Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

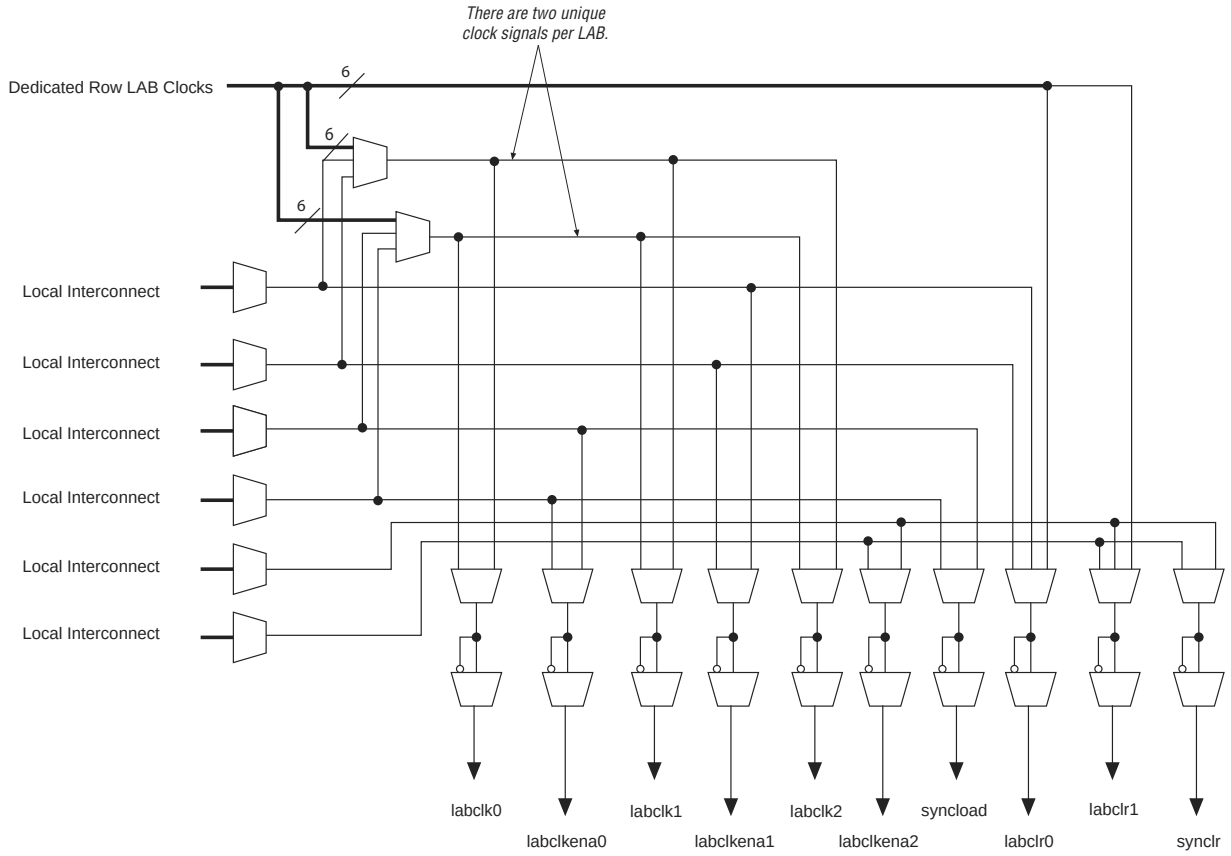
Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling 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	13500
Number of Logic Elements/Cells	337500
Total RAM Bits	18822144
Number of I/O	744
Number of Gates	-
Voltage - Supply	0.86V ~ 1.15V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	1152-BBGA, FCBGA
Supplier Device Package	1152-HBGA (40x40)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep3sl340h1152c3n

Figure 2–4. LAB-Wide Control Signals



Adaptive Logic Modules

The basic building block of logic in the Stratix III architecture, the adaptive logic module (ALM), provides advanced features with efficient logic utilization. Each ALM contains a variety of look-up table (LUT)-based resources that can be divided between two combinational adaptive LUTs (ALUTs) and two registers. With up to eight inputs to the two combinational ALUTs, one ALM can implement various combinations of two functions. This adaptability allows an ALM to be completely backward-compatible with four-input LUT architectures. One ALM can also implement any function of up to six inputs and certain seven-input functions.

In addition to the adaptive LUT-based resources, each ALM contains two programmable registers, two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. Through these dedicated resources, an ALM can efficiently implement various arithmetic functions and shift registers. Each ALM drives all types of interconnects: local, row, column, carry chain, shared arithmetic chain, register chain, and direct link interconnects. Figure 2–5 shows a high-level block diagram of the Stratix III ALM while Figure 2–6 shows a detailed view of all the connections in an ALM.

In simple dual-port mode, M9K and M144K blocks support separate write-enable and read-enable signals. You can save power by keeping the read-enable signal low (inactive) when not reading. Read-during-write operations to the same address can either output a don't care value or old data. To choose the desired behavior, set the read-during-write behavior to either don't care or old data in the RAM MegaWizard Plug-In Manager in the Quartus II software. See "Read During Write" on page 4–21 for more details about this behavior.

MLABs only support a write-enable signal. Read-during-write behavior for the MLABs can be either don't care, new data, or old data. The available choices depend on the configuration of the MLAB.

Figure 4–13 shows the timing waveforms for read and write operations in simple dual-port mode with unregistered outputs in M9K and M144K. Registering the RAM's outputs would simply delay the q output by one clock cycle in M9k and M144K.

Figure 4–13. Stratix III Simple Dual-Port Timing Waveforms for M9K and M144K

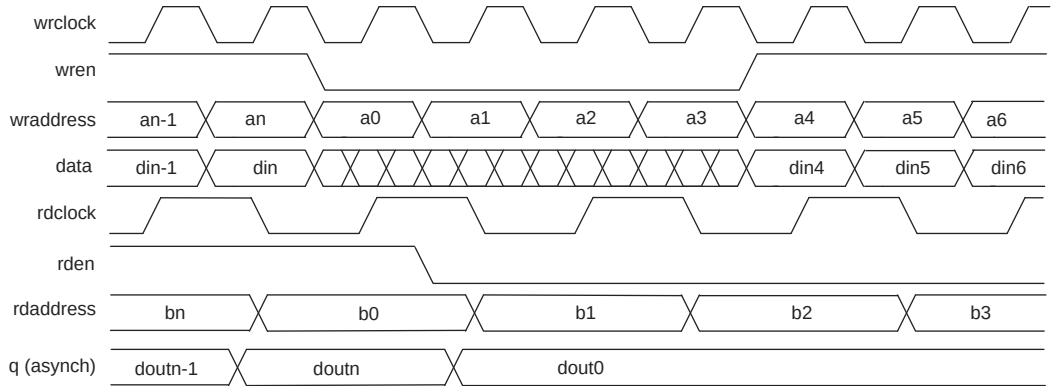
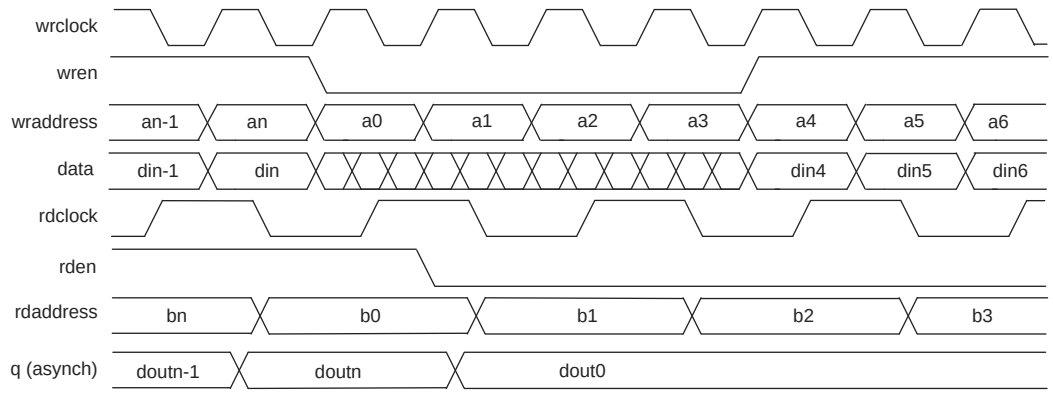


Figure 4–14 shows the timing waveforms for read and write operations in simple dual-port mode with unregistered outputs in MLABs. In MLABs, the write operation is triggered by the falling clock edges.

Figure 4–14. Stratix III Simple Dual-Port Timing Waveforms for MLABs



The structure shown in Figure 5–2 is very useful for building more complex structures, such as complex multipliers and 36 × 36 multipliers, as described in later sections.

Each Stratix III DSP block contains four Two-Multiplier Adder units (two Two-Multiplier Adder units per half-block). Therefore, there are eight 18 × 18 multiplier functionalities per DSP block.

Following the Two-Multiplier Adder units are the pipeline registers, the second-stage adders, and an output register stage. You can configure the second-stage adders to provide the following alternative functions per Half-Block:

Equation 5–2. Four-Multiplier Adder Equation

$$Z[37..0] = P_0[36..0] + P_1[36..0]$$

Equation 5–3. Four-Multiplier Adder Equation (44-Bit Accumulation)

$$W_n[43..0] = W_{n-1}[43..0] \pm Z_n[37..0]$$

In these equations, n denotes sample time, and P[36..0] are the results from the Two-Multiplier Adder units.

Equation 5–2 provides a sum of four 18-bit × 18-bit multiplication operations (Four-Multiplier Adder), and Equation 5–3 provides a four 18-bit × 18-bit multiplication operation but with maximum of a 44-bit accumulation capability by feeding the output of the unit back to itself. This is shown in Figure 5–3.

You can bypass all register stages depending on which mode you select.

High Precision Multiplier Adder

In the high precision multiplier adder configuration shown in Figure 5–18, the DSP block can implement two two-multiplier adders, with multiplier precision of 18 × 36 (one two-multiplier adder per DSP half block). This mode is useful in filtering or FFT applications where a data path greater than 18 bits is required, yet 18 bits is sufficient for the coefficient precision. This can occur in cases where that data has a high dynamic range. If the coefficients are fixed, as in FFT and most filter applications, the precision of 18 bits will provide a dynamic range over 100 dB if the largest coefficient is normalized to the maximum 18-bit representation.

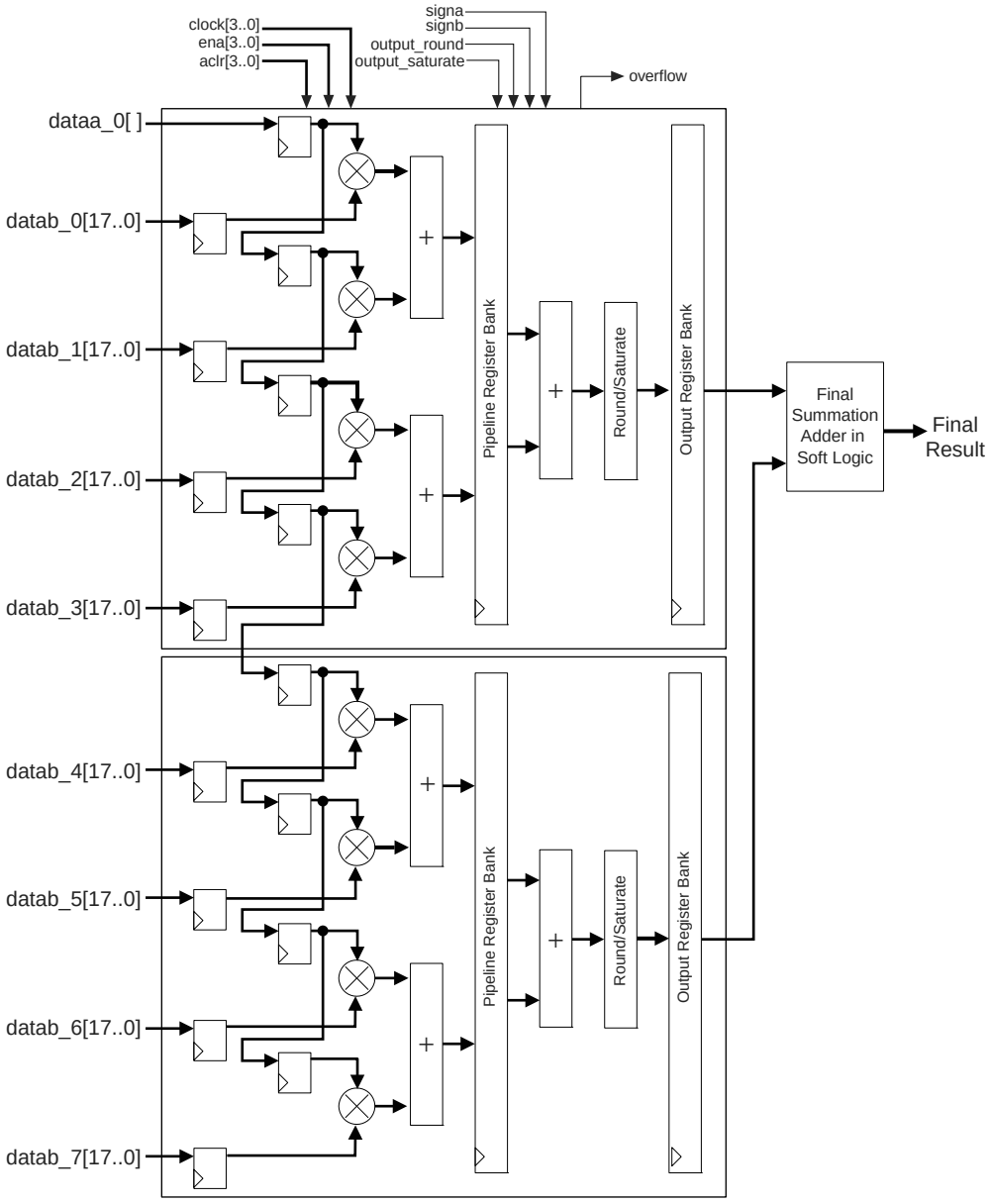
In these situations, the data path can be up to 36 bits, allowing ample headroom to bit growth, or gain changes in the signal source without loss of precision. This mode is also extremely useful in single precision block floating point applications.

The high precision multiplier adder is preformed in two stages. The 18 × 36 multiply is decomposed into two 18 × 18 multipliers. The multiplier with the LSB of the data source is performed unsigned, while the multiplier with the MSB of the data source can be signed or signed. The latter multiplier has its result left shifted by 18 bits prior to the first adder stage, creating an effective 18 × 36 multiplier. The results of these two adder blocks are then summed in the second stage adder block to produce the final result.

Equation 5–5. High Precision Multiplier Adder Equation

$$Z[54..0] = P_0[53..0] + P_1[53..0] \text{ where}$$
$$P_0 = A[17..0] \acute{} B[35..0] \text{ and } P_1 = C[17..0] \acute{} D[35..0]$$

Figure 5–22. FIR Filter Using Tap-Delay Line Input and Tree Summation of Final Result



For faster and more efficient chained cascade summation, the DSP block can implement the chainout function in the cascade mode. This mode uses the second-stage 44-bit adder to add the current Four-Multiplier Adder of the half DSP block to the adjacent half DSP block of the Four-Multiplier Adder as shown in Figure 5–23.

This scheme is possible because each half DSP block has two second-stage adders. One of the two second-stage adders is used to add the current Four-Multiplier Adder. The second second-stage adder takes the output of the first second-stage adder and adds it to the adjacent half DSP block of the Four-Multiplier Adder result.

Regional Clock Networks

The regional clock (RCLK) networks only pertain to the quadrant they drive into. The RCLK networks provide the lowest clock delay and skew for logic contained within a single device quadrant. Stratix III device I/O elements and internal logic within a given quadrant can also drive RCLKs to create internally generated regional clocks and other high fan-out control signals; for example, synchronous or asynchronous clears and clock enables. Figure 6–2 to Figure 6–4 show CLK pins and PLLs that can drive RCLK networks in Stratix III devices. The EP3SL50, EP3SL70, EP3SL110, EP3SL150, EP3SE50, EP3SE80, and EP3SE110 devices contain 64 RCLKs; the EP3SL200, EP3SE260, and EP3SL340 devices contain 88 RCLKs.

Figure 6–2. Regional Clock Networks (EP3SL50, EP3SL70, and EP3SE50 Devices)

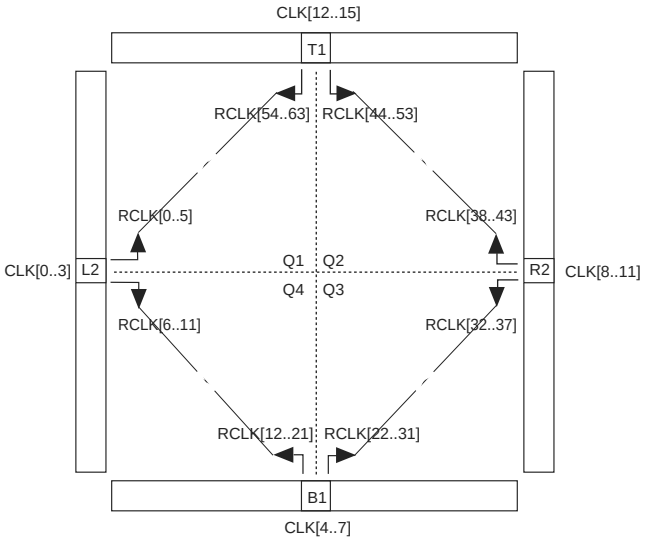
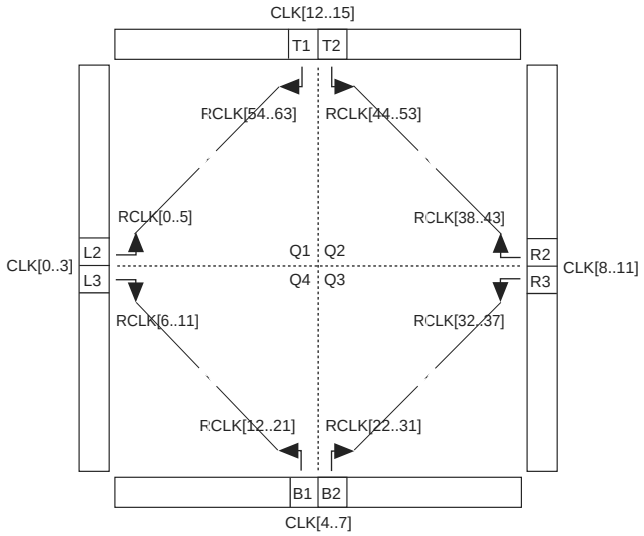


Figure 6–3. Regional Clock Networks (EP3SL110, EP3SL150, EP3SE80, and EP3SE110 Devices)



PLL Control Signals

You can use the following three signals to observe and control the PLL operation and resynchronization.

pfdena

Use the `pfdena` signal to maintain the most recent locked frequency so your system has time to store its current settings before shutting down. The `pfdena` signal controls the PFD output with a programmable gate. If you disable the PFD, the VCO is free running and the PLL output drifts. The PLL output jitter may not meet the datasheet specifications. The lock signal cannot be used as an indicator when the PFD is disabled.

areset

The `areset` signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals. When `areset` is driven high, the PLL counters reset, clearing the PLL output and placing the PLL out-of-lock. The VCO is then set back to its nominal setting. When `areset` is driven low again, the PLL will resynchronize to its input as it re-locks.

You should assert the `areset` signal every time the PLL loses lock to guarantee the correct phase relationship between the PLL input clock and output clocks. You can set up the PLL to automatically reset (self reset) upon a loss-of-lock condition using the Quartus II MegaWizard Plug-In Manager. You should include the `areset` signal in designs if the following condition is true:

- PLL reconfiguration or clock switchover is enabled in the design.

1 If the input clock to the PLL is not toggling or is unstable upon power up, assert the `areset` signal after the input clock is stable and within specifications.

locked

The lock signal is an asynchronous output of the PLL. The locked output of the PLL indicates that the PLL has locked onto the reference clock and the PLL clock outputs are operating at the desired phase and frequency set in the Quartus II MegaWizard Plug-In Manager. The lock detection circuit provides a signal to the core logic that gives an indication if the feedback clock has locked onto the reference clock both in phase and frequency.

1 Altera recommends that you use the `areset` and `locked` signals in your designs to control and observe the status of your PLL.

Clock Switchover

The clock switchover feature allows the PLL to switch between two reference input clocks. Use this feature for clock redundancy or for a dual-clock domain application such as in a system that turns on the redundant clock if the previous clock stops running. The design can perform clock switchover automatically, when the clock is no longer toggling or based on a user control signal, `clkswitch`.

Chapter Revision History

Table 6–23 lists the revision history for this chapter.

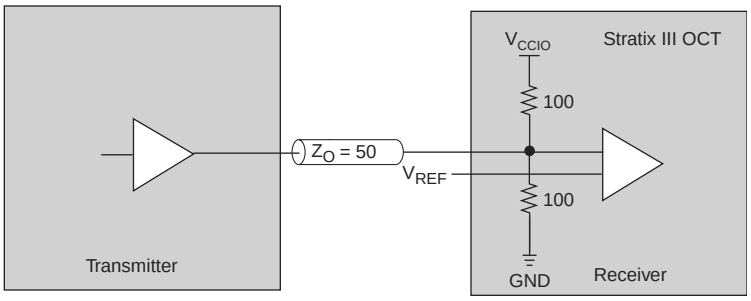
Table 6–23. Chapter Revision History (Part 1 of 2)

Date	Version	Changes Made
July 2010	2.0	Updated Figure 6–44.
March 2010	1.9	Updated for the Quartus II software version 9.1 SP2 release: Updated Table 6–10 and Table 6–11. Updated Figure 6–42. Updated the "Guidelines" and "PLL Cascading and Clock Network Guidelines" sections. Removed "sub-regional clock networks" information. Minor text edits.
July 2009	1.8	Updated "Clock Switchover" section. Updated Figure 6–37.
May 2009	1.7	Added "PLL and Clock Network Guidelines for External Memory Interface" and "Zero-Delay Buffer Mode" sections. Updated Figure 6–17.
February 2009	1.6	Updated Table 6–7 and Table 6–10. Updated Figure 6–23. Updated "PLL Clock I/O Pins", "Logic Array Blocks (LABs)", and "Clock Feedback Modes" sections. Removed "Reference Documents" section.
October 2008	1.5	Updated Table 6–10, Table 6–13, and Table 6–14. Updated "locked", "Manual Override", "Bypassing PLL", "PLL Clock I/O Pins", and "Dynamic Phase-Shifting" sections. Updated Figure 6–22, Figure 6–24, and Figure 6–26. Updated (Note 2) to Figure 6–22. Added (Note 3) to Table 6–14. Added Figure 6–27. Updated New Document Format.
May 2008	1.4	Updated Table 6–3, Table 6–4, Table 6–5, Table 6–6, Table 6–7, and Table 6–14. Added new Figure 6–5 through Figure 6–9 to "Periphery Clock Networks" section. Updated "Logic Array Blocks (LABs)", "External Feedback Mode", "Phase-Shift Implementation", and "Spread-Spectrum Tracking" sections. Updated notes to Figure 6–17. Updated notes to Figure 6–22. Updated notes to Figure 6–27. Updated Figure 6–43.
November 2007	1.3	Updated "pfdena" on page 6–42.

On-Chip Parallel Termination with Calibration

Stratix III devices support OCT R_T with calibration in all banks. OCT R_T with calibration is only supported for input or bi-directional pin configurations. For input pins, you can enable OCT R_T continuously. However, for bi-directional I/O, OCT R_T is enabled or disabled depending on whether or not the bi-directional I/O acts as a transmitter or receiver. Output pin configurations do not support OCT R_T with calibration. Figure 7–10 shows OCT R_T with calibration. When OCT R_T is used, the V_{CCIO} of the bank has to match the I/O standard of the pin where the parallel OCT is enabled.

Figure 7–10. On-Chip Parallel Termination with Calibration for Stratix III Devices



The OCT R_T calibration circuit compares the total impedance of the I/O buffer to the external 50- ±1% resistors connected to the RUP and RDN pins and dynamically enables or disables the transistors until they match. 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. Table 7–10 lists the I/O standards that support OCT R_T with calibration.

Table 7–10. Selectable I/O Standards that Support On-Chip Parallel Termination with Calibration

I/O Standard	On-Chip Parallel Termination Setting (Column I/O)	On-Chip Parallel Termination Setting (Row I/O)	Unit
SSTL-2 Class I, II	50	50	
SSTL-18 Class I, II	50	50	
SSTL-15 Class I, II	50	50	
HSTL-18 Class I, II	50	50	
HSTL-15 Class I, II	50	50	
HSTL-12 Class I, II	50	50	
Differential SSTL-2 Class I, II	50	50	
Differential SSTL-18 Class I, II	50	50	
Differential SSTL-15 Class I, II	50	50	
Differential HSTL-18 Class I, II	50	50	
Differential HSTL-15 Class I, II	50	50	
Differential HSTL-12 Class I, II	50	50	

Table 8–2. Number of DQS/DQ Groups in Stratix III Devices per Side (Part 2 of 2)

Device	Package	Side	×4 *(1)*	×8/×9	×16/×18	×32/×36 *(2)*
EP3SL340	1152-pin Hybrid FineLine BGA	Left/ Right	26	12	4	0
		Top/ Bottom	26	12	4	0
	1517-pin FineLine BGA	Left/ Right	34	16	6	0
		Top/ Bottom	38	18	8	4
	1760-pin FineLine BGA	Left/ Right	40	18	6	0
		Top/ Bottom	44	22	10	4

Notes to Table 8–2:

(1) Some of the ×4 groups may use configuration or RUP/RDN pins. You cannot use these ×4 groups if the pins are used for configuration or as RUP and RDN pins for OCT calibration.

(2) To interface with a ×36 QDR II+/QDR II SRAM device in a Stratix III FPGA that does not support the ×32/×36 DQS/DQ group, refer to the *Device, Pin, and Board Layout Guidelines* in volume 2 of the *External Memory Interface Handbook*.

Figure 8–7. DQS/DQ Bus Mode Support per Bank in EP3SL340 Devices in the 1760-pin FineLine BGA Package

Notes to Figure 8–7:

(1) You can also use DQS/DQSn pins in some of the ×4 groups as RUP/RDN pins. You cannot use a ×4 group for memory interfaces if two pins of the group are being used as RUP and RDN pins for OCT calibration. You can still use the ×16/×18 or ×32/×36 groups that includes these ×4 groups. However, there are restrictions on using ×8/×9 groups that include these ×4 groups as described on page 8–5.

(2) All I/O pin counts include eight dedicated clock inputs (CLK1p, CLK1n, CLK3p, CLK3n, CLK8p, CLK8n, CLK10p, and CLK10n) and eight dedicated corner PLL clock inputs (PLL_L1_CLKp, PLL_L1_CLKn, PLL_L4_CLKp, PLL_L4_CLKn, PLL_R4_CLKp, PLL_R4_CLKn, PLL_R1_CLKp, and PLL_R1_CLKn) that can be used for data inputs.

(3) Some of the DQS/DQ pins in this bank can also be used as configuration pins. Choose the DQS/DQ pins that are not going to be used by your configuration scheme.

Table 8–9. DLL Reference Clock Input for EP3SL200, EP3SE260 and EP3SL340 Devices *(Note 1)*, *(2)*

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)
DLL0	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L1 PLL_L2
DLL1	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L3 PLL_L4
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B2	PLL_R3 PLL_R4
DLL3	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T2	PLL_R1 PLL_R2

Notes to Table 8–9:

(1) PLLs L1, L3, L4, B2, R1, R3, R4, and T2 are not available for the EP3SL200 H780 package.

(2) PLLs L1, L4, R1 and R4 are not available for the EP3SL200 F1152 package.

Figure 8–12 shows a simple block diagram of the DLL. The input reference clock goes into the DLL to a chain of up to 16 delay elements. The phase comparator compares the signal coming out of the end of the delay chain block to the input reference clock. The phase comparator then issues the upndn signal to the Gray-code counter. This signal increments or decrements a 6-bit delay setting (DQS delay settings) that increases or decreases the delay through the delay element chain to bring the input reference clock and the signals coming out of the delay element chain in phase.

f For the frequency range of each mode, refer to the DC and Switching Characteristics of Stratix III Devices chapter.

Table 8–10. Stratix III DLL Frequency Modes

Frequency Mode	Available Phase Shift	Number of Delay Chains
0	22.5°, 45°, 67.5°, 90°	16
1	30°, 60°, 90°, 120°	12
2	36°, 72°, 108°, 144°	10
3	45°, 90°, 135°, 180°	8
4	30°, 60°, 90°, 120°	12
5	36°, 72°, 108°, 144°	10
6	45°, 90°, 135°, 180°	8
7	60°, 120°, 180°, 240°	6

For 0° shift, the DQS signal bypasses both the DLL and DQS logic blocks. The Quartus II software automatically sets DQ input delay chains so that the skew between the DQ and DQS pin at the DQ IOE registers is negligible when the 0° shift is implemented. You can feed the DQS delay settings to the DQS logic block and logic array.

The shifted DQS signal goes to the DQS bus to clock the IOE input registers of the DQ pins. The signal can also go into the logic array for resynchronization if you are not using the IOE resynchronization registers. The shifted CQn signal can only go to the negative-edge input register in the DQ IOE and is only used for QDR II+ and QDR II SRAM interfaces.

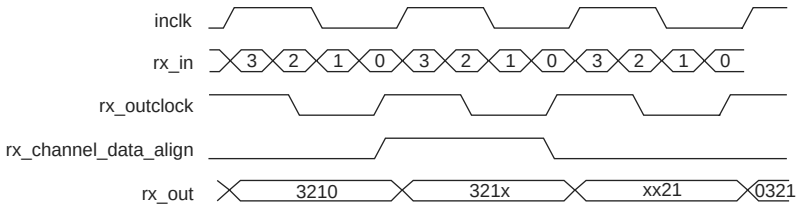
Phase Offset Control

Each DLL has two phase-offset modules and can provide two separate DQS delay settings with independent offset, one for the top and bottom I/O bank and one for the left and right I/O bank, so you can fine-tune the DQS phase shift settings between two different sides of the device. Even though you have independent phase offset control, the frequency of the interface using the same DLL has to be the same. Use the phase offset control module for making small shifts to the input signal; Use the DQS phase-shift circuitry for larger signal shifts. For example, if the DLL only offers a multiple of 30° phase shift, but your interface requires a 67.5° phase shift on the DQS signal, you can use two delay chains in the DQS logic blocks to give you 60° phase shift and use the phase offset control feature to implement the extra 7.5° phase shift.

You can either use a static phase offset or a dynamic phase offset to implement the additional phase shift. The available additional phase shift is implemented in 2's-complement in Gray-code between settings –64 to +63 for frequency modes 0, 1, 2, and 3, and between settings –32 to +31 for frequency modes 4, 5, and 6. An additional bit indicates whether the setting has a positive or negative value. The DQS phase shift is the sum of the DLL delay settings and the user selected phase offset settings. The maximum is setting 64 for frequency modes 0, 1, 2, and 3, and setting 32 for frequency modes 4, 5, 6, and 7 so the actual physical offset setting range is 64 or 32 subtracted by the DQS delay settings from the DLL.

Figure 9–7 shows receiver output (RX_OUT) after one bit slip pulse with the deserialization factor set to 4.

Figure 9–7. Data Realignment Timing



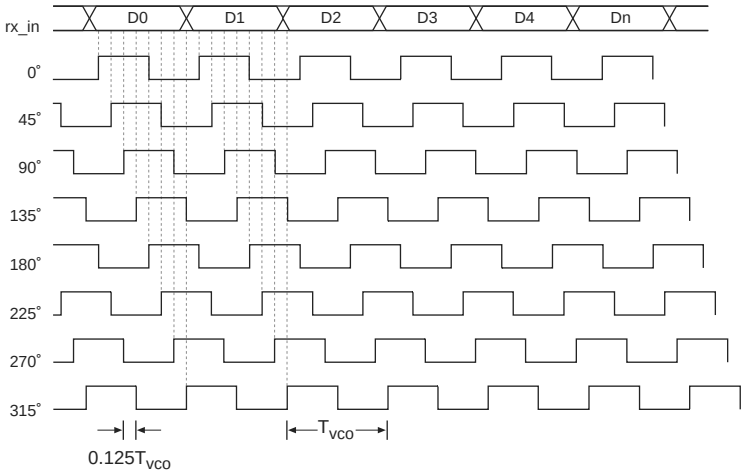
The data realignment circuit can have up to 11 bit-times of insertion before a rollover occurs. The programmable bit rollover point can be from 1 to 11 bit-times, independent of the deserialization factor. An optional status port, RX_CDA_MAX, is available to the FPGA from each channel to indicate when the preset rollover point is reached.

Dynamic Phase Aligner (DPA)

The DPA block takes in high-speed serial data from the differential input buffer and selects one of the eight phase clocks from the left/right PLL to sample the data. The DPA chooses the phase closest to the phase of the serial data. The maximum phase offset between the received data and the selected phase is 1/8 UI, which is the maximum quantization error of the DPA. The eight phases of the clock are equally divided, giving a 45° resolution.

Figure 9–8 shows the possible phase relationships between the DPA clocks and the incoming serial data.

Figure 9–8. DPA Clock Phase-to-Serial Data Timing Relationship

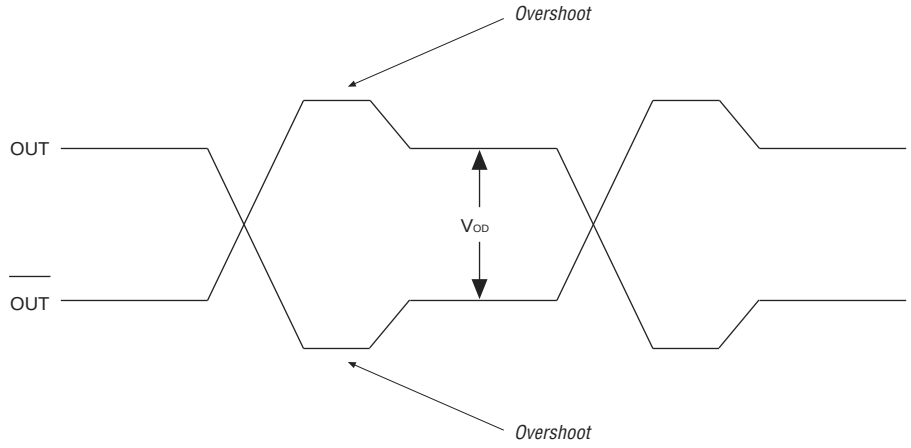


The DPA block continuously monitors the phase of the incoming serial data and selects a new clock phase if required. You can prevent the DPA from selecting a new clock phase by asserting the optional RX_DPLL_HOLD port, which is available for each channel.

Programmable Pre-Emphasis and Programmable V_{OD}

Stratix III LVDS transmitters support programmable pre-emphasis and programmable voltage output differential (V_{OD}). Pre-emphasis increases the amplitude of the high frequency component of the output signal, and thus helps compensate for the frequency dependent attenuation along the transmission line. Figure 9–10 shows an LVDS output with pre-emphasis. The overshoot is produced by pre-emphasis. This overshoot should not be included in the V_{OD} voltage. The definition of V_{OD} is also shown in Figure 9–10.

Figure 9–10. Programmable V_{OD}



Pre-emphasis is an important feature for high-speed transmission. Without pre-emphasis, the output current is limited by the V_{OD} setting and the output impedance of the driver. At high frequency, the slew rate may not be fast enough to reach the full V_{OD} before the next edge, producing a pattern dependent jitter.

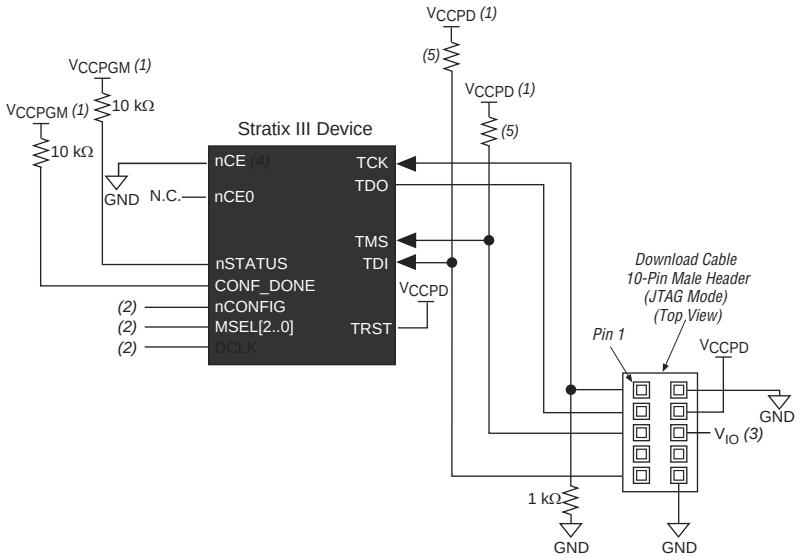
With pre-emphasis, the output current is boosted momentarily during switching to increase the output slew rate. The overshoot introduced by the extra current happens only during switching and does not ring, unlike the overshoot caused by signal reflection. The amount of pre-emphasis required depends on the attenuation of the high-frequency component along the transmission line.

Stratix III pre-emphasis is programmable to create the right amount of overshoot at different transmission conditions. There are four settings for pre-emphasis: zero, low, medium, and high. The default setting is low. In the Quartus II Assignment Editor, pre-emphasis settings are represented in numbers with 0 (zero), 1 (low), 2 (medium) and 3 (high). For a particular design, simulation with an LVDS buffer and transmission line can be used to determine the best pre-emphasis setting.

The V_{OD} is also programmable with four settings: low, medium low, medium high, and high. The default setting is medium low. In the Quartus II Assignment Editor, programmable V_{OD} settings are represented in numbers with 0 (low), 1 (medium low), 2 (medium high) and 3 (high).

Figure 11–19 shows JTAG configuration of a single Stratix III device.

Figure 11–19. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 11–19:

(1) You should connect the pull-up resistor to the same supply voltage as the USB-Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cables. The voltage supply can be connected to the V_{CCPD} of the device.

(2) You should connect the nCONFIG and MSEL[2..0] pins to support a non-JTAG configuration scheme. If you only use the JTAG configuration, connect nCONFIG to V_{CCPGM}, and MSEL[2..0] to ground. Pull DCLK either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCPD}. Refer to the *MasterBlaster Serial/USB Communications Cable Data Sheet* for this value. In the USB-Blaster, ByteBlaster II, ByteBlasterMV, and EthernetBlaster, this pin is a no connect.

(4) You must connect nCE to GND or drive it low for successful JTAG configuration.

(5) Pull-up resistor values can vary from 1 k to 10 k.

To configure a single device in a JTAG chain, the programming software places all other devices in bypass mode. In bypass mode, devices pass programming data from the TDI pin to the TDO pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the TDO pin one clock cycle later.

The Quartus II software verifies successful JTAG configuration upon completion. At the end of configuration, the software checks the state of CONF_DONE through the JTAG port. When Quartus II generates a JAM file (.jam) for a multi-device chain, it contains instructions so that all the devices in the chain will be initialized at the same time. If CONF_DONE is not high, the Quartus II software indicates that configuration has failed. If CONF_DONE is high, the software indicates that configuration was successful. After the configuration bitstream is transmitted serially through the JTAG TDI port, the TCK port is clocked an additional 1,094 cycles to perform device initialization.

Stratix III devices have dedicated JTAG pins that always function as JTAG pins. Not only can you perform JTAG testing on Stratix III devices before and after, but also during configuration. While other device families do not support JTAG testing during configuration, Stratix III devices support the bypass, id code, and sample instructions during configuration without interrupting configuration. All other JTAG instructions may only be issued by first interrupting configuration and reprogramming I/O pins using the CONFIG_IO instruction.

The CONFIG_IO instruction allows I/O buffers to be configured by using the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix III device or waiting for a configuration device to complete configuration. When configuration has been interrupted and JTAG testing is complete, you must reconfigure the part by using JTAG (PULSE_CONFIG instruction) or by pulsing nCONFIG low.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Stratix III devices do not affect JTAG boundary-scan or programming operations. Toggling these pins does not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration of Stratix III devices, consider the dedicated configuration pins. Table 11–12 lists how these pins should be connected during JTAG configuration.

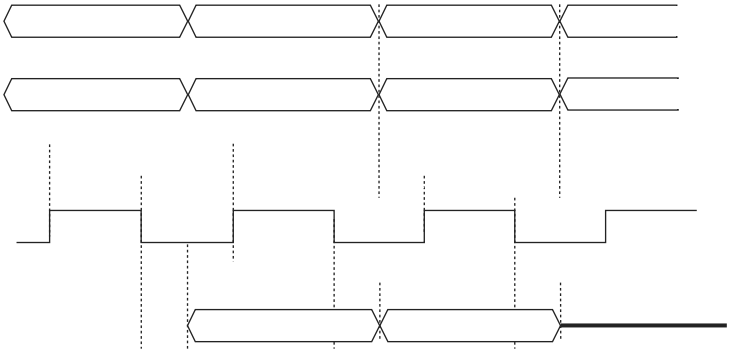
Table 11–12. Dedicated Configuration Pin Connections During JTAG Configuration

Signal	Description
nCE	On all Stratix III devices in the chain, nCE should be driven low by connecting it to ground, pulling it low by using a resistor, or driving it by some control circuitry. For devices that are also in multi-device FPP, AS, or PS configuration chains, the nCE pins should be connected to GND during JTAG configuration or JTAG configured in the same order as the configuration chain.
nCEO	On all Stratix III devices in the chain, you can leave nCEO floating or connected to the nCE of the next device.
MSEL	These pins must not be left floating. These pins support whichever non-JTAG configuration is used in production. If you only use JTAG configuration, tie these pins to ground.
nCONFIG	Driven high by connecting to V_{CCPGM}, pull up by using a resistor, or driven high by some control circuitry.
nSTATUS	Pull to V_{CCPGM} by using a 10-k resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin should be pulled up to V_{CCPGM} individually.
CONF_DONE	Pull to V_{CCPGM} by using a 10-k resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin should be pulled up to V_{CCPGM} individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration.
DCLK	Should not be left floating. Drive low or high, whichever is more convenient on your board.

When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer.

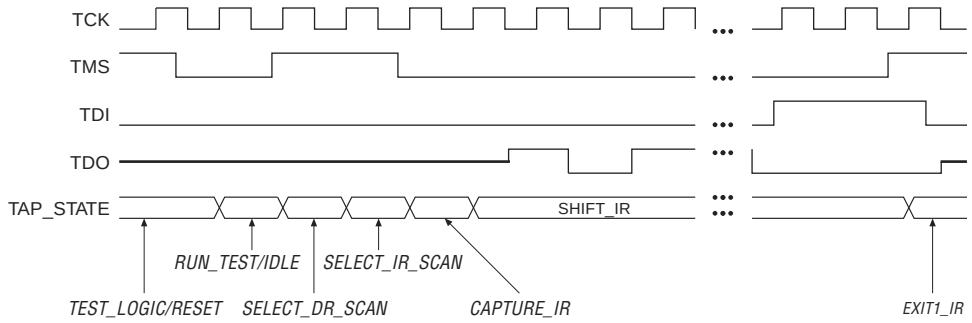
JTAG-chain device programming is ideal when the system contains multiple devices, or when testing your system using JTAG BST circuitry. Figure 11–20 shows multi-device JTAG configuration.

Figure 13–6. IEEE Std. 1149.1 Timing Waveforms



To start IEEE Std. 1149.1 operation, select an instruction mode by advancing the TAP controller to the shift instruction register (SHIFT_IR) state and shift in the appropriate instruction code on the TDI pin. The waveform diagram in Figure 13–7 shows the entry of the instruction code into the instruction register. It also shows the values of TCK, TMS, TDI, TDO, and the states of the TAP controller. From the RESET state, TMS is clocked with the pattern 01100 to advance the TAP controller to SHIFT_IR.

Figure 13–7. Selecting the Instruction Mode



The TDO pin is tri-stated in all states except in the SHIFT_IR and SHIFT_DR states. The TDO pin is activated at the first falling edge of TCK after entering either of the shift states and is tri-stated at the first falling edge of TCK after leaving either of the shift states.

When the SHIFT_IR state is activated, TDO is no longer tri-stated, and the initial state of the instruction register is shifted out on the falling edge of TCK. TDO continues to shift out the contents of the instruction register as long as the SHIFT_IR state is active. The TAP controller remains in the SHIFT_IR state as long as TMS remains low.