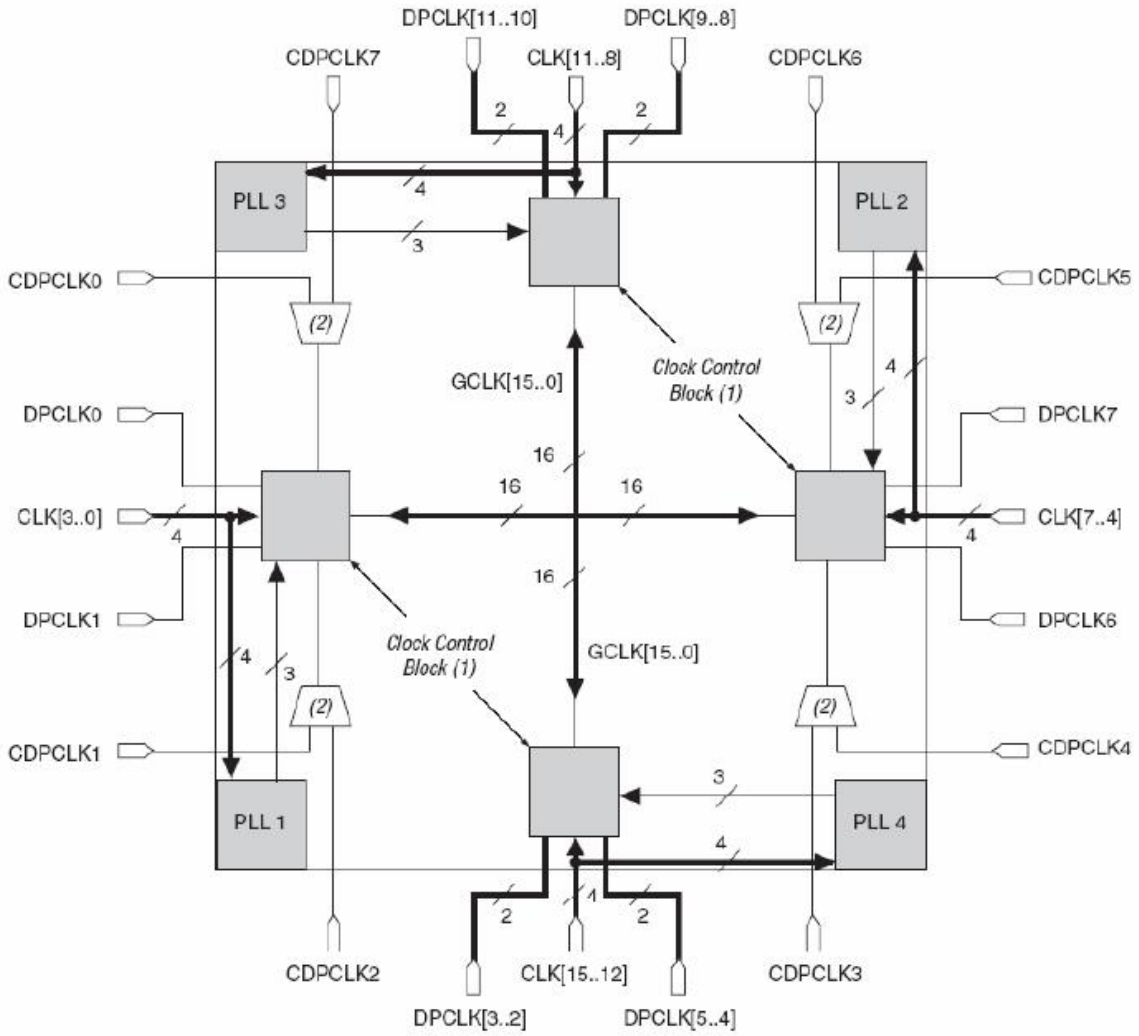


The Design Warrior's Guide to FPGAs
Devices, Tools, and Flows. ISBN 0750676043

Global Clock Networks & PLLs

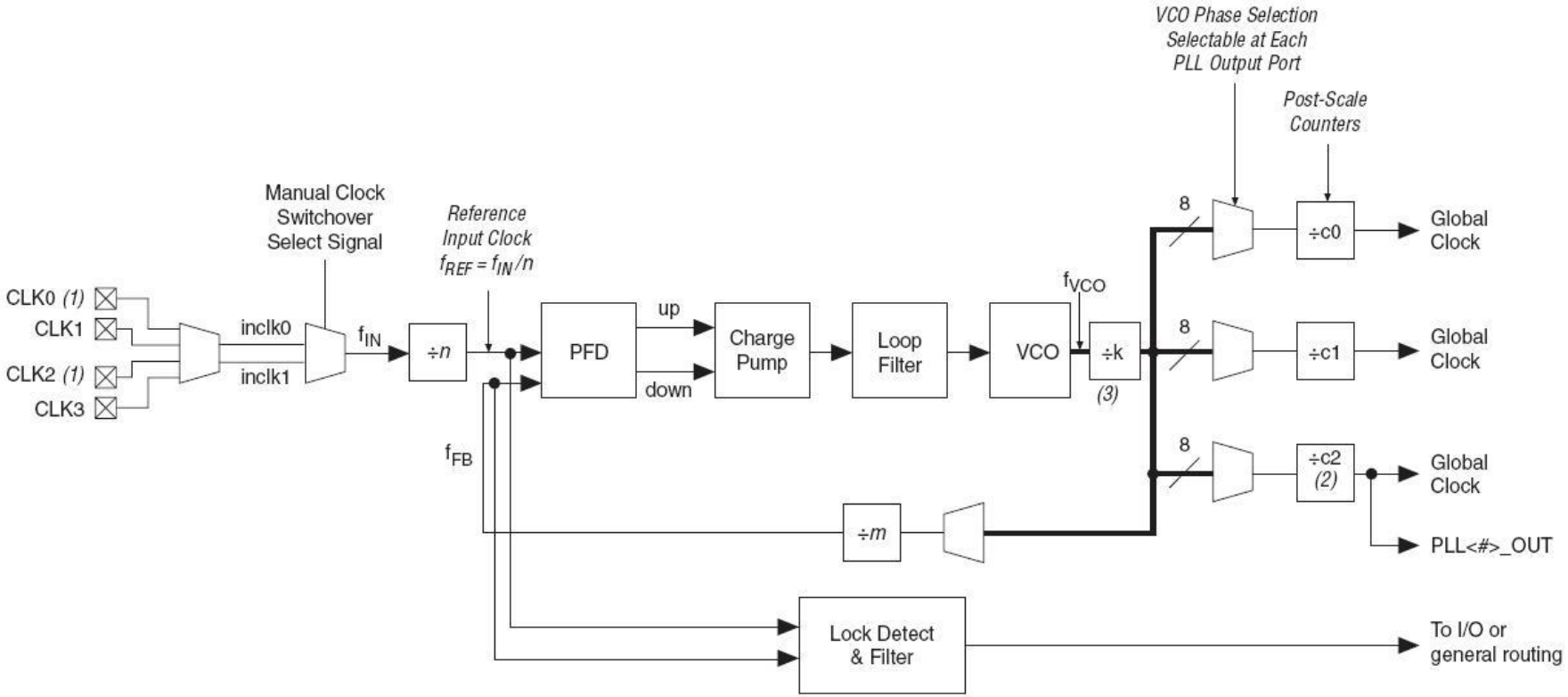
Altera Cyclone IV

Figure 2–12. EP2C20 & Larger PLL, CLK[], DPCLK[] & Clock Control Block Locations



Block Diagram Altera Cyclone II PLLs

Partitioning a Design in an FPGA



Cyclone IV PLL Features

Table 7–2. Cyclone II PLL Features

Feature	Description
Clock multiplication and division	m / (n × post-scale counter) *(1)*
Phase shift	Down to 125-ps increments *(2)*, *(3)*
Programmable duty cycle	✓
Number of internal clock outputs	Up to three per PLL *(4)*
Number of external clock outputs	One per PLL *(4)*
Locked port can feed logic array	✓
PLL clock outputs can feed logic array	✓
Manual clock switchover	✓
Gated lock	✓

Notes to Table 7–2:

(1) m and post-scale counter values range from 1 to 32. n ranges from 1 to 4.

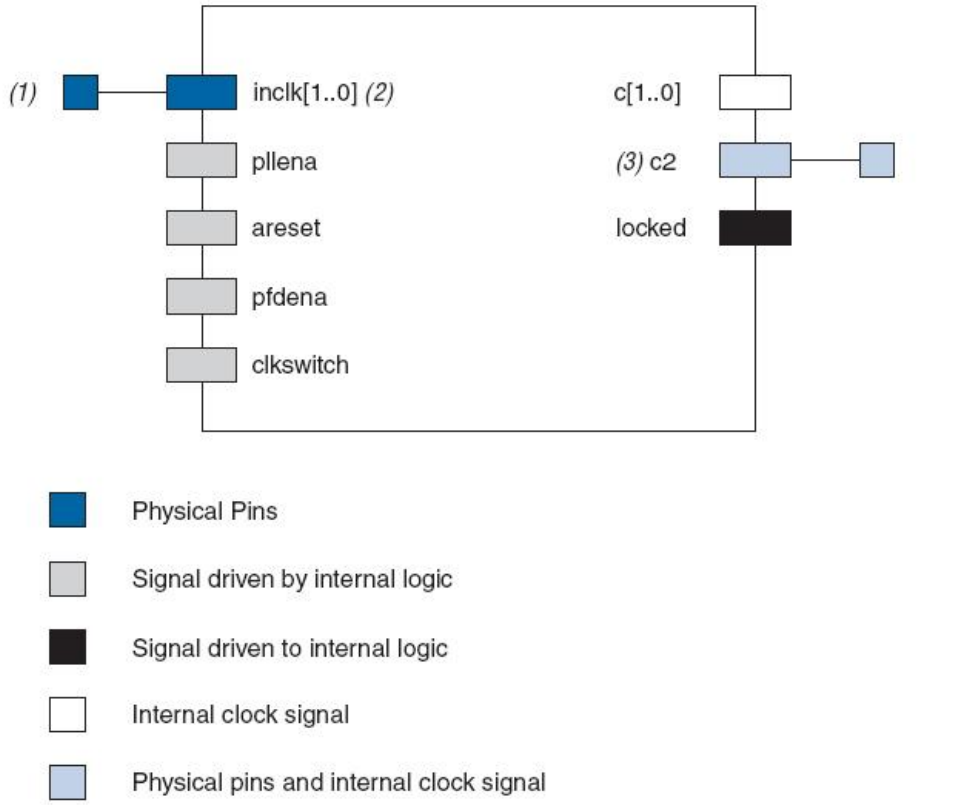
(2) The smallest phase shift is determined by the voltage control oscillator (VCO) period divided by 8.

(3) For degree increments, Cyclone II devices can shift output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the VCO frequency.

(4) The Cyclone II PLL has three output counters that drive the global clock network. One of these output counters (c2) can also drive a dedicated external I/O pin (single ended or differential). This counter output can also drive the external clock output (`PLL<#>_OUT`) and internal global clock network at the same time.

Cyclone IV PLL Signals

Figure 7–3. Cyclone II PLL Signals



Notes to Figure 7–3:

(1) These signals can be assigned to either a single-ended or differential I/O standard.
(2) The `inclk` must be driven by one of two dedicated clock input pins.
(3) This counter output can drive both a dedicated external clock output (`PLL<#>_OUT`) and the global clock network.

Cyclone IV Signals

Port Name	Type	Condition	Description
fbclk	Input	Optional	The external feedback input port for the PLL. The Altera PLL IP core creates this port when the PLL is operating in external feedback mode or zero-delay buffer mode. To complete the feedback loop, a board-level connection must connect the fbclk port and the external clock output port of the PLL.
fboutclk	Output	Optional	The port that feeds the fbclk port through the mimic circuitry. The fboutclk port is available only if the PLL is in external feedback mode.
locked	Output	Optional	The Altera PLL IP core drives this port high when the PLL acquires lock. The port remains high as long as the PLL is locked. The PLL asserts the locked port when the phases and frequencies of the reference clock and feedback clock are the same or within the lock circuit tolerance. When the difference between the two clock signals exceeds the lock circuit tolerance, the PLL loses lock.
outclk[]	Output	Required	The clock output of the PLL. The frequency of the output clock depends on the parameter settings.

Cyclone IV PLL Input Signals

Port Name	Type	Condition	Description
refclk	Input	Required	The reference clock that drives the clock network.
reset	Input	Required	The asynchronous reset port for the output clocks. Drive this port high to reset all output clocks to the initial value of 0.
zdbfbclk	Bidirectional	Optional	The bidirectional port that connects to the mimic circuitry. This port must connect to a bidirectional pin that is placed on the positive feedback dedicated output pin of the PLL. The zdbfbclk port is available only if the PLL is in zero-delay buffer mode.
refclk1	Input	Required	Second input clock signal that feeds into the PLL.

Cyclone IV PLL Signals

Port Name	Type	Condition	Description
extswitch	Input	Required	Assert this input signal high (1'b1) to manually switch the clock for at least 3 cycles.
activeclk	Output	Optional	Output signal to determine which input clock is in use by the PLL.
clkbad	Output	Optional	Output signal to determine which input clock is working.
cclk[8]	Input	Optional	c-Counter clock source from the fracturable fractional PLL output counter 4 or 13.
adjpllin	Input	Optional	Adjacent fractional PLL clock source.
cascade_out	Output	Optional	Output signal to feed into other fractional PLLs. This port acts as a bus port when the upstream PLL has two or more output clocks.

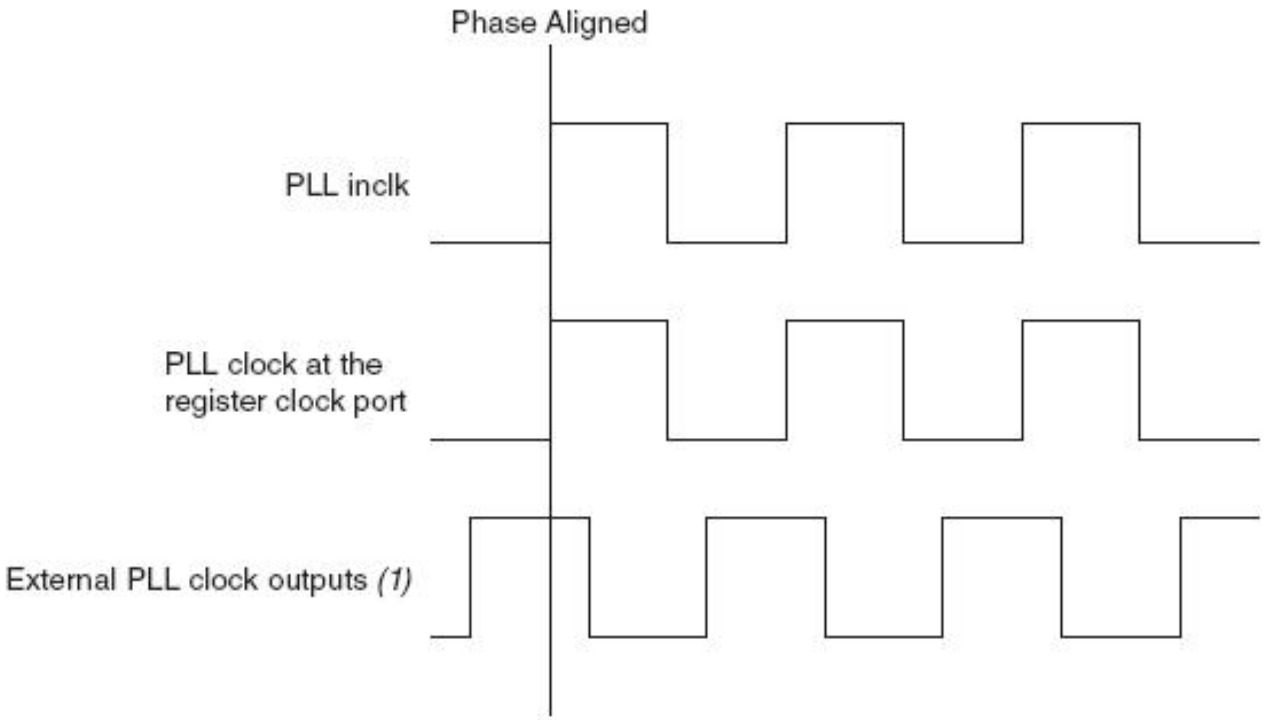
Global Clock Control Block Cyclone IV

Table 7–9. Clock Control Block Inputs

Input	Description
Dedicated clock inputs	Dedicated clock input pins can drive clocks or global signals, such as asynchronous clears, presets, or clock enables onto a given global clock network.
Dual-purpose clock (DPCLK and CDPCLK) I/O inputs	DPCLK and CDPCLK I/O pins are bidirectional dual function pins that can be used for high fan-out control signals, such as protocol signals, TRDY and IRDY signals for PCI, or DQS for DDR, via the global clock network.
PLL outputs	The PLL counter outputs can drive the global clock network.
Internal logic	The global clock network can also be driven through the logic array routing to enable internal logic (LEs) to drive a high fan-out, low skew signal path.

I/O Path Compensation in Cyclone PLLs

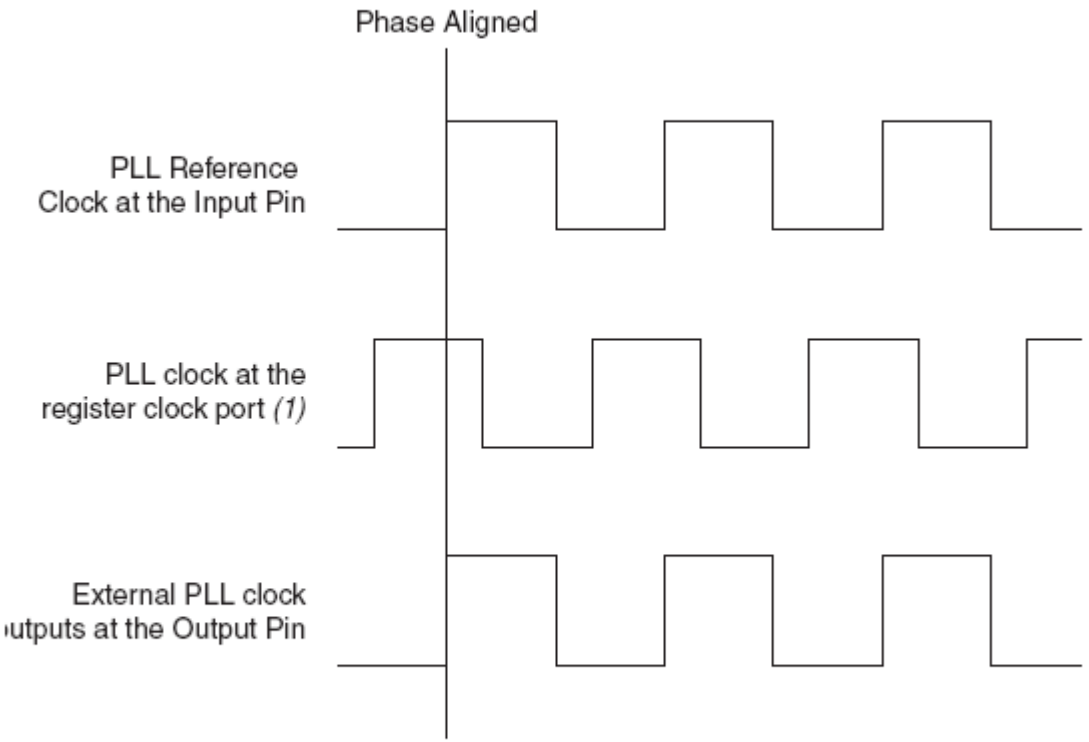
Figure 7–4. Phase Relationship between Cyclone II PLL Clocks in Normal Mode



Note
(1) The external clock output can lead or lag the PLL clock signals.

I/O Path Compensation in Cyclone II PLLs

Figure 7–5. Phase Relationship between Cyclone II PLL Clocks in Zero Delay Buffer Mode

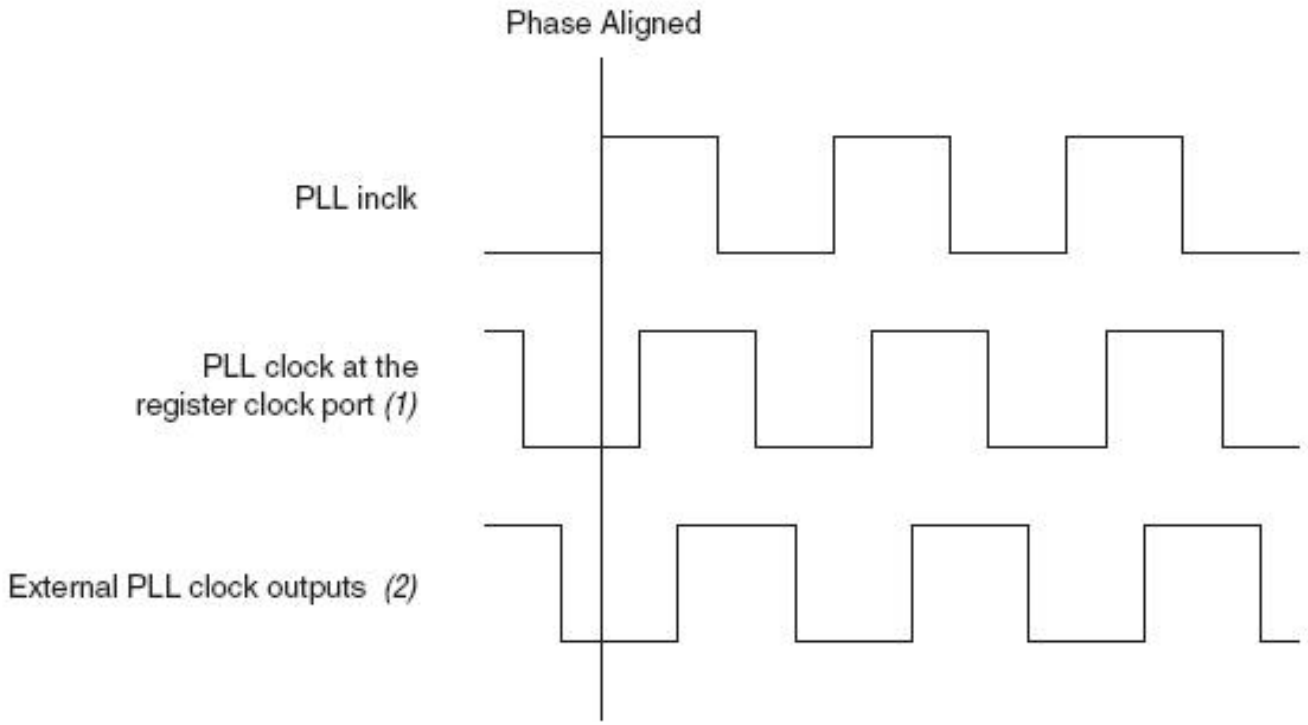


Note to Figure 7–5:

(1) The internal clock output(s) can lead or lag the external PLL clock output (PLL<#>_OUT) signals.

I/O Path Compensation in Cyclone II PLLs

Figure 7–6. Phase Relationship between Cyclone II PLL Clocks in No Compensation Mode



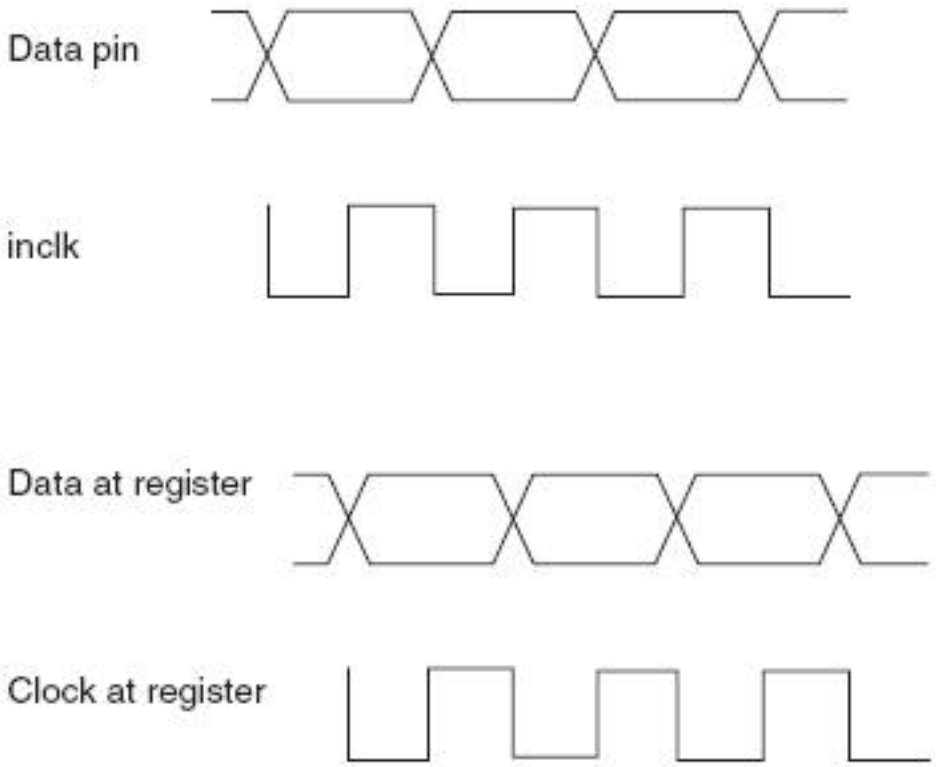
Notes to Figure 7–6:

(1) Internal clocks fed by the PLL are in phase with each other.

(2) The external clock outputs can lead or lag the PLL internal clocks.

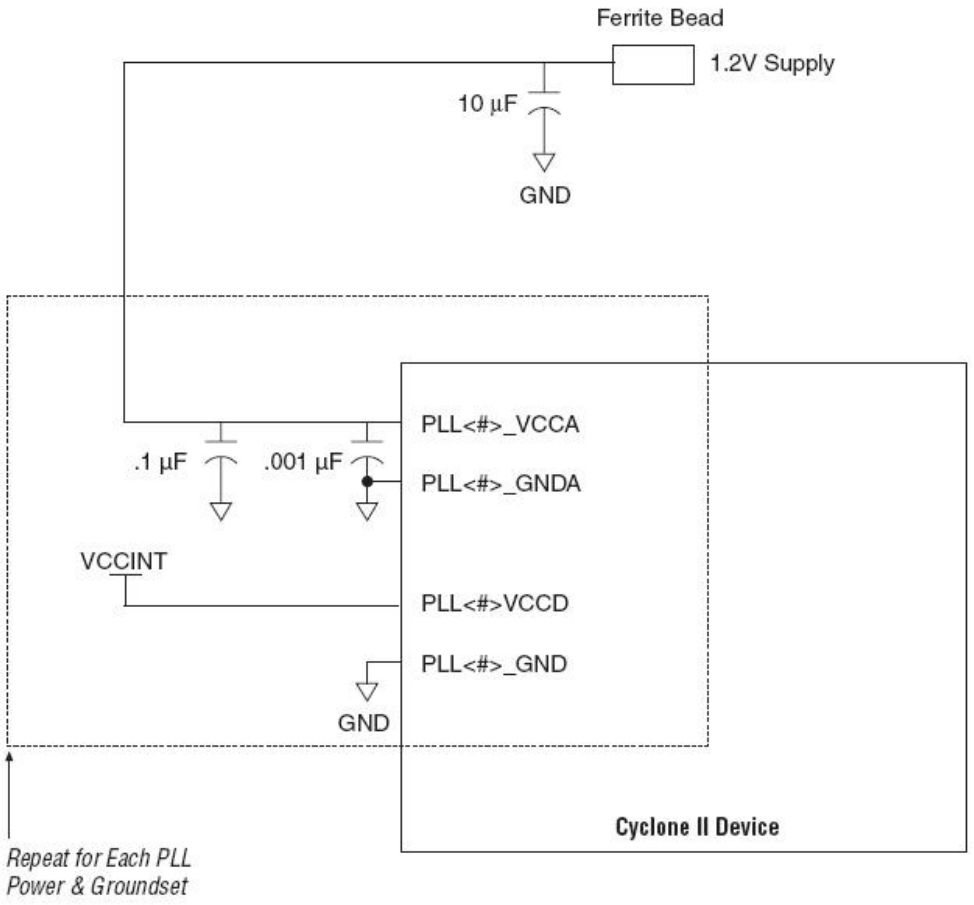
I/O Path Compensation in Cyclone II PLLs

Figure 7–7. Phase Relationship between Cyclone II PLL Clocks in Source-Synchronous Compensation Mode



PLL Power Management Issues

Figure 7–17. PLL Power Schematic for Cyclone II PLLs



Note to Figure 7–17:

(1) Applies to PLLs 1 through 4.

"Gating" Clocks for Reduced Power Consumption in Cyclone II FPGAs

Figure 7–14. clkena Implementation

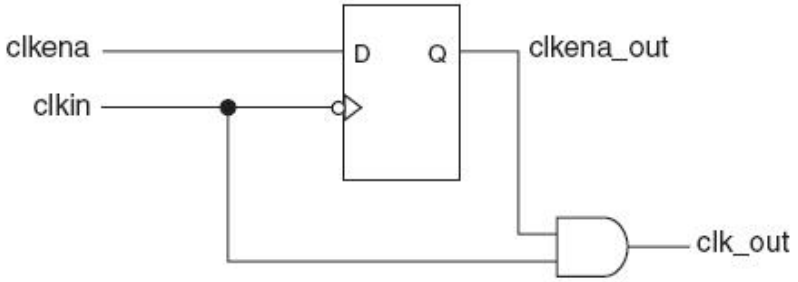
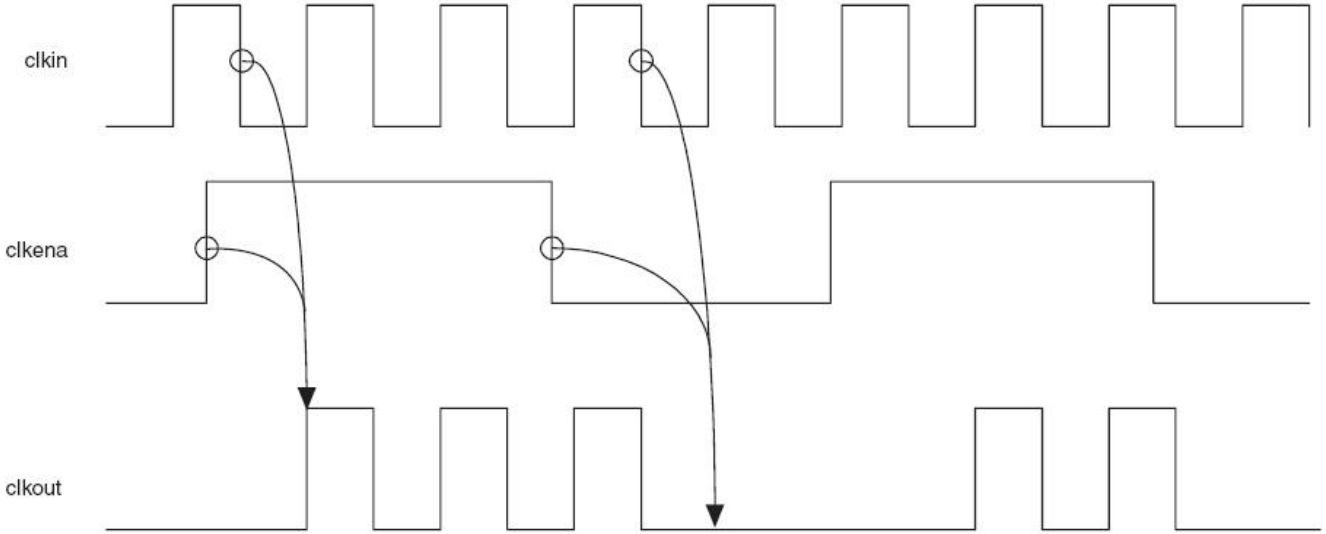
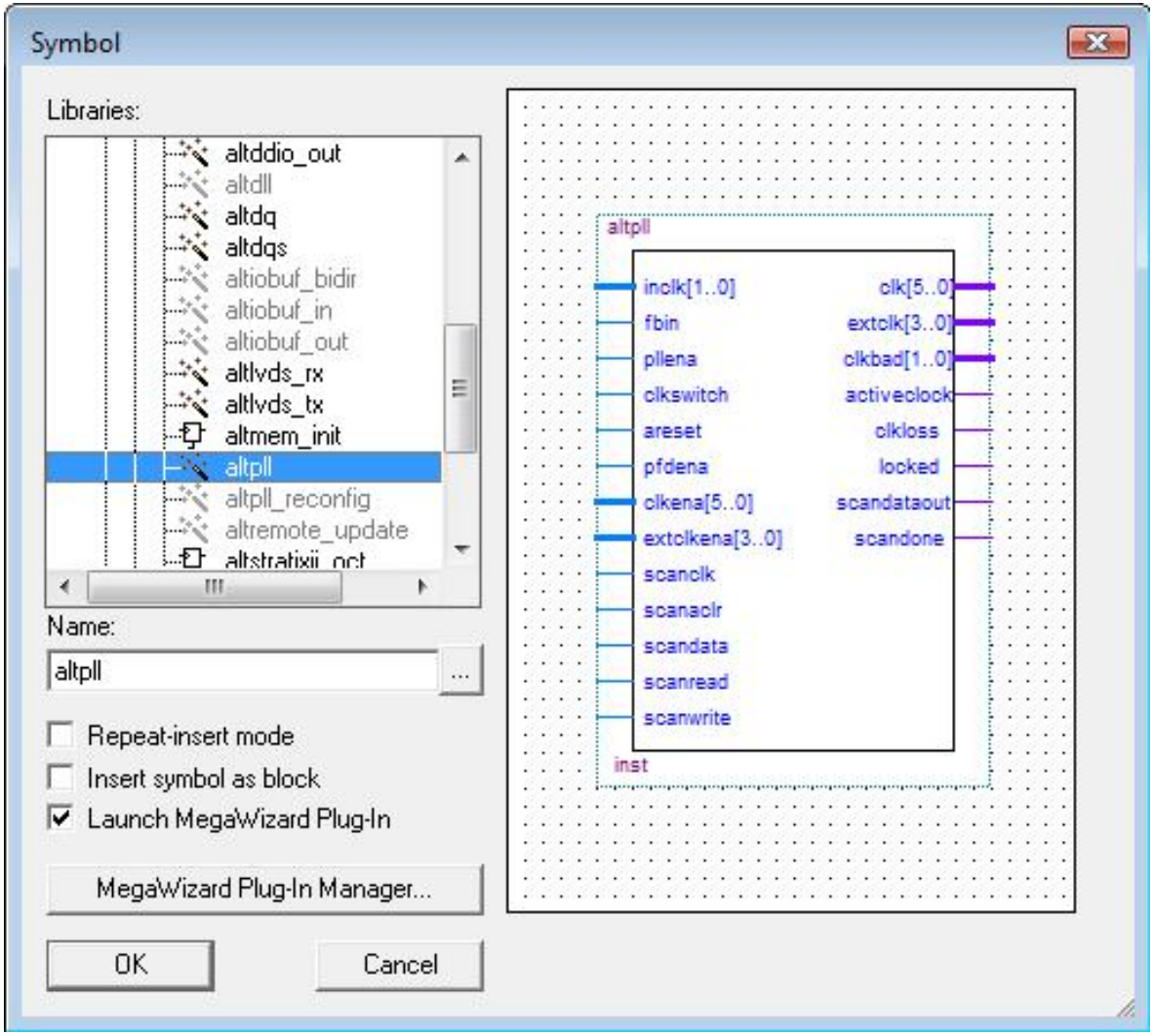


Figure 7–15. clkena Implementation



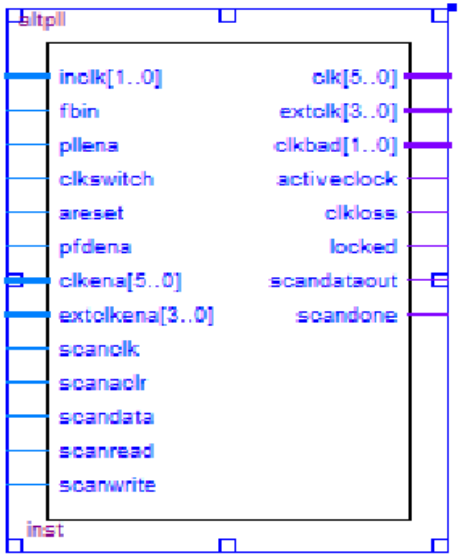
Configuring Cyclone II PLLs in Quartus II



PLL Instantiation

```
// This is the model of the clock converter device that utilizes one
// of the PLL elements in the Cyclone IV E to multiply the external
// clock by a factor of 4/5. This results in a 40 Mhz clock being
// produced whenever the module is driven by a 50Mhz signal. This is
// the frequency needed to obtain 800 x 600 60 Hz SVGA resolution.
module pll_clk_50_to_40(input clk_in,output clk_out);
   wire [4:0] clockgrp;
   assign  clk_out = clockgrp[0:0];

   altpll   altpll_component (
            .inclk ({1'h0, clk_in}),
            .clk (clockgrp),
            .activeclock (),
            .areset (1'b0),
            .clkbad (),
            .clkena ({6{1'b1}}),
            .clkloss (),
            .clkswitch (1'b0),
            .configupdate (1'b0),
            .enable0 (),
            .enable1 (),
            .extclk (),
            .extclkena ({4{1'b1}}),
            .fbin (1'b1),
            .fbmimicbidir (),
            .fbout (),
            .fref (),
            .icdrclk (),
            .locked (),
            .pfdena (1'b1),
            .phasecounterselect ({4{1'b1}}),
            .phasedone (),
            .phasestep (1'b1),
            .phaseupdown (1'b1),
            .pllena (1'b1),
            .scanaclr (1'b0),
            .scanclk (1'b0),
            .scanclkena (1'b1),
            .scandata (1'b0),
            .scandataout (),
            .scandone (),
            .scanread (1'b0),
            .scanwrite (1'b0),
            .sclkout0 (),
            .sclkout1 (),
            .vcooverrange (),
            .vcounderrange ());
```




```
defparam
   altpll_component.bandwidth_type = "AUTO",
   altpll_component.clk0_divide_by = 5,
   altpll_component.clk0_duty_cycle = 50,
   altpll_component.clk0_multiply_by = 4,
   altpll_component.clk0_phase_shift = "0",
   altpll_component.compensate_clock = "CLK0",
   altpll_component.inclk0_input_frequency = 20000,
   altpll_component.intended_device_family = "Cyclone IV E",
   altpll_component.lpm_hint = "CBX_MODULE_PREFIX=altpll0",
   altpll_component.lpm_type = "altpll",
   altpll_component.operation_mode = "NORMAL",
   altpll_component.pll_type = "AUTO",
   altpll_component.port_activeclock = "PORT_UNUSED",
   altpll_component.port_areset = "PORT_UNUSED",
   altpll_component.port_clkbad0 = "PORT_UNUSED",
   altpll_component.port_clkbad1 = "PORT_UNUSED",
   altpll_component.port_clkloss = "PORT_UNUSED",
   altpll_component.port_clkswitch = "PORT_UNUSED",
   altpll_component.port_configupdate = "PORT_UNUSED",
   altpll_component.port_fbin = "PORT_UNUSED",
   altpll_component.port_inclk0 = "PORT_USED",
   altpll_component.port_inclk1 = "PORT_UNUSED",
   altpll_component.port_locked = "PORT_UNUSED",
   altpll_component.port_pfdena = "PORT_UNUSED",
   altpll_component.port_phasecounterselect = "PORT_UNUSED",
   altpll_component.port_phasedone = "PORT_UNUSED",
   altpll_component.port_phasestep = "PORT_UNUSED",
   altpll_component.port_phaseupdown = "PORT_UNUSED",
   altpll_component.port_pllena = "PORT_UNUSED",
   altpll_component.port_scanaclr = "PORT_UNUSED",
   altpll_component.port_scanclk = "PORT_UNUSED",
   altpll_component.port_scanclkena = "PORT_UNUSED",
   altpll_component.port_scandata = "PORT_UNUSED",
   altpll_component.port_scandataout = "PORT_UNUSED",
   altpll_component.port_scandone = "PORT_UNUSED",
   altpll_component.port_scanread = "PORT_UNUSED",
   altpll_component.port_scanwrite = "PORT_UNUSED",
   altpll_component.port_clk0 = "PORT_USED",
   altpll_component.port_clk1 = "PORT_UNUSED",
   altpll_component.port_clk2 = "PORT_UNUSED",
   altpll_component.port_clk3 = "PORT_UNUSED",
   altpll_component.port_clk4 = "PORT_UNUSED",
   altpll_component.port_clk5 = "PORT_UNUSED",
   altpll_component.port_clkena0 = "PORT_UNUSED",
   altpll_component.port_clkena1 = "PORT_UNUSED",
   altpll_component.port_clkena2 = "PORT_UNUSED",
   altpll_component.port_clkena3 = "PORT_UNUSED",
   altpll_component.port_clkena4 = "PORT_UNUSED",
   altpll_component.port_clkena5 = "PORT_UNUSED",
   altpll_component.port_extclk0 = "PORT_UNUSED",
   altpll_component.port_extclk1 = "PORT_UNUSED",
   altpll_component.port_extclk2 = "PORT_UNUSED",
   altpll_component.port_extclk3 = "PORT_UNUSED",
   altpll_component.width_clock = 5;
```