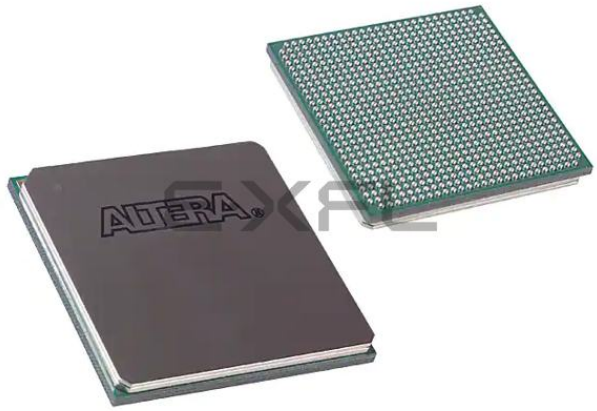


Welcome to E-XFL.COM

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	3200
Number of Logic Elements/Cells	80000
Total RAM Bits	6843392
Number of I/O	488
Number of Gates	-
Voltage - Supply	0.86V ~ 1.15V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	780-BBGA, FCBGA
Supplier Device Package	780-FBGA (29x29)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep3se80f780c3n

The design security feature is available when configuring Stratix III FPGAs using the fast passive parallel (FPP) configuration mode with an external host (such as a MAX II device or microprocessor), or when using fast active serial (AS) or passive serial (PS) configuration schemes.

For more information about the design security feature, refer to the *Design Security in Stratix III Devices* chapter.

SEU Mitigation

Stratix III devices have built-in error detection circuitry to detect data corruption due to soft errors in the configuration random-access memory (CRAM) cells. This feature allows all CRAM contents to be read and verified continuously during user mode operation to match a configuration-computed CRC value. The enhanced CRC circuit and frame-based configuration architecture allows detection and location of multiple, single, and adjacent bit errors which, in conjunction with a soft circuit supplied as a reference design, allows don't-care soft errors in the CRAM to be ignored during device operation. This provides a steep decrease in the effective soft error rate, increasing system reliability.

On-chip memory block SEU mitigation is also offered using the ninth bit and a configurable megafunction in the Quartus II software for MLAB and M9K blocks while the M144K memory blocks have built-in error correction code (ECC) circuitry.

For more information about the dedicated error detection circuitry, refer to the *SEU Mitigation in Stratix III Devices* chapter.

Programmable Power

Stratix III delivers Programmable Power, the only FPGA with user programmable power options balancing today's power and performance requirements. Stratix III devices utilize the most advanced power-saving techniques, including a variety of process, circuit, and architecture optimizations and innovations. In addition, user controllable power reduction techniques provide an optimal balance of performance and power reduction specific for each design configured into the Stratix III FPGA. The Quartus II software (starting from version 6.1) automatically optimizes designs to meet the performance goals while simultaneously leveraging the programmable power-saving options available in the Stratix III FPGA without the need for any changes to the design flow.

For more information about Programmable Power in Stratix III devices, refer to the following documents:

- *Programmable Power and Temperature Sensing Diode in Stratix III Devices* chapter
- *AN 437: Power Optimization in Stratix III FPGAs*
- *Stratix III Programmable Power White Paper*

C12 column interconnects span a length of 12 LABs and provide the fastest resource for column connections between distant LABs, TriMatrix memory blocks, DSP blocks, and IOEs. C12 interconnects drive LAB local interconnects via C4 and R4 interconnects and do not drive LAB local interconnects directly.

All embedded blocks communicate with the logic array through interconnects similar to LAB-to-LAB interfaces. Each block (for example, TriMatrix memory blocks and DSP blocks) connects to row and column interconnects and has local interconnect regions driven by row and column interconnects. These blocks also have direct link interconnects for fast connections to and from a neighboring LAB.

Table 3–1 shows the Stratix III device's routing scheme.

Table 3–1. Stratix III Device Routing Scheme

Source	Destination															
	Shared Arithmetic Chain	Carry Chain	Register Chain	Local Interconnect	Direct Link Interconnect	R4 Interconnect	R20 Interconnect	C4 Interconnect	C12 Interconnect	ALM	MLAB RAM Block	M9K RAM Block	M144K Block	DSP Blocks	Column IOE	Row IOE
Shared arithmetic chain	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
Carry chain	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
Register chain	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
Local interconnect	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
Direct link interconnect	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
R4 interconnect	—	—	—	✓	—	✓	✓	✓	✓	—	—	—	—	—	—	—
R20 interconnect	—	—	—	✓	—	✓	✓	✓	✓	—	—	—	—	—	—	—
C4 interconnect	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—	—
C12 interconnect	—	—	—	✓	—	✓	✓	✓	✓	—	—	—	—	—	—	—
ALM	✓	✓	✓	✓	✓	✓	—	✓	—	—	—	—	—	—	—	—
MLAB RAM block	—	—	—	✓	✓	✓	—	✓	—	—	—	—	—	—	—	—
M9K RAM block	—	—	—	—	✓	✓	—	✓	—	—	—	—	—	—	—	—
M144K block	—	—	—	—	✓	✓	—	✓	—	—	—	—	—	—	—	—
DSP blocks	—	—	—	—	✓	✓	—	✓	—	—	—	—	—	—	—	—
Column IOE	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—
Row IOE	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—

Notes to Table 3–1:

(1) Except column IOE local interconnects.
(2) Row IOE local interconnects.
(3) Column IOE local interconnects.

Clocking Modes

Stratix III TriMatrix memory blocks support the following clocking modes:

- Independent
- Input/output
- Read/write
- Single clock

Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

Altera recommends using a memory block clock that comes through global clock routing from an on-chip PLL set to 50% output duty cycle to achieve the maximum memory block performance. Use Quartus II to report timing for this and other memory block clocking schemes.

For more information refer to the *Stratix III Device Family Errata Sheet*.

Table 4–9 shows the clocking mode versus memory mode support matrix.

Table 4–9. Stratix III TriMatrix Memory Clock Modes

Clocking Mode	True Dual-Port Mode	Simple Dual-Port Mode	Single-Port Mode	ROM Mode	FIFO Mode
Independent	✓	—	—	✓	—
Input/output	✓	✓	✓	✓	—
Read/write	—	✓	—	—	✓
Single clock	✓	✓	✓	✓	✓

Independent Clock Mode

Stratix III TriMatrix memory blocks can implement independent clock mode for true dual-port memories. In this mode, a separate clock is available for each port (A and B). Clock A controls all registers on the port A side, while clock B controls all registers on the port B side. Each port also supports independent clock enables for port A and port B registers. Asynchronous clears are supported only for output latches and output registers on both ports.

Input/Output Clock Mode

Stratix III TriMatrix memory blocks can implement input/output clock mode for true and simple dual-port memories. In this mode, an input clock controls all registers related to the data input to the memory block, including data, address, byte-enables, read enables, and write enables. An output clock controls the data output registers. Asynchronous clears are available on output latches and output registers only.

Table 5–5. Examples of Shift Operations *(Note 1)*

Example	Signa	Signb	Shift_right	Rotate	A-input	B-input	Result
Logical Shift Left LSL[N]	Unsigned	Unsigned	0	0	0xAABBCCDD	0x00000100	0xBBCCDD00
Logical Shift Right LSR[32-N]	Unsigned	Unsigned	1	0	0xAABBCCDD	0x00000100	0x000000AA
Arithmetic Shift Left ASL[N]	Signed	Unsigned	0	0	0xAABBCCDD	0x00000100	0xBBCCDD00
Arithmetic Shift Right ASR[32-N]	Signed	Unsigned	1	0	0xAABBCCDD	0x00000100	0xFFFFFFAA
Rotation ROT[N]	Unsigned	Unsigned	0	1	0xAABBCCDD	0x00000100	0xBBCCDDAA

Note to Table 5–5:

(1) The value of the shift is equal to the value in the bracket where [N] is the position of bit '1' on the B-Input. In the above examples, [N] is 8 and is calculated from the LSB to the MSB where LSB=0 and MSB=31.

Rounding and Saturation Mode

Round and saturation functions are often required in DSP arithmetic. Rounding is used to limit bit growth and its side effects and saturation is used to reduce overflow and underflow side effects.

Two rounding modes are supported in Stratix III devices:

- Round-to-nearest-integer mode
- Round-to-nearest-even mode

You must select one of the two options at compile time.

Round-to-nearest-integer mode provides the biased rounding support and is the simplest form of rounding commonly used in DSP arithmetic. The round-to-nearest-even method provides unbiased rounding support and is used where DC offsets are a concern. Table 5–6 lists how round-to-nearest-even mode works. Examples of the difference between the two modes are listed in Table 5–7. In this example, a 6-bit input is rounded to 4 bits. You can observe from Table 5–7 that the main difference between the two rounding options is when the residue bits are exactly half way between its nearest two integers and the LSB is zero (even).

Table 5–6. Example of Round-To-Nearest-Even Mode

6- to 4-bits Rounding	Odd/Even (Integer)	Fractional	Add to Integer	Result
010111	x	> 0.5 (11)	1	0110
001101	x	< 0.5 (01)	0	0011
001010	Even (0010)	= 0.5 (10)	0	0010
001110	Odd (0011)	= 0.5 (10)	1	0100
110111	x	> 0.5 (11)	1	1110
101101	x	< 0.5 (01)	0	1011
110110	Odd (1101)	= 0.5 (10)	1	1110
110010	Even (1100)	= 0.5 (10)	0	1100

Phase-Shift Implementation

Phase shift is used to implement a robust solution for clock delays in Stratix III devices. Phase shift is implemented by using a combination of the VCO phase output and the counter starting time. The VCO phase output and counter starting time is the most accurate method of inserting delays, since it is based purely on counter settings, which are independent of process, voltage, and temperature.

You can phase-shift the output clocks from the Stratix III PLLs in either of these two resolutions:

- Fine resolution using VCO phase taps
- Coarse resolution using counter starting time

Fine-resolution phase shifts are implemented by allowing any of the output counters (`C[n..0]`) or the `m` counter to use any of the eight phases of the VCO as the reference clock. This allows you to adjust the delay time with a fine resolution. The minimum delay time that you can insert using this method is defined by Equation 6–1.

Equation 6–1.

$$\Phi_{fine} = \frac{1}{8} T_{VCO} = \frac{1}{8 f_{VCO}} = \frac{N}{8 M f_{REF}}$$

where f_{REF} is the input reference clock frequency.

For example, if f_{REF} is 100 MHz, n is 1, and m is 8, then f_{VCO} is 800 MHz and Φ_{fine} equals 156.25 ps. This phase shift is defined by the PLL operating frequency, which is governed by the reference clock frequency and the counter settings.

Coarse-resolution phase shifts are implemented by delaying the start of the counters for a predetermined number of counter clocks. You can express coarse phase shift as shown in Equation 6–2.

Equation 6–2.

$$\Phi_{coarse} = \frac{C - 1}{f_{VCO}} = \frac{(C - 1)N}{M f_{REF}}$$

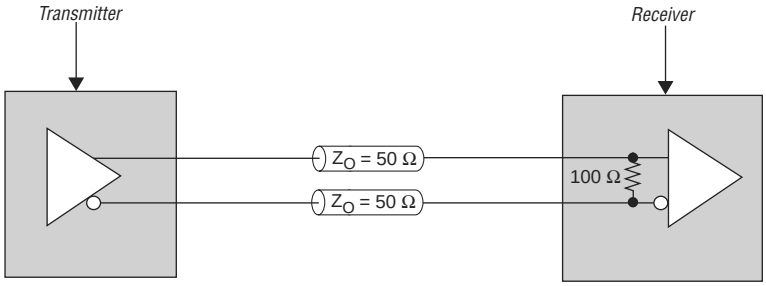
where C is the count value set for the counter delay time (this is the initial setting in the PLL usage section of the compilation report in the Quartus II software). If the initial value is 1, C – 1 = 0° phase shift.

Figure 6–39 shows an example of phase-shift insertion with the fine resolution using the VCO phase taps method. The eight phases from the VCO are shown and labeled for reference. For this example, `CLK0` is based off the `0phase` from the VCO and has the C value for the counter set to one. The `CLK0` signal is divided by four, two VCO clocks for high time and two VCO clocks for low time. `CLK1` is based off the 135° phase tap from the VCO and also has the C value for the counter set to one. The `CLK1` signal is also divided by 4. In this case, the two clocks are offset by $3\Phi_{fine}$. `CLK2` is based off the `0phase` from the VCO but has the C value for the counter set to three. This arrangement creates a delay of $2\Phi_{coarse}$ (two complete VCO periods).

LVDS Input On-Chip Termination (R_D)

Stratix III devices support OCT for differential LVDS input buffers with a nominal resistance value of 10 Ω, as shown in Figure 7–12. You can enable OCT R_D in row I/O banks when V_{CCIO} and V_{CCPD} are set to 2.5 V. The column I/O banks do not support OCT R_D. The dedicated clock input pairs `CLK[1,3,8,10][p,n]`, `PLL_L[1,4]_CLK[p,n]`, and `PLL_R[1,4]_CLK[p,n]` on the row I/O banks of the Stratix III devices do not support OCT R_D. Dedicated clock input pairs `CLK[0,2,9,11][p,n]` on row I/O banks support OCT R_D. Dedicated clock input pairs `CLK[4,5,6,7][p,n]` and `CLK[12,13,14,15][p,n]` on column I/O banks do not support OCT R_D.

Figure 7–12. Differential Input On-Chip Termination



For more information about OCT R_D, refer to the *High Speed Differential I/O Interfaces with DPA in Stratix III Devices* chapter.

Table 7–11 lists the assignment name and its value for OCT R_D in the Quartus II software Assignment Editor.

You must set the V_{CCIO} to 2.5 V when OCT R_D is used for the LVDS input buffer, even if the LVDS input buffer is powered by V_{CCPD}.

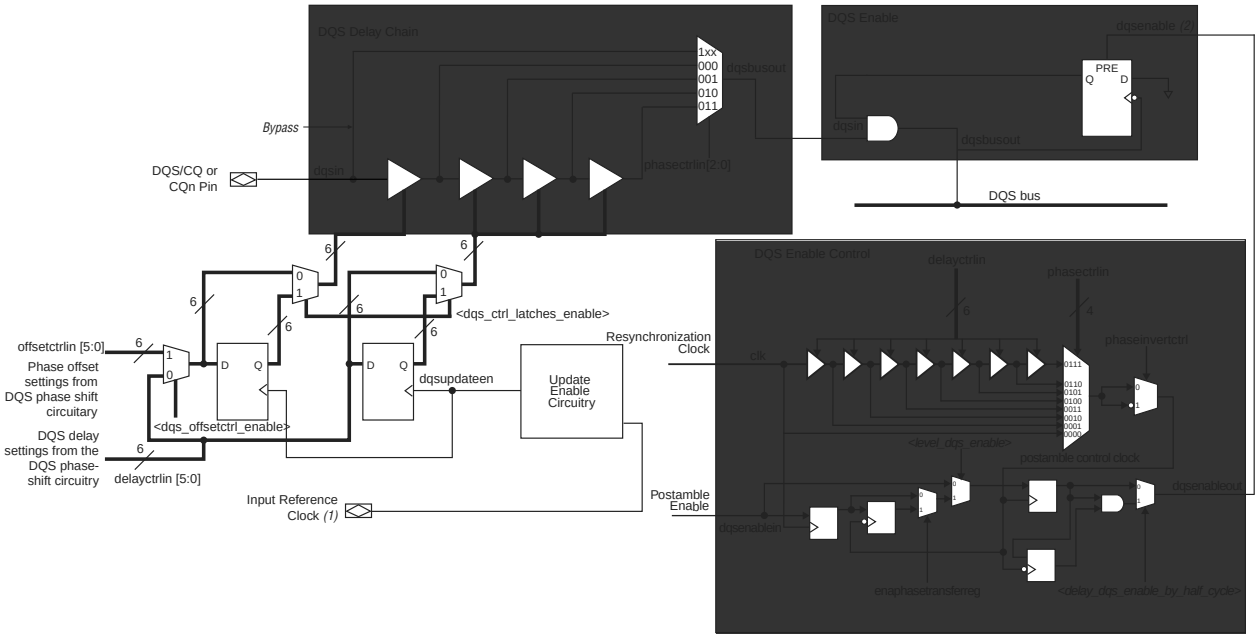
Table 7–11. On-Chip Differential Termination in Quartus II Software Assignment Editor

Assignment Name	Allowed Values	Applies To
Input Termination (Accepts wildcards/groups)	Parallel 50 Ω with calibration	Input buffers for single-ended and differential-HSTL/SSTL standards
	Differential	Input buffers for LVDS receivers on row I/O banks.
Output Termination	Series 25 Ω without calibration	Output buffers for single-ended LVTTL/LVCMOS and HSTL/SSTL standards as well as differential HSTL/SSTL standards.
	Series 50 Ω without calibration	
	Series 25 Ω with calibration	
	Series 40 Ω with calibration	
	Series 50 Ω with calibration	
	Series 60 Ω with calibration	

DQS Logic Block

Each DQS and CQn pin is connected to a separate DQS logic block, which consists of the DQS delay chains, update enable circuitry, and DQS postamble circuitry, as shown in Figure 8–13.

Figure 8–13. Stratix III DQS Logic Block



Notes to Figure 8–13:

(1) The input reference clock for the DQS phase-shift circuitry can come from a PLL output clock or an input clock pin. For the exact PLL and input clock pin location, refer to Table 8–6 through Table 8–9.

(2) The `dqsenable` signal can also come from the Stratix III FPGA fabric.

DQS Delay Chain

The DQS delay chains consist of a set of variable delay elements to allow the input DQS/CQ and CQn signals to be shifted by the amount specified by the DQS phase-shift circuitry or the logic array. There are four delay elements in the DQS delay chain; the first delay chain closest to the DQS/CQ pin can either be shifted by the DQS delay settings or by the sum of the DQS delay setting and the phase-offset setting. The number of delay chains required is transparent because the ALTMEMPHY megafunction automatically sets it when you choose the operating frequency. The DQS delay settings can come from the DQS phase-shift circuitry on either end of the I/O banks or from the logic array.

Delay elements in the DQS logic block have the same characteristics as the delay elements in the DLL. When the DLL is not used to control the DQS delay chains, you can input your own Gray-coded 6-bit or 5-bit settings using the `dqs_delayctrlin[5..0]`signals available in the ALTMEMPHY megafunction. These settings control 1, 2, 3, or all 4 delay elements in the DQS delay chains. The ALTMEMPHY megafunction can also dynamically choose the number of DQS delay chains required for the system. The amount of delay is equal to the sum of the delay element's intrinsic delay and the product of the number of delay steps and the value of the delay steps.

The output path is designed to route combinatorial or registered single data rate (SDR) outputs and full-rate or half-rate DDR outputs from the FPGA core. Half-rate data is converted to full-rate using the HDR block and is clocked by the half-rate clock from the PLL. Resynchronization registers are also clocked by the same 0° system clock, except in the DDR3 SDRAM interface where the leveling registers are clocked by the write-leveling clock.

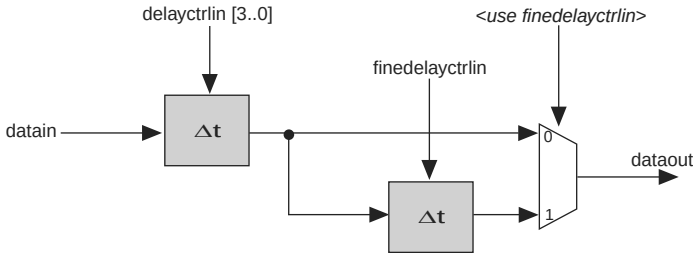
For more information about the write leveling delay chain, refer to "Leveling Circuitry" on page 8–31.

The output-enable path has structure similar to the output path. You can have a combinatorial or registered output in SDR applications and you can use half-rate or full-rate operation in DDR applications. You also have the resynchronization registers similar to the output path registers structure, ensuring that the output-enable path goes through the same delay and latency as the output path.

Delay Chain

Stratix III devices have run-time adjustable delay chains in the I/O blocks and the DQS logic blocks. You can control the delay chain setting through the I/O or the DQS configuration block output. Figure 8–22 shows the delay chain ports.

Figure 8–22. Delay Chain



Every I/O block contains the following:

- Two delay chains in series between the output registers and output buffer
- One delay chain between the input buffer and input register
- Two delay chains between the output enable and output buffer
- Two delay chains between the OCT R_T enable control register and output buffer

Table 8–13. Chapter Revision History (Part 2 of 2)

Date and Revision	Changes Made	Summary of Changes
November 2007, version 1.3	■ Updated Table 8–5. ■ Updated Figure 8–6.	Minor updates to content.
October 2007, version 1.2	■ Updated Table 8–1, Table 8–3, Table 8–4, Table 8–5. ■ Added Table 8–2. ■ Minor text edits. ■ Updated Figure 8–3, note 3 to Figure 8–4, note 3 to Figure 8–5, note 2 to Figure 8–6, added a note to Figure 8–7, added a note and updated Figure 8–10, notes to Figure 8–11, and updated Figure 8–12. ■ Added new material to “Memory Clock Pins” on page 8–21. ■ Added section “Referenced Documents”. ■ Added live links for references.	Minor updates to content.
May 2007, version 1.1	■ Updated Figure 8–5, Figure 8–8, Figure 8–14, Figure 8–18, Figure 8–19, Figure 8–20, and Figure 8–21. ■ Added new figure, Figure 8–17. ■ Added memory support information for -4L in Table 8–1, Table 8–8, Table 8–10, and Table 8–11. ■ Added new material to section “Phase Offset Control” on page 8–32.	Minor updates to content.
November 2006, version 1.0	Initial Release.	—

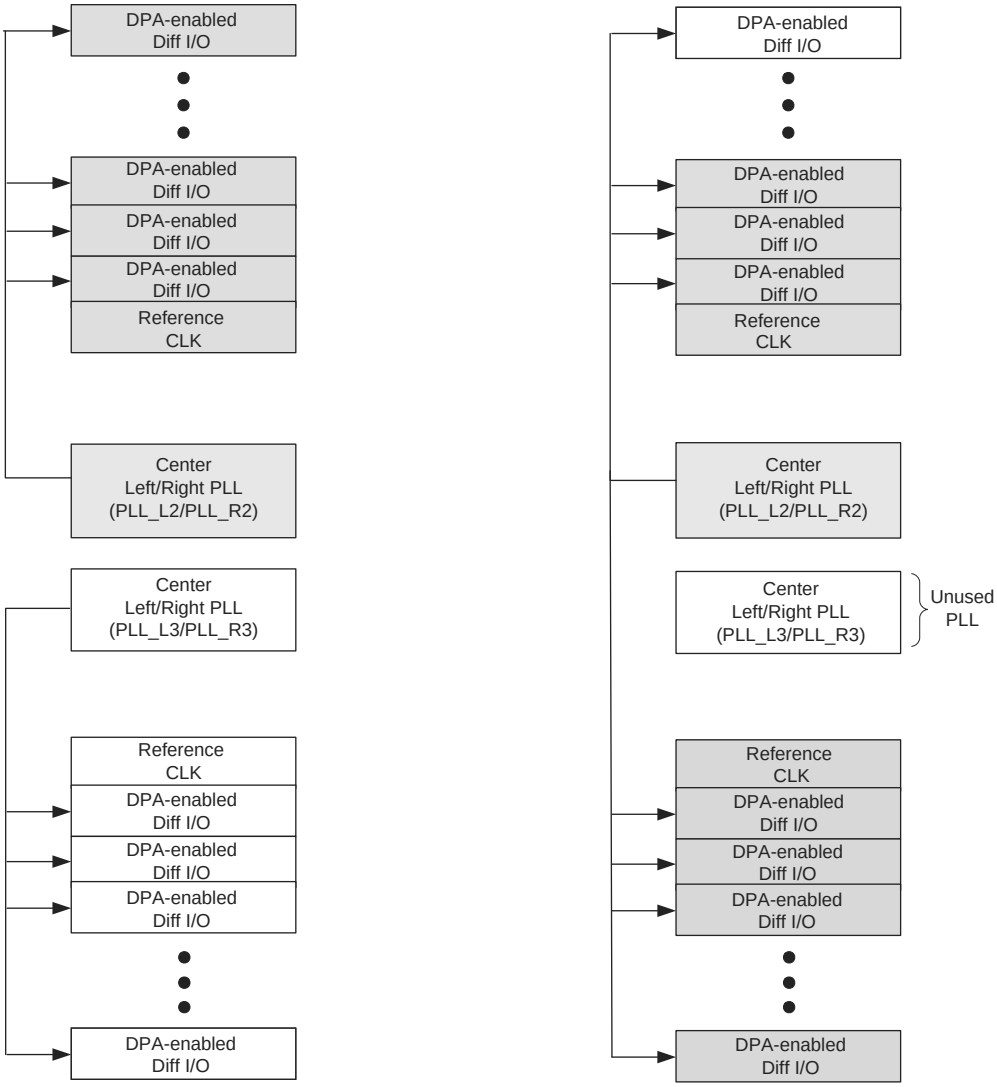
Using Both Center Left/Right PLLs

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



Guidelines for DPA-Disabled Differential Channels

When DPA-disabled channels are used in the left and right banks of a Stratix III device, you must adhere to the guidelines in the following sections.

DPA-Disabled Channels and Single-Ended I/Os

The placement rules for DPA-disabled channels and single-ended I/Os are the same as those for DPA-enabled channels and single-ended I/Os.

DPA-Disabled Channel Driving Distance

Each left/right PLL can drive all the DPA-disabled channels in the entire bank.

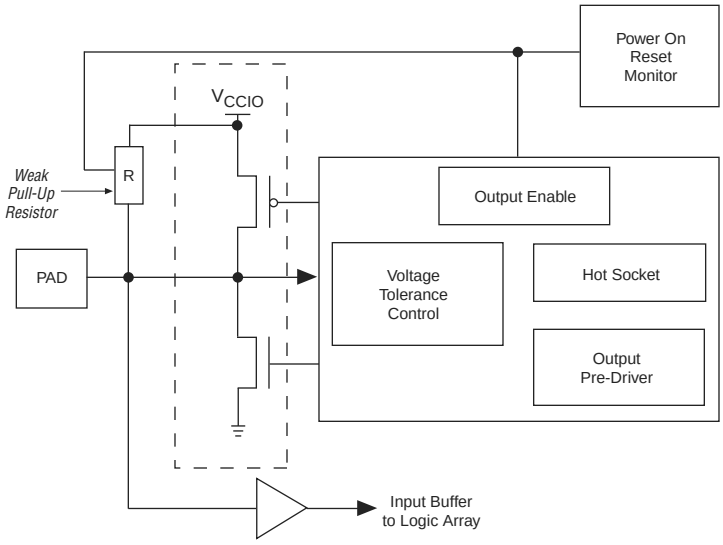
Using Corner and Center Left/Right PLLs

A corner left/right PLL can be used to drive all transmitter channels and a center left/right PLL can be used to drive all DPA-disabled receiver channels within the same differential bank. In other words, a transmitter channel and a receiver channel in the same LAB row can be driven by two different PLLs, as shown in Figure 9–21.

A corner left/right PLL and a center left/right PLL can drive duplex channels in the same differential bank as long as the channels driven by each PLL are not interleaved. No separation is necessary between the group of channels driven by the corner and center left/right PLLs. Refer to Figure 9–21 and Figure 9–22.

Figure 10–1 shows the Stratix III device's I/O pin circuitry.

Figure 10–1. Hot-Socketing Circuitry for Stratix III Devices



The POR circuit monitors the voltage level of power supplies (V_{CC}, V_{CCL}, V_{CCPD}, V_{CCPGM} and V_{CCPT}) and keeps the I/O pins tri-stated until the device is in user mode. The weak pull-up resistor (R) in the Stratix III input/output element (IOE) keeps the I/O pins from floating. The 3.3-V tolerance control circuit permits the I/O pins to be driven by 3.3 V before V_{CCIO}, V_{CC}, V_{CCPD}, and/or V_{CCPGM} supplies are powered, and it prevents the I/O pins from driving out when the device is not in user mode.

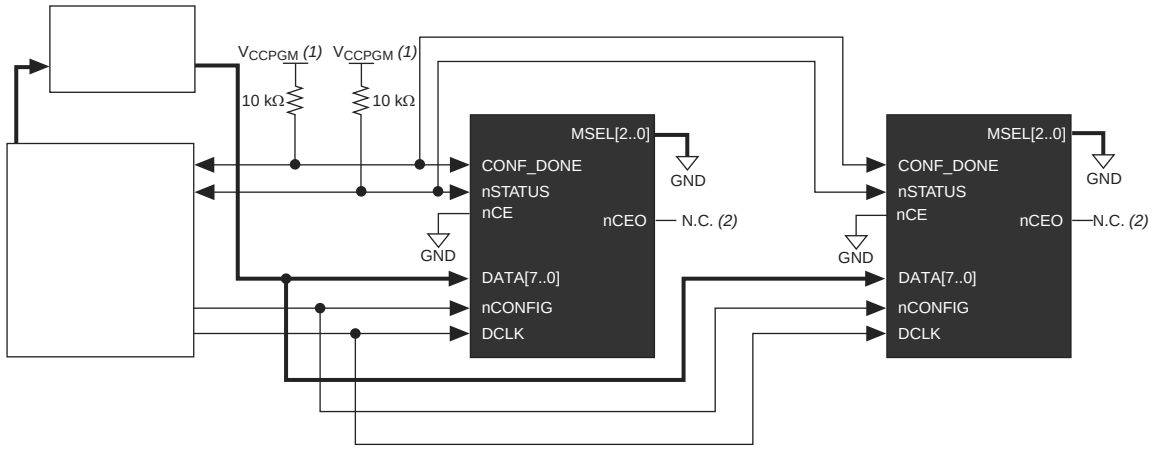
Altera uses GND as reference for hot-socketing operation and I/O buffer designs. To ensure proper operation, you must connect the GND between boards before connecting the power supplies. This will prevent the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled up GND could otherwise cause an out-of-specification I/O voltage or current condition with the Altera device.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

In a multi-device FPP configuration chain, all Stratix III devices in the chain must either enable or disable the decompression feature, design security feature, or both. You cannot selectively enable the decompression feature, design security feature, or both for each device in the chain because of the DATA and DCLK relationship. If the chain contains devices that do not support design security, you should use a serial configuration scheme.

If a system has multiple devices that contain the same configuration data, tie all device nCE inputs to GND, and leave nCEO pins floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices start and complete configuration at the same time. Figure 11–5 shows a multi-device FPP configuration when both Stratix III devices are receiving the same configuration data.

Figure 11–5. Multiple-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 11–5:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix III devices on the chain. V_{CCPGM} should be high enough to meet the V_{IH} specification of the I/O on the external host. It is recommended to power up all configuration system's I/O with V_{CCPGM}.

(2) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

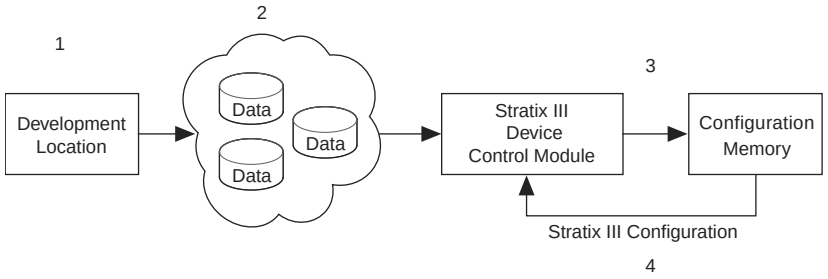
You can use a single configuration chain to configure Stratix III devices with other Altera devices that support FPP configuration, such as other types of Stratix devices. To ensure that all devices in the chain complete configuration at the same time, or that an error flagged by one device initiates reconfiguration in all devices, tie all of the device CONF_DONE and nSTATUS pins together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to *Configuring Mixed Altera FPGA Chains* in the *Configuration Handbook*.

3. The Nios II processor (or user logic) initiates a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any error(s) that might occur during or after the reconfiguration cycle, and provides error status information to the user design.

Figure 12–1 shows the steps required for performing remote configuration updates. (The numbers in the figure below coincide with the steps above.)

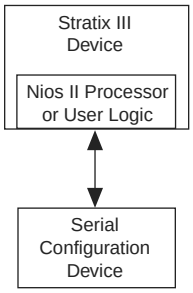
Figure 12–1. Functional Diagram of Stratix III Remote System Upgrade



Stratix III devices only support remote system upgrade in the single device Fast AS configuration scheme.

Figure 12–2 shows the block diagrams for implementing a remote system upgrade with the Stratix III Fast AS configuration scheme.

Figure 12–2. Remote System Upgrade Block Diagram for Stratix III Fast AS Configuration Scheme



You must set the mode select pins (`MSEL[2..0]`) to Fast AS mode to use the remote system upgrade in your system. Table 12–1 lists the `MSEL` pin settings for Stratix III devices in standard configuration mode and remote system upgrade mode. The following sections describe the remote update of remote system upgrade mode.

For more information about standard configuration schemes supported in Stratix III devices, refer to the *Configuring Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

Table 12–2. Remote System Upgrade Registers (Part 2 of 2)

Register	Description
Update register	Contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a capture in a factory configuration, this register is read into the shift register.
Status register	Written to by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

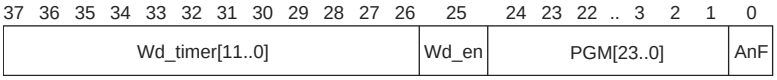
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (RU_CLK).

Remote System Upgrade Control Register

The remote system upgrade control register stores the application configuration page address and user watchdog timer settings. The control register functionality depends on the remote system upgrade mode selection. In remote update mode, the control register page address bits are set to all zeros (24'b0 = 0×000000) at power up in order to load the factory configuration. A factory configuration in remote update mode has write access to this register.

The control register bit positions are shown in Figure 12–6 and defined in Table 12–3. In the figure, the numbers show the bit position of a setting within a register. For example, bit number 8 is the enable bit for the watchdog timer.

Figure 12–6. Remote System Upgrade Control Register



The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix III device is the factory configuration or an application configuration. This bit is set low by the remote system upgrade circuitry when an error condition causes a fall-back to the factory configuration. When the AnF bit is high, the control register access is limited to read operations. When the AnF bit is low, the register allows write operations and disables the watchdog timer.

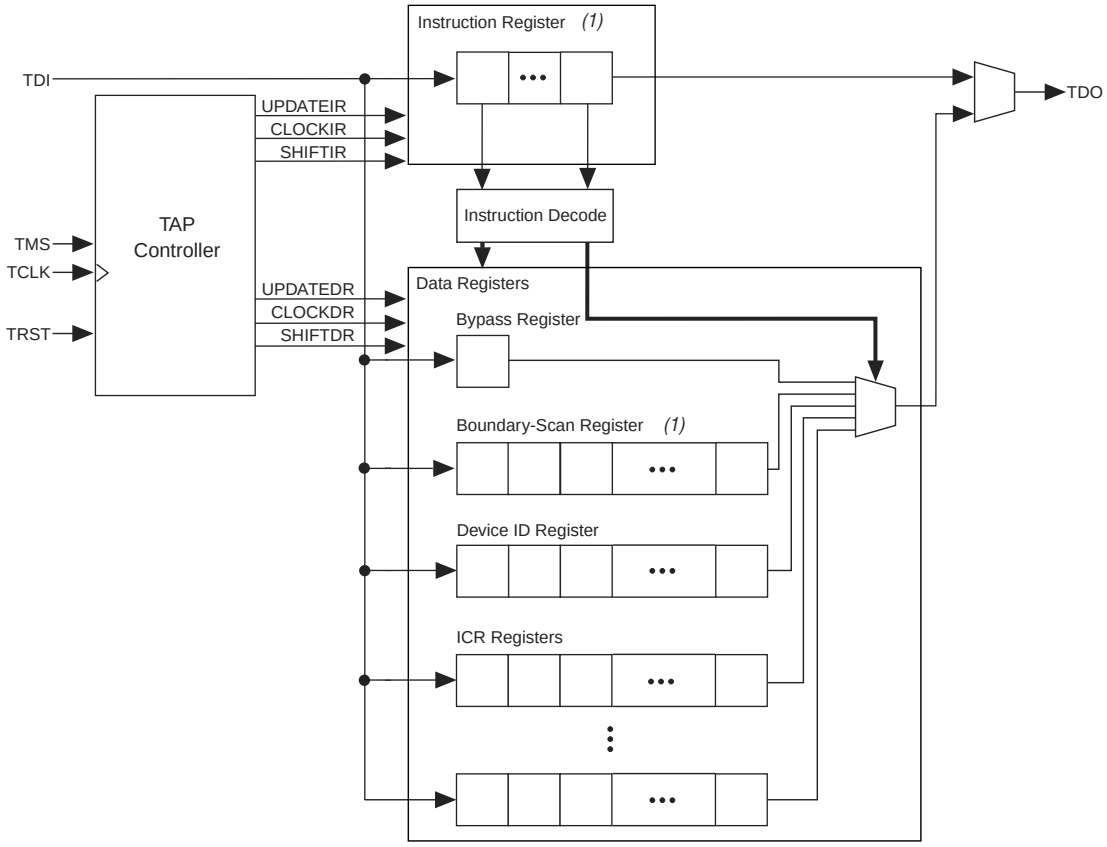
In remote update mode, factory configuration design sets this bit high (1'b1) when updating the contents of the update register with the application page address and watchdog timer settings.

Table 12–3. Remote System Upgrade Control Register Contents (Part 1 of 2)

Control Register Bit	Remote System Upgrade Mode	Value *(2)*	Definition
AnF *(1)*	Remote update	1'b0	Application not factory
PGM[23..0]	Remote update	24'b0×000000	AS configuration start address (StAdd[23..0])

Figure 13–2 shows a functional model of the IEEE Std. 1149.1 circuitry.

Figure 13–2. IEEE Std. 1149.1 Circuitry



Note to Figure 13–2:

(1) For register lengths, refer to the device datasheet in the *Configuring Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

IEEE Std. 1149.1 boundary-scan testing is controlled by a TAP controller. For more information about the TAP controller, refer to "IEEE Std. 1149.1 BST Operation Control" on page 13–7. The TMS and TCK pins operate the TAP controller. The TDI and TDO pins provide the serial path for the data registers. The TDI pin also provides data to the instruction register, which then generates control logic for the data registers.

Table 13–3 lists the capture and update register capabilities of all boundary-scan cells within Stratix III devices.

Table 13–3. Stratix III Device Boundary Scan Cell Descriptions *(Note 1)*

Pin Type	Captures			Drives			Comments
	Output Capture Register	OE Capture Register	Input Capture Register	Output Update Register	OE Update Register	Input Update Register	
User I/O pins	OUTJ	OEJ	PIN_IN	PIN_OUT	PIN_OE	INJ	—
Dedicated clock input	0	1	PIN_IN	—	—	—	PIN_IN drives to clock network or logic array
Dedicated input *(2)*	0	1	PIN_IN	—	—	—	PIN_IN drives to control logic
Dedicated bi-directional (open drain) *(3)*	0	OEJ	PIN_IN	—	—	—	PIN_IN drives to configuration control
Dedicated bi-directional *(4)*	OUTJ	OEJ	PIN_IN	—	—	—	PIN_IN drives to configuration control and OUTJ drives to output buffer
Dedicated output *(5)*	OUTJ	0	0	—	—	—	OUTJ drives to output buffer

Notes to Table 13–3:

(1) TDI, TDO, TMS, TCK, TRST, all V_{CC} and GND pin types, VREF, and TEMP_DIODE pins do not have BSCs.
(2) This includes pins PLL_ENA, nCONFIG, MSEL0, MSEL1, MSEL2, nCE, PORSEL, and nIO_PULLUP.
(3) This includes pins CONF_DONE and nSTATUS.
(4) This includes pin DCLK.
(5) This includes pin nCEO.

15. SEU Mitigation in Stratix III Devices

SIII51015-1.7

This chapter describes how to use the error detection cyclical redundancy check (CRC) feature when a Stratix® III device is in user mode and recovers from CRC errors. The purpose of the error detection CRC feature is to detect a flip in any of the configuration CRAM bits in Stratix III devices due to a soft error. By using the error detection circuitry, you can continuously verify the integrity of the configuration CRAM bits.

In critical applications such as avionics, telecommunications, system control, and military applications, it is important to be able to do the following:

- Confirm that the configuration data stored in a Stratix III device is correct.
- Alert the system to the occurrence of a configuration error.

The error detection feature has been enhanced in the Stratix III device family. In addition, the error detection and recovery time for single event upset (SEU) in Stratix III devices is reduced compared to Stratix II devices.

For Stratix III devices, use of the error detection CRC feature is provided in the Quartus® II software version 6.1 and onwards.

Stratix III devices only support the error detection CRC feature at 1.1 V for V_{CCL}. This feature is not supported in Stratix III devices operating at 0.9 V for V_{CCL}.

Dedicated circuitry is built into Stratix III devices and consists of a CRC error detection feature that can optionally check for SEUs continuously and automatically.

This section describes how to activate and use the error detection CRC feature when your Stratix III device is in user mode and describes how to recover from configuration errors caused by CRC errors.

Information about SEU is located on the Products page of the Altera® website at www.altera.com.

For more information regarding the test methodology for the enhanced error detection in Stratix III, refer to *AN 539: Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices*.

For more information, refer to the *Robust SEU Mitigation with Stratix III FPGAs White Paper*.

Using CRC error detection for the Stratix III family has no impact on fitting or performance of your device.

Lowering the core voltage reduces both static and dynamic power, but causes a reduction in performance. You need to set the correct core supply voltage in the Quartus II software settings under **Operating Conditions**, since the Quartus II software analyzes the core power consumption and timing delays based on this selection. When you compile a design, you can select either 0.9-V or 1.1-V core voltage. You can compare the power and performance trade-offs of a 0.9-V core voltage compilation result and a 1.1-V core voltage compilation result and then choose the most desirable core voltage for your design. By default, the Quartus II software sets the core voltage to 1.1 V.

Ensure that the board has a separate 0.9-V power supply to utilize the lower voltage option and be sure to connect V_{CCL} to the voltage level that you set in the Quartus II software. The Stratix III device cannot distinguish which core voltage level is used on the board. Connecting to the wrong voltage level gives you different timing delays and power consumption than what is reported by the Quartus II software.

For information about selectable core voltage performance and power effects on sample designs, refer to *AN 437: Power Optimization Techniques*.

Programmable Power Technology

In addition to the variable core voltage, Stratix III devices also offer the ability to configure portions of the core, called tiles, for high-speed or low-power mode of operation performed by the Quartus II software without user intervention. This programmable power technology, used to reduce static power, uses an on-chip voltage regulator powered by V_{CCPT}. In a design compilation, the Quartus II software determines whether a tile needs to be in high-speed or low-power mode based on the timing constraints of the design.

For more information about how the Quartus II software uses programmable power technology when compiling a design, refer to *AN 437: Power Optimization Techniques*.

A Stratix III tile can consist of the following:

- MLAB/LAB pairs with routing to the pair
- MLAB/LAB pairs with routing to the pair and to adjacent DSP/memory block routing
- TriMatrix memory blocks
- DSP blocks
- I/O interfaces

All blocks and routing associated with the tile share the same setting of either high speed or low power. Tiles that include DSP blocks, memory blocks, or I/O interfaces are set to high-speed mode by default for optimum performance when used in the design. Unused DSP blocks, memory blocks, and I/O elements are set to low-power mode to minimize static power. Clock networks do not support programmable power technology.