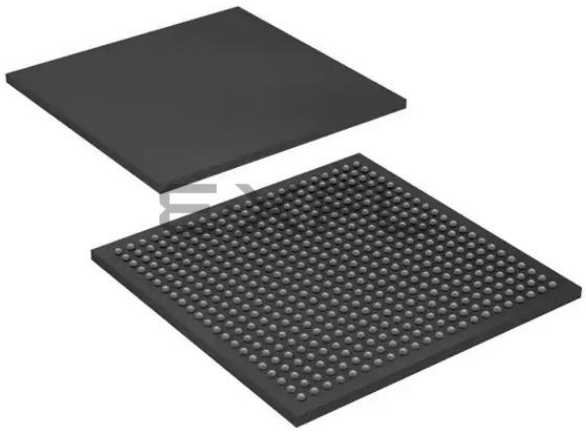


Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Obsolete
Number of LABs/CLBs	2700
Number of Logic Elements/Cells	67500
Total RAM Bits	2699264
Number of I/O	296
Number of Gates	-
Voltage - Supply	0.86V ~ 1.15V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BBGA, FCBGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep3sl70f484c3n

LAB Power Management Techniques

The following techniques are used to manage static and dynamic power consumption within the LAB:

- Stratix III low-voltage devices (L ordering code suffix) offer selectable core voltage to reduce both DC and AC power.
- To save AC power, Quartus II forces all adder inputs low when ALM adders are not in use.
- Stratix III LABs operate in high-performance mode or low-power mode. The Quartus II software automatically chooses the appropriate mode for an LAB based on the design to optimize speed vs. leakage trade-offs.
- Clocks represent a significant portion of dynamic power consumption due to their high switching activity and long paths. The LAB clock that distributes a clock signal to registers within a LAB is a significant contributor to overall clock power consumption. Each LAB's clock and clock enable signal are linked. For example, a combinational ALUT or register in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. To disable LAB-wide clock power consumption without disabling the entire clock tree, use the LAB-wide clock enable to gate the LAB-wide clock. The Quartus II software automatically promotes register-level clock enable signals to the LAB-level. All registers within an LAB that share a common clock and clock enable are controlled by a shared gated clock. To take advantage of these clock enables, use a clock enable construct in your HDL code for the registered logic.

Refer to the *Power Optimization* chapter in section 3 of the *Quartus II Handbook* for details on implementation.

For detailed information about Stratix III programmable power capabilities, refer to the *Programmable Power and Temperature Sensing Diode in Stratix III Devices chapter in volume 1* of the *Stratix III Device Handbook*.

Conclusion

Logic array block and adaptive logic modules are the basic building blocks of the Stratix III device. You can use these to configure logic functions, arithmetic functions, and register functions. The ALM provides advanced features with efficient logic utilization and is completely backward-compatible.

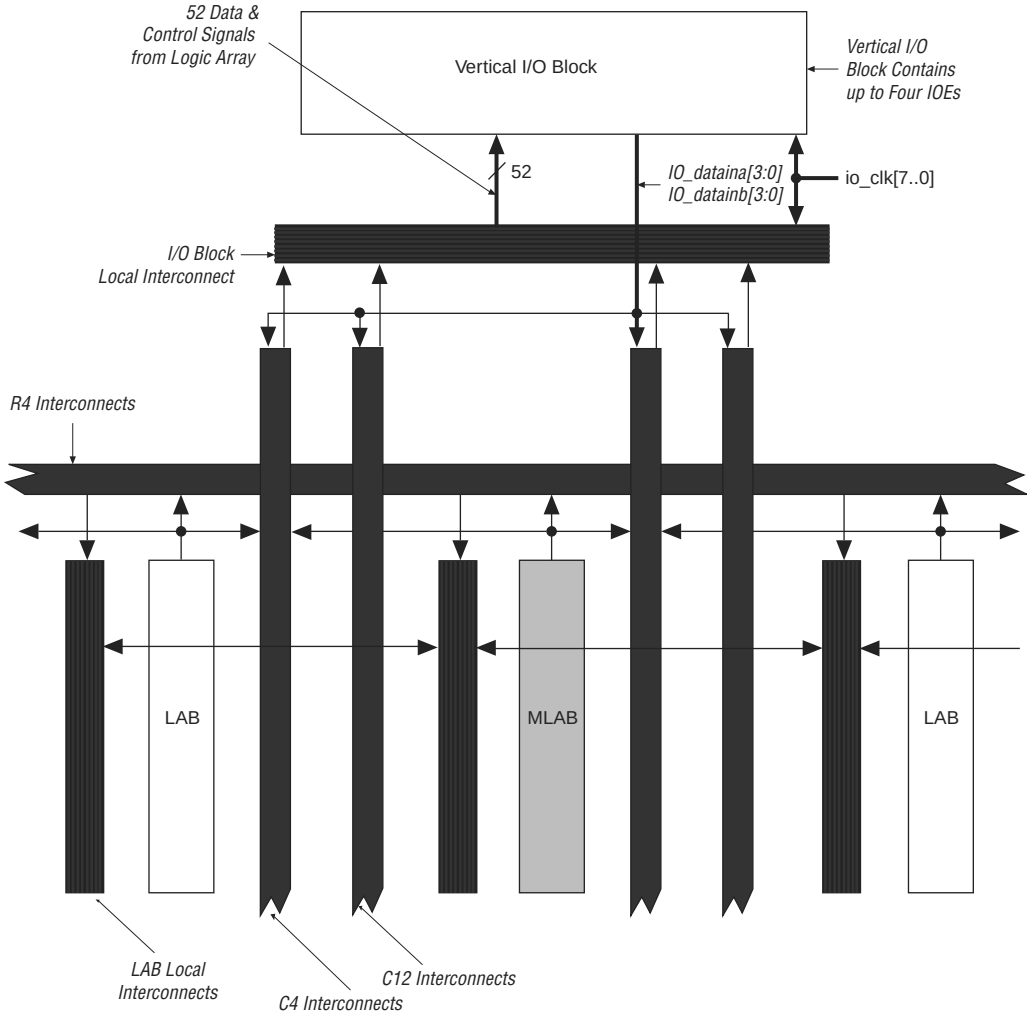
Chapter Revision History

Table 2–1shows the revision history for this document.

Table 2–1. Chapter Revision History(Sheet 1 of 2)

Date and Revision	Changes Made	Summary of Changes
February 2009, version 1.5	Removed “Referenced Documents” section.	—
October 2008, version 1.4	■ Updated “LAB Control Signals”, and “Carry Chain” Sections. ■ Updated New Document Format.	—

Figure 3–10. Column I/O Block Connection to Interconnect



Conclusion

Stratix III devices consist of an array of logic blocks such as LABs, TriMatrix memory, DSP blocks, and IOEs. These blocks communicate with themselves and one another through the MultiTrack interconnect structures. The Quartus II compiler automatically routes critical design paths on faster interconnects to improve design performance and optimize the device resources.

MLABs support byte-enable via emulation. There will be increased logic utilization when the byte-enables are emulated.

The default value for the byte-enable signals is high (enabled), in which case writing is controlled only by the write enable signals. The byte-enable registers have no clear port. When using parity bits on the M9K and M144K blocks, the byte-enable controls all nine bits (eight bits of data plus one parity bit). When using parity bits on the MLAB, the byte-enable controls all 10 bits in the widest mode.

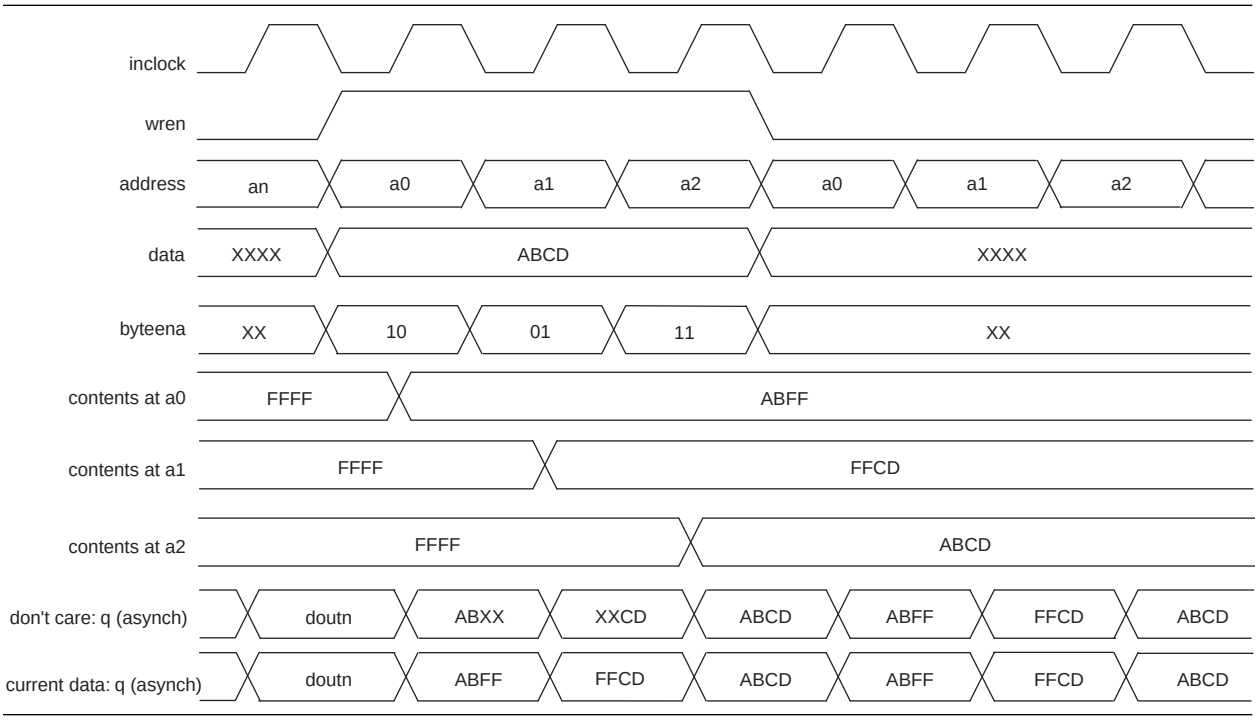
Byte-enables operate in a one-hot fashion, with the LSB of the `byteena` signal corresponding to the least significant byte of the data bus. For example, if you are using a RAM block in ×18 mode, with `byteena = 01`, `data[8..0]` is enabled and `data[17..9]` is disabled. Similarly, if `byteena = 11`, both `data[8..0]` and `data[17..9]` are enabled. Byte-enables are active high.

You cannot use the byte-enable feature when using the ECC feature on M144K blocks.

Figure 4–1 shows how the write enable (`wren`) and byte-enable (`byteena`) signals control the operations of the M9K and M144K.

When a byte-enable bit is de-asserted during a write cycle, the corresponding data byte output can appear as either a "don't care" value or the current data at that location. The output value for the masked byte is controllable via the Quartus II software. When a byte-enable bit is asserted during a write cycle, the corresponding data byte output also depends on the setting chosen in the Quartus II software.

Figure 4–1. Stratix III Byte-Enable Functional Waveform for M9K and M144K



Read/Write Clock Mode

Stratix III TriMatrix memory blocks can implement read/write clock mode for simple dual-port memories. In this mode, a write clock controls the data-input, write-address, and write-enable registers. Similarly, a read clock control the data-output, read-address, and read-enable registers. The memory blocks support independent clock enables for both the read and write clocks. Asynchronous clears are available on data output latches and registers only.

When using read/write mode, if you perform a simultaneous read/write to the same address location, the output read data will be unknown. If you require the output data to be a known value in this case, use either single-clock mode or input/output clock mode and choose the appropriate read-during-write behavior in the Megawizard.

Single Clock Mode

Stratix III TriMatrix memory blocks can implement single-clock mode for true dual-port, simple dual-port, and single-port memories. In this mode, a single clock, together with a clock enable, is used to control all registers of the memory block. Asynchronous clears are available on output latches and output registers only.

Design Considerations

This section describes guidelines for designing with TriMatrix memory blocks.

Selecting TriMatrix Memory Blocks

The Quartus II software automatically partitions user-defined memory into embedded memory blocks by taking into account both speed and size constraints placed on your design. For example, the Quartus II software may spread out a memory across multiple memory blocks when resources are available to increase the performance of the design. You can manually assign the memory to a specific block size via the RAM MegaWizard Plug-In Manager.

MLABs can implement single-port SRAM through emulation via the Quartus II software. Emulation results in minimal additional logic resources being used. Because of the dual-purpose architecture of the MLAB, it only has data input registers and output registers in the block. MLABs gain input address registers and additional optional data output registers from adjacent ALMs by using register packing.

For more information about register packing, refer to the *Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

Conflict Resolution

When using the memory blocks in true dual-port mode, it is possible to attempt two write operations to the same memory location (address). Since no conflict resolution circuitry is built into the memory blocks, this results in unknown data being written to that location. Therefore, you must implement conflict resolution logic external to the memory block to avoid address conflicts.

Each Half Block has its own `signa` and `signb` signal. Therefore, all of the `data A` inputs feeding the same DSP Half Block must have the same sign representation. Similarly, all of the `data B` inputs feeding the same DSP Half Block must have the same sign representation. The multiplier offers full precision regardless of the sign representation in all operational modes except for full precision 18 x 18 loopback and Two-Multiplier Adder modes. Refer to "Two-Multiplier Adder Sum Mode" on page 5–21 for details.

☞ When the `signa` and `signb` signals are unused, the Quartus II software sets the multiplier to perform unsigned multiplication by default.

The outputs of the multipliers are the only outputs that can feed into the first-stage adder, as shown in Figure 5–6. There are four first-stage adders in a DSP block (two adders per half DSP block). The first-stage adder block has the ability to perform addition and subtraction. The control signal for addition or subtraction is static and has to be configured upon compile time. The first-stage adders are used by the sum modes to compute the sum of two multipliers, 18 × 18-complex multipliers, and to perform the first stage of a 36 × 36 multiply and shift operation.

Depending on your specifications, the output of the first-stage adder has the option to feed into the pipeline registers, second-stage adder, round and saturation unit, or the output registers.

Pipeline Register Stage

The output from the first-stage adder can either feed or bypass the pipeline registers, as shown in Figure 5–6. Pipeline registers increase the DSP block's maximum performance (at the expense of extra cycles of latency), especially when using the subsequent DSP block stages. Pipeline registers split up the long signal path between the input-registers/multiplier/first-stage adder and the second-stage adder/round-and-saturation/output-registers, creating two shorter paths.

Second-Stage Adder

There are four individual 44-bit second-stage adders per DSP block (2 adders per half DSP block). You can configure the second-stage adders as follows:

- The final stage of a 36-bit multiplier
- A sum of four (18 × 18)
- An accumulator (44-bits maximum)
- A chained output summation (44-bits maximum)

☞ The chained-output adder can be used at the same time as a second-level adder in chained output summation mode.

☞ The output of the second-stage adder has the option to go into the round and saturation logic unit or the output register.

☞ You cannot use the second-stage adder independently from the multiplier and first-stage adder.

Figure 5–14. Two-Multiplier Adder Mode for Half-DSP Block

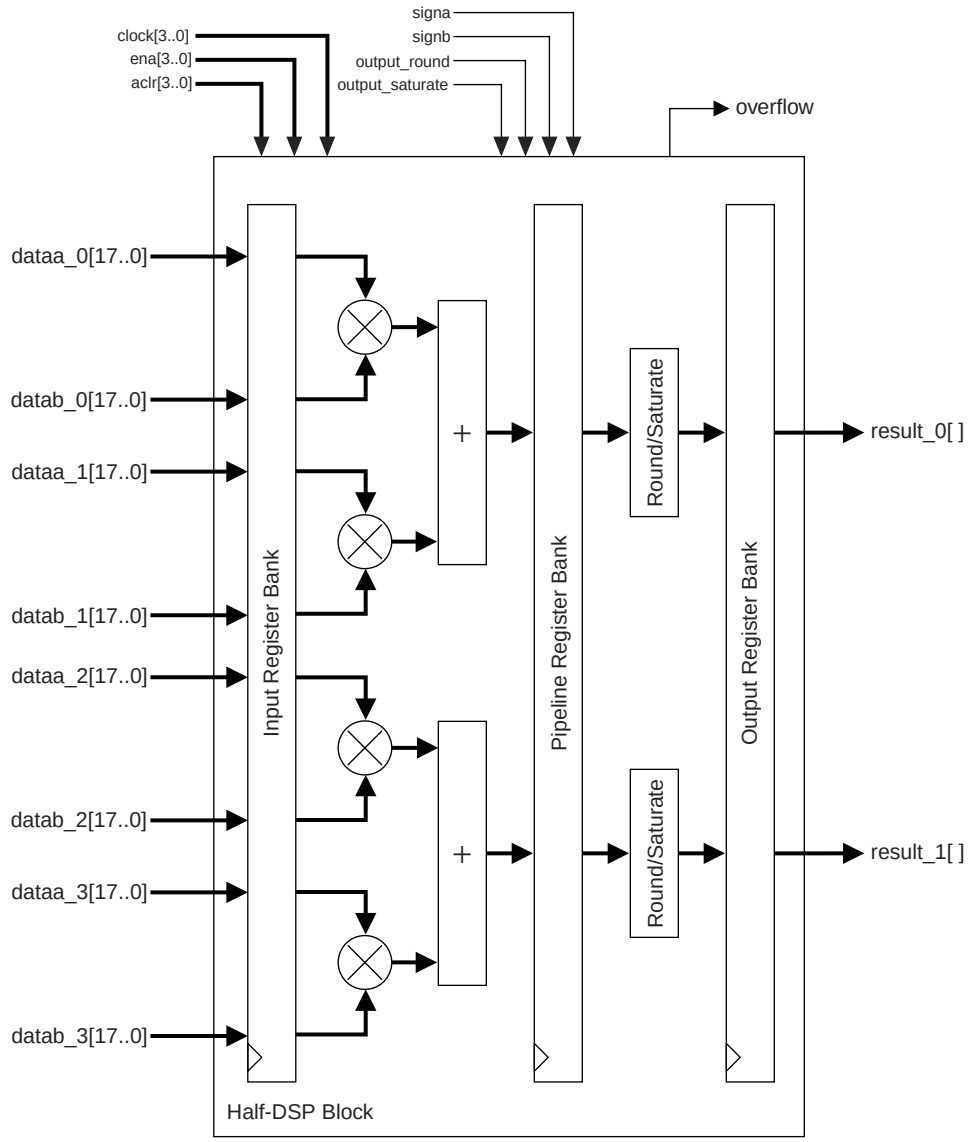


Table 6–6 lists the connectivity between the dedicated clock input pins and RCLKs in device Quadrant 4. A given clock input pin can drive two adjacent regional clock networks to create a dual-regional clock network.

Table 6–6. Clock Input Pin Connectivity to Regional Clock Networks (Quadrant 4)

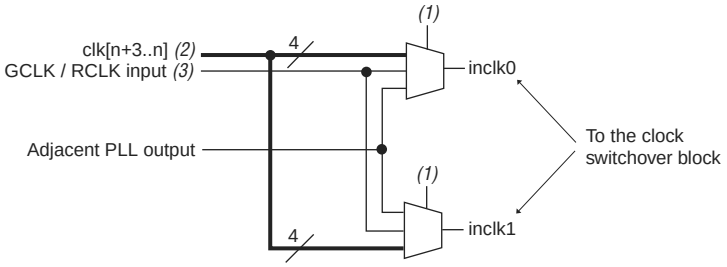
Clock Resource	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK6	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK7	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK8	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK9	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
RCLK10	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK11	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK12	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK13	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK14	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK15	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
RCLK16	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK17	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK18	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK19	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
RCLK20	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK21	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—

Table 6–7 lists the dedicated clock input pin connectivity to Stratix III device PLLs.

Table 6–7. Stratix III Device PLLs and PLL Clock Pin Drivers (Part 1 of 2) *(Note 1)*

Dedicated Clock Input Pin (CLKp/n pins)	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
CLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK4	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK5	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK6	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK7	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK8	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK9	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK10	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK11	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK12	—	—	—	—	—	—	—	—	—	—	✓	✓

Figure 6–11. Clock Input Multiplexer Logic for L2, L3, T1, T2, B1, B2, R2, and R3 PLLs



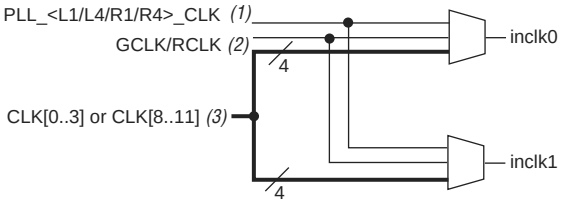
Notes to Figure 6–11:

(1) The input clock multiplexing is controlled through a configuration file (**.sof** or **.pof**) only and cannot be dynamically controlled in user mode operation.

(2) n=0 for L2 and L3 PLLs; n=4 for B1 and B2 PLLs; n=8 for R2 and R3 PLLs, and n=12 for T1 and T2 PLLs.

(3) The global (GCLK) or regional (RCLK) clock input can be driven by an output from another PLL, a pin-driven global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin driven dedicated global or regional clock. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

Figure 6–12. Clock Input Multiplexer Logic for L1, L4, R1, and R4 PLLs



Notes to Figure 6–12:

(1) Dedicated clock input pins to PLLs - L1, L4, R1 and R4, respectively. For example, `PLL_L1_CLK` is the dedicated clock input for `PLL_L1`.

(2) The global (GCLK) or regional (RCLK) clock input can be driven by an output from another PLL, a pin-driven global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin driven dedicated global or regional clock. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

(3) The center clock pins can feed the corner PLLs on the same side directly, through a dedicated path. However, these paths may not be fully compensated.

Clock Control Block

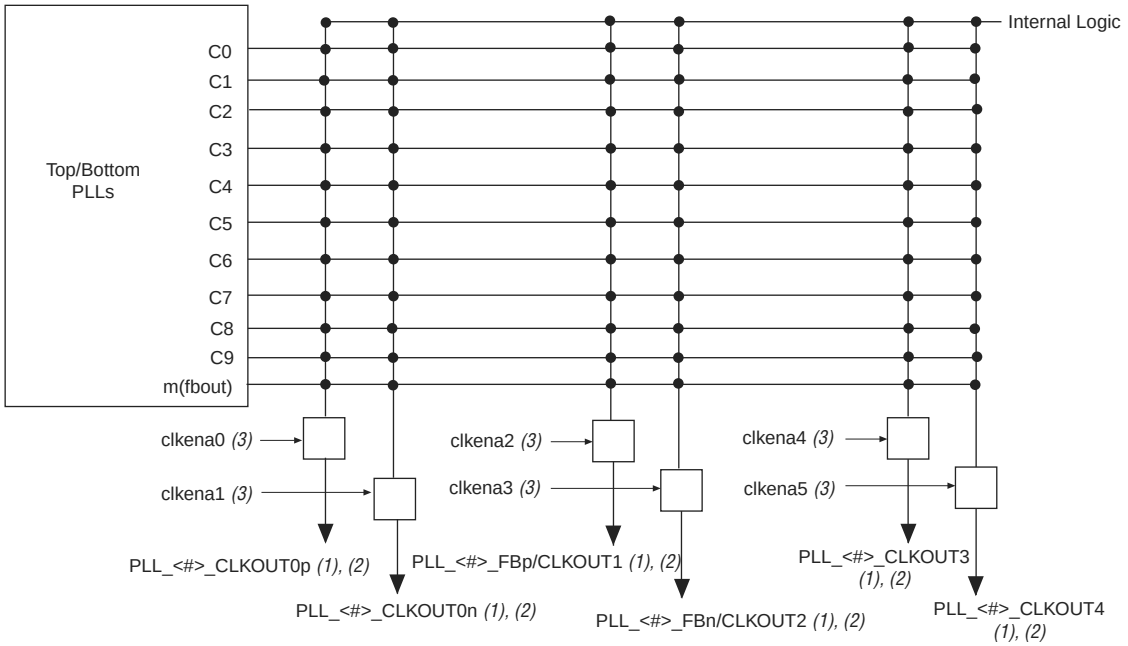
Every global and regional clock network has its own clock control block. The control block provides the following features:

- Clock source selection (dynamic selection for global clocks)
- Global clock multiplexing
- Clock power down (static or dynamic clock enable or disable)

You can select the clock source for the global clock select block either statically or dynamically. You can either statically select the clock source using a setting in the Quartus II software, or you can dynamically select the clock source using internal logic to drive the multiplexer select inputs. When selecting the clock source dynamically, you can either select two PLL outputs (such as `CLK0` or `CLK1`), or a combination of clock pins or PLL outputs.

Figure 6–20 shows the clock I/O pins associated with Top/Bottom PLLs.

Figure 6–20. External Clock Outputs for Top/Bottom PLLs



Notes to Figure 6–20:

(1) These clock output pins can be fed by any one of the C[9..0], m counters.

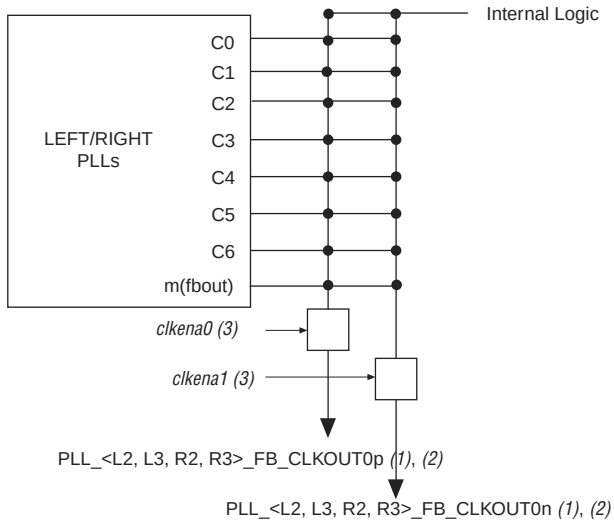
(2) The CLKOUT0p and CLKOUT0n pins can be either single-ended or differential clock outputs. CLKOUT1 and CLKOUT2 pins are dual-purpose I/O pins that can be used as two single-ended outputs, one differential external feedback input pin pair or one single-ended external feedback input pin (CLKOUT1 only). CLKOUT3 and CLKOUT4 pins are two single-ended output pins.

(3) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

Any of the output counters (C[9..0] on Top/Bottom PLLs and C[6..0] on Left/Right PLLs) or the M counter can feed the dedicated external clock outputs, as shown in Figure 6–20 and Figure 6–21. Therefore, one counter or frequency can drive all output pins available from a given PLL.

Each Left/Right PLL supports two clock I/O pins, configured as either two single-ended I/Os or one differential I/O pair. When using both pins as single-ended I/Os, one of them can be the clock output while the other pin is the external feedback input (FB) pin. Hence, Left/Right PLLs only support external feedback mode for single-ended I/O standards.

Figure 6–21. External Clock Outputs for Left/Right PLLs



Notes to Figure 6–21:

(1) These clock output pins can be fed by any one of the `C[6..0]`, m counters.

(2) The `CLKOUT0p` and `CLKOUT0n` pins are dual-purpose I/O pins that can be used as two single-ended outputs or one single-ended output and one external feedback input pin.

(3) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

Each pin of a single-ended output pair can either be in-phase or 180-degrees out-of-phase. The Quartus II software places the NOT gate in the design into the IOE to implement 180-degrees phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins (in the top and bottom banks) as well as LVDS, LVPECL, differential HSTL, and differential SSTL.

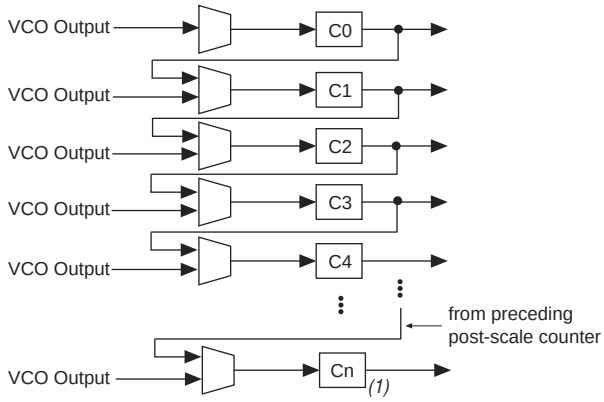
To determine which I/O standards are supported by the PLL clock input and output pins, refer to the *Stratix III Device I/O Features* chapter.

Stratix III PLLs can also drive out to any regular I/O pin through the global or regional clock network. You can use the external clock output pins as user I/O pins if external PLL clocking is not needed.

Post-Scale Counter Cascading

The Stratix III PLLs support post-scale counter cascading to create counters larger than 512. This is automatically implemented in the Quartus II software by feeding the output of one C counter into the input of the next C counter as shown in Figure 6–31.

Figure 6–31. Counter Cascading



Note to Figure 6–31:

(1) n = 6 or n = 9

When cascading post-scale counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings. For example, if C0 = 40 and C1 = 20, then the cascaded value is C0*C1 = 800.

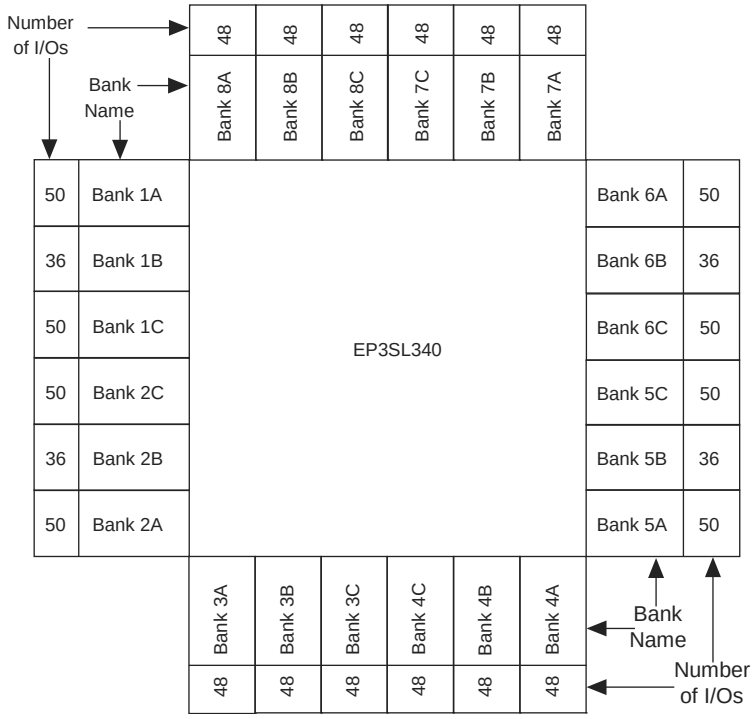
Post-scale counter cascading is set in the configuration file. It cannot be done using PLL reconfiguration.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on the PLL post-scale counters. The duty-cycle setting is achieved by a low and high time-count setting for the post-scale counters. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices. The post-scale counter value determines the precision of the duty cycle. The precision is defined by 50% divided by the post-scale counter value. For example, if the C0 counter is 10, then steps of 5% are possible for duty-cycle choices between 5% to 90%.

If the PLL is in external feedback mode, you must set the duty cycle for the counter driving the fbin pin to 50%. Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

Figure 7–6. Number of I/Os in Each Bank in EP3SL340 Devices in the 1760-pin FineLine BGA Package *(Note 1), (2)*



Notes to Figure 7–6:

(1) All I/O pin counts include dedicated clock inputs pins. The pin count includes all general purpose I/O, dedicated clock pins, and dual-purpose configuration pins. Dedicated configuration pins are not included in the pin count.

(2) Figure 7–6 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

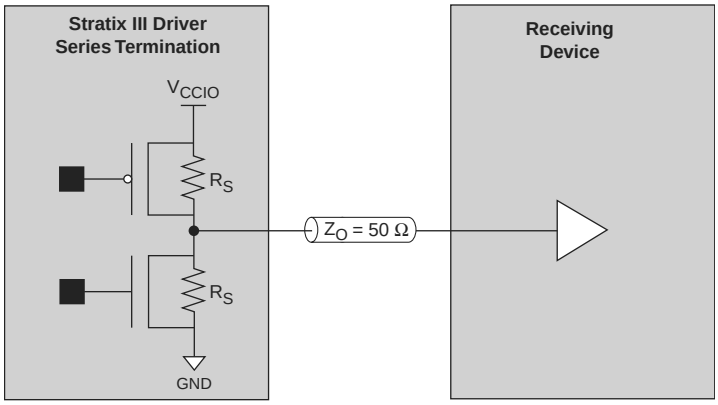
Stratix III I/O Structure

The I/O element (IOE) in Stratix III devices contains a bi-directional I/O buffer and I/O registers to support a complete embedded bi-directional single data rate or DDR transfer. The IOEs are located in I/O blocks around the periphery of the Stratix III device. There are up to four IOEs per row I/O block and four IOEs per column I/O block. The row IOEs drive row, column, or direct link interconnects. The column IOEs drive column interconnects.

The Stratix III bi-directional IOE also supports the following features:

- Programmable input delay
- Programmable output-current strength
- Programmable slew rate
- Programmable output delay
- Programmable bus-hold
- Programmable pull-up resistor
- Open-drain output
- On-chip series termination with calibration

Figure 7–8. On-Chip Series Termination without Calibration for Stratix III Devices



To use OCT for the SSTL Class I standard, you should select the **50-Ω on-chip series termination** setting, eliminating the external 25-Ω R_S (to match the 50-Ω transmission line). For the SSTL Class II standard, you should select the **25-Ω on-chip series termination** setting (to match the 50-Ω transmission line and the near-end external 50-Ω pull-up to V_{TT}).

On-Chip Series Termination with Calibration

Stratix III devices support OCT R_S with calibration in all banks. The OCT R_S calibration circuit compares the total impedance of the I/O buffer to the external 25-Ω ±1% or 50-Ω ±1% resistors connected to the RUP and RDN pins, and dynamically enables or disables the transistors until they match. The R_S shown in Figure 7–9 is the intrinsic impedance of transistors. Calibration occurs at the end of device configuration. When the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers. When calibration is not taking place, the RUP and RDN pins go to a tri-state condition.

Figure 7–9. On-Chip Series Termination with Calibration for Stratix III Devices

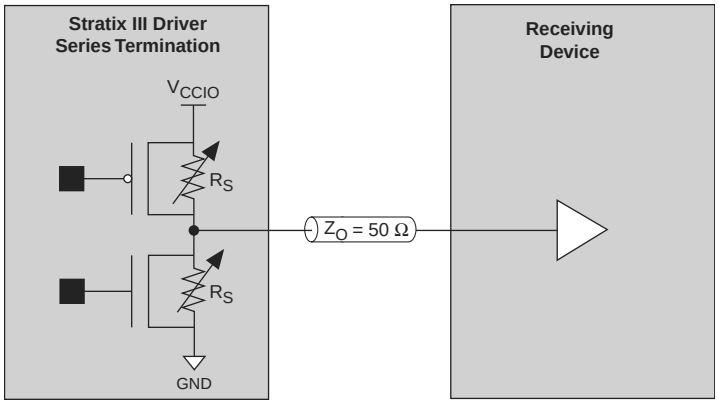


Table 7–8 lists I/O standards that support OCT R_S with calibration.

I/O Termination

I/O termination requirements for single-ended and differential I/O standards are discussed in this section.

Single-Ended I/O Standards

Although single-ended, non-voltage-referenced I/O standards do not require termination, impedance matching may be necessary to reduce reflections and improve signal integrity.

Voltage-referenced I/O standards require both an input reference voltage, $V_{REF,}$ and a termination voltage, V_{TT}. The reference voltage of the receiving device tracks the termination voltage of the transmitting device. Each voltage-referenced I/O standard requires a unique termination setup. For example, a proper resistive signal termination scheme is critical in SSTL2 standards to produce a reliable DDR memory system with superior noise margin.

Stratix III OCT R_S and OCT R_T provide the convenience of no external components. Alternatively, you can use external pull-up resistors to terminate the voltage-referenced I/O standards, such as SSTL and HSTL.

Differential I/O Standards

Differential I/O standards typically require a termination resistor between the two signals at the receiver. The termination resistor must match the differential load impedance of the signal line. Stratix III devices provide an optional differential on-chip resistor when using LVDS.

For PCB layout guidelines, refer to *AN 224: High-Speed Board Layout Guidelines* and *AN 315: Guidelines for Designing High-Speed FPGA PCBs*.

I/O Banks Restrictions

Each I/O bank can simultaneously support multiple I/O standards. The following sections provide guidelines for mixing non-voltage-referenced and voltage-referenced I/O standards in Stratix III devices.

Non-Voltage-Referenced Standards

Each Stratix III device I/O bank has its own V_{CCIO} pins and supports only one V_{CCIO}, either 1.2, 1.5, 1.8, 2.5, 3.0, or 3.3 V. An I/O bank can simultaneously support any number of input signals with different I/O standard assignments, as listed in Table 7–2.

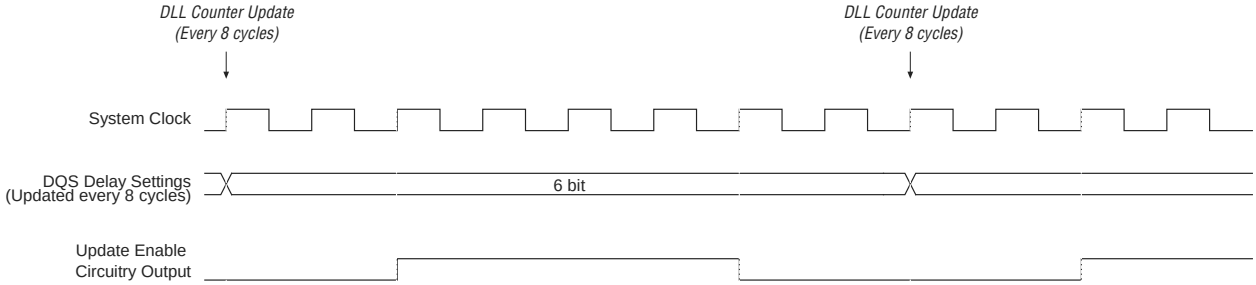
For output signals, a single I/O bank supports non-voltage-referenced output signals that are driving at the same voltage as V_{CCIO}. Since an I/O bank can only have one V_{CCIO} value, it can only drive out that one value for non-voltage-referenced signals. For example, an I/O bank with a 2.5-V V_{CCIO} setting can support 2.5-V standard inputs and outputs and 3-V LVCMOS inputs (not output or bi-directional pins).

You can also bypass the DQS delay chain to achieve 0° phase shift.

Update Enable Circuitry

Both the DQS delay settings and phase-offset settings pass through a register before going into the DQS delay chains. The registers are controlled by the update enable circuitry to allow enough time for any changes in the DQS delay setting bits to arrive at all the delay elements. This allows them to be adjusted at the same time. The update enable circuitry enables the registers to allow enough time for the DQS delay settings to travel from the DQS phase-shift circuitry or core logic to all the DQS logic blocks before the next change. It uses the input reference clock or a user clock from the core to generate the update enable output. The ALTMEMPHY megafunction uses this circuit by default. See Figure 8–14 for an example waveform of the update enable circuitry output.

Figure 8–14. Example of a DQS Update Enable Waveform



DQS Postamble Circuitry

For external memory interfaces that use a bi-directional read strobe like DDR3, DDR2, and DDR SDRAM, the DQS signal is low before going to or coming from a high-impedance state. The state where DQS is low, just after a high-impedance state, is called the preamble. The state where DQS is low, just before it returns to a high-impedance state, is called the postamble. There are preamble and postamble specifications for both read and write operations in DDR3, DDR2, and DDR SDRAM. The DQS postamble circuitry ensures that the data is not lost if there is noise on the DQS line during the end of a read operation that occurs while the DQS is in a postamble state.

Stratix III devices have a dedicated postamble register that you can control to ground the shifted DQS signal used to clock the DQ input registers at the end of a read operation. This ensures that any glitches on the DQS input signals at the end of the read postamble time do not affect the DQ IOE registers.

In addition to the dedicated postamble register, Stratix III devices also have an HDR block inside the postamble enable circuitry. These registers are used if the controller is running at half the frequency of the I/Os.

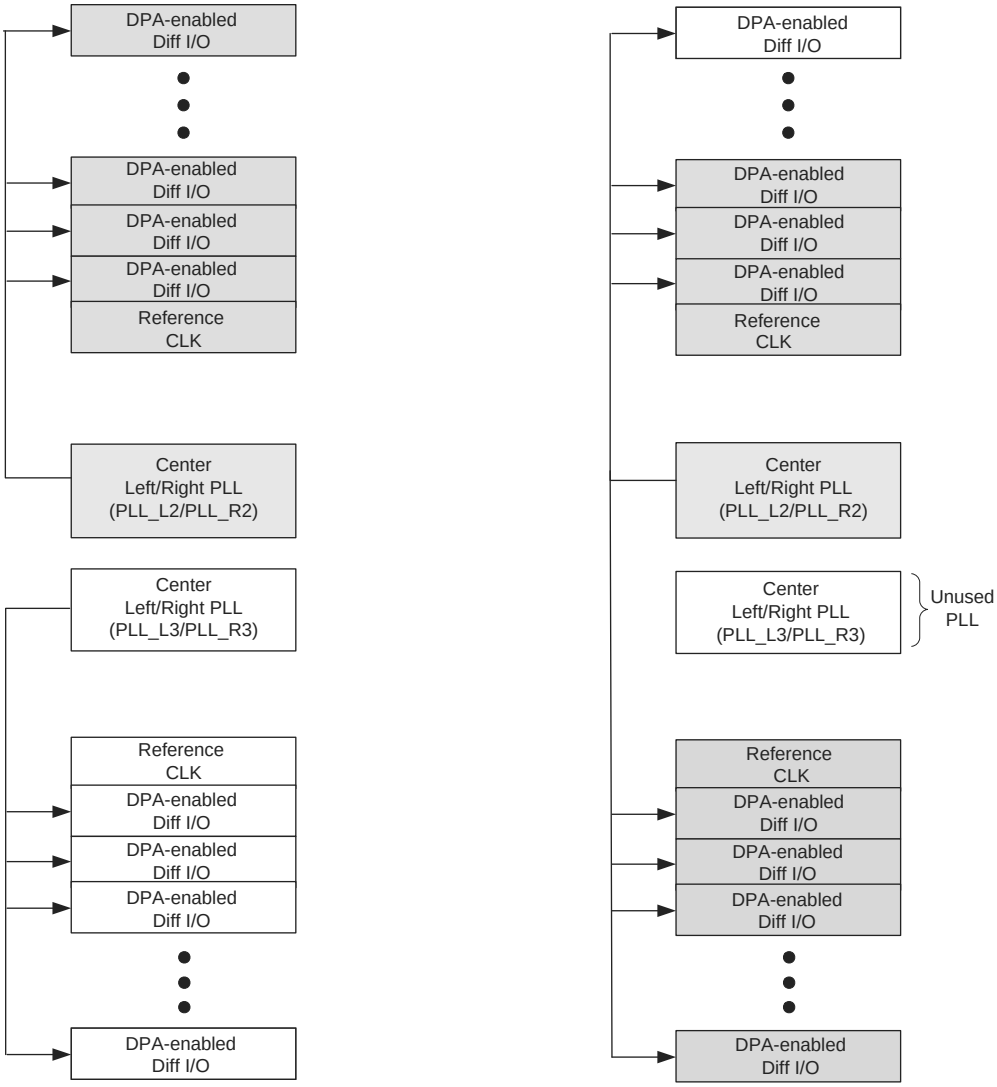
Using Both Center Left/Right PLLs

Both center left/right PLLs can be used to drive DPA-enabled channels simultaneously, as long as they drive these channels in their adjacent banks only, as shown in Figure 9–19.

If one of the center left/right PLLs drive the top and bottom banks, the other center left/right PLL cannot be used to drive the differential channels, as shown in Figure 9–19.

If the top PLL_L2/PLL_R2 drives DPA-enabled channels in the lower differential bank, the PLL_L3/PLL_R3 cannot drive DPA-enabled channels in the upper differential banks and vice versa. In other words, the center left/right PLLs cannot drive cross-banks simultaneously, as shown in Figure 9–20.

Figure 9–19. Center Left/Right PLLs Driving DPA-Enabled Differential I/Os



Upon power-up, the Stratix III devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. If PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS low, and tri-state all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration in this scheme, the download cable generates a low-to-high transition on the nCONFIG pin.

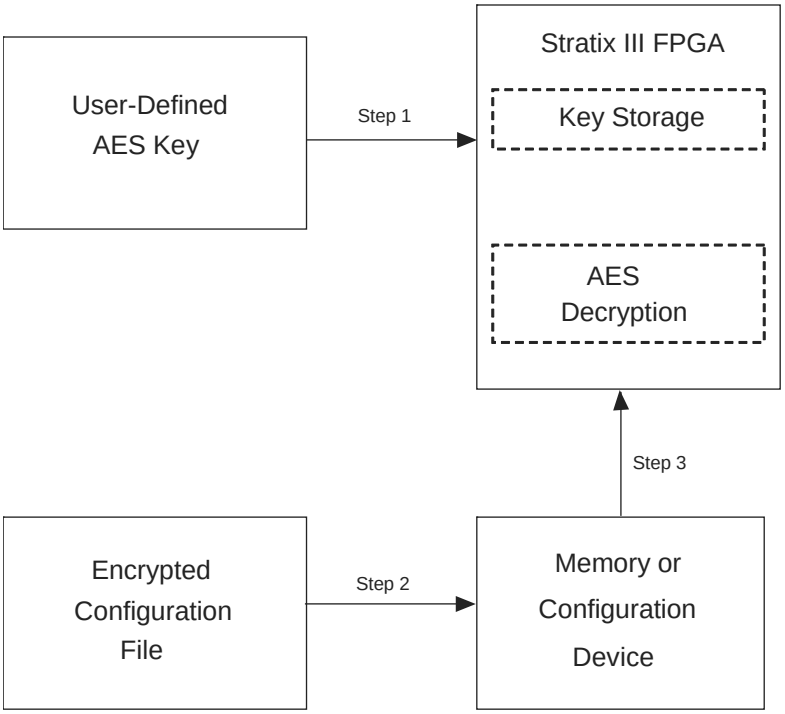
To begin configuration, power the V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} voltages (for the banks where the configuration and JTAG pins reside) to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. The programming hardware or download cable then places the configuration data one bit at a time on the device's DATA0 pin. The configuration data is clocked into the target device until CONF_DONE goes high. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

When using a download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs. Additionally, the **Enable user-supplied start-up clock** (**CLKUSR**) option has no affect on the device initialization since this option is disabled in the SOF when programming the device using the Quartus II programmer and download cable. Therefore, if you turn on the CLKUSR option, you do not need to provide a clock on CLKUSR when you are configuring the device with the Quartus II programmer and a download cable.

Figure 11–17 shows PS configuration for Stratix III devices using a USB-Blaster, MasterBlaster, ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable.

Figure 14–1. Design Security *(Note 1)*



Note to Figure 14–1:

(1) Step 1, Step 2, and Step 3 correspond to the procedure detailed in the "Stratix III Design Security Solution" section.

Security Modes Available

There are several security modes available on the Stratix III device, which are described as follows:

Volatile Key

Secure operation with volatile key programmed and required external battery—this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board-level testing only.

Non-Volatile Key

Secure operation with one time programmable (OTP) security key programmed—this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board-level testing only.

Non-Volatile Key with Tamper Protection Bit Set

Secure operation in tamper resistant mode with OTP security key programmed—only encrypted configuration bitstreams are allowed to configure the device.
Tamper protection disables JTAG configuration with unencrypted configuration bitstream.

Error Detection Fundamentals

Error detection determines if the data received through a medium is corrupted during transmission. To accomplish this, the transmitter uses a function to calculate a checksum value for the data and appends the checksum to the original data frame. The receiver uses the same calculation methodology to generate a checksum for the received data frame and compares the received checksum to the transmitted checksum. If the two checksum values are equal, the received data frame is correct and no data corruption occurred during transmission or storage.

The error detection CRC feature uses the same concept. When Stratix III devices have been configured successfully and are in user mode, the error detection CRC feature ensures the integrity of the configuration data.

There are two CRC error checks. One always runs during configuration, the second optional CRC error check runs in the background in user mode. Both CRC error checks use the same CRC polynomial but different error detection implementations.

For more information, refer to "Configuration Error Detection" and "User Mode Error Detection".

Configuration Error Detection

In configuration mode, a frame-based CRC is stored within the configuration data and contains the CRC value for each data frame.

During configuration, the Stratix III device calculates the CRC value based on the frame of data that is received and compares it against the frame CRC value in the data stream. Configuration continues until either the device detects an error or configuration is complete.

In Stratix III devices, the CRC value is calculated during the configuration stage. A parallel CRC engine generates 16 CRC check bits per frame and stores them into CRAM. The CRAM chain used for storing CRC check bits is 16 bits wide; its length is equal to the number of frames in the device.

User Mode Error Detection

Stratix III devices have built-in error detection circuitry to detect data corruption by soft errors in the CRAM cells. This feature allows all CRAM contents to be read and verified to match a configuration-computed CRC value. Soft errors are changes in a CRAM's bit state due to an ionizing particle.

The error detection capability continuously computes the CRC of the configured CRAM bits and compares it with the pre-calculated CRC. If the CRCs match, there is no error in the current configuration CRAM bits. The process of error detection continues until the device is reset (by setting `nCONFIG` low).

As soon as the device transitions into user mode, you can enable the error detection process if you enable the CRC error detection option. The internal 100-MHz configuration oscillator is divided down by a factor of 2 to 256 (at powers of 2) to be used as the clock source during the error detection process. Set the clock divide factor in the option setting in the Quartus II software.

Thermal Resistance

For Stratix III devices thermal resistance specifications, refer to the *Stratix Series Device Thermal Resistance Data Sheet*.

Package Outlines

You can download Stratix III device package outlines from the *Device Packaging Specifications* web page.

Chapter Revision History

Table 17–2 lists the revision history for this chapter.

Table 17–2. Chapter Revision History

Date	Version	Changes Made
March 2010	1.7	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated Table 17–1. ■ Minor text edits.
February 2009	1.6	Removed "Referenced Documents" section.
October 2008	1.5	Updated New Document Format.
May 2008	1.4	Updated "Package Outlines" section hyperlink.
November 2007	1.3	Updated Table 17–1.
October 2007	1.2	■ Added new section "Referenced Documents". ■ Added live links for references.
May 2007	1.1	Removed thermal resistance and package outline information and replaced with links referencing this information.
November 2006	1.0	Initial Release.