2. Stratix Architecture

S51002-3.2

Functional Description

Stratix® devices contain a two-dimensional row- and column-based architecture to implement custom logic. A series of column and row interconnects of varying length and speed provide signal interconnects between logic array blocks (LABs), memory block structures, and DSP blocks.

The logic array consists of LABs, with 10 logic elements (LEs) in each LAB. An LE is a small unit of logic providing efficient implementation of user logic functions. LABs are grouped into rows and columns across the device.

M512 RAM blocks are simple dual-port memory blocks with 512 bits plus parity (576 bits). These blocks provide dedicated simple dual-port or single-port memory up to 18-bits wide at up to 318 MHz. M512 blocks are grouped into columns across the device in between certain LABs.

M4K RAM blocks are true dual-port memory blocks with 4K bits plus parity (4,608 bits). These blocks provide dedicated true dual-port, simple dual-port, or single-port memory up to 36-bits wide at up to 291 MHz. These blocks are grouped into columns across the device in between certain LABs.

M-RAM blocks are true dual-port memory blocks with 512K bits plus parity (589,824 bits). These blocks provide dedicated true dual-port, simple dual-port, or single-port memory up to 144-bits wide at up to 269 MHz. Several M-RAM blocks are located individually or in pairs within the device's logic array.

Digital signal processing (DSP) blocks can implement up to either eight full-precision 9 × 9-bit multipliers, four full-precision 18 × 18-bit multipliers, or one full-precision 36 × 36-bit multiplier with add or subtract features. These blocks also contain 18-bit input shift registers for digital signal processing applications, including FIR and infinite impulse response (IIR) filters. DSP blocks are grouped into two columns in each device.

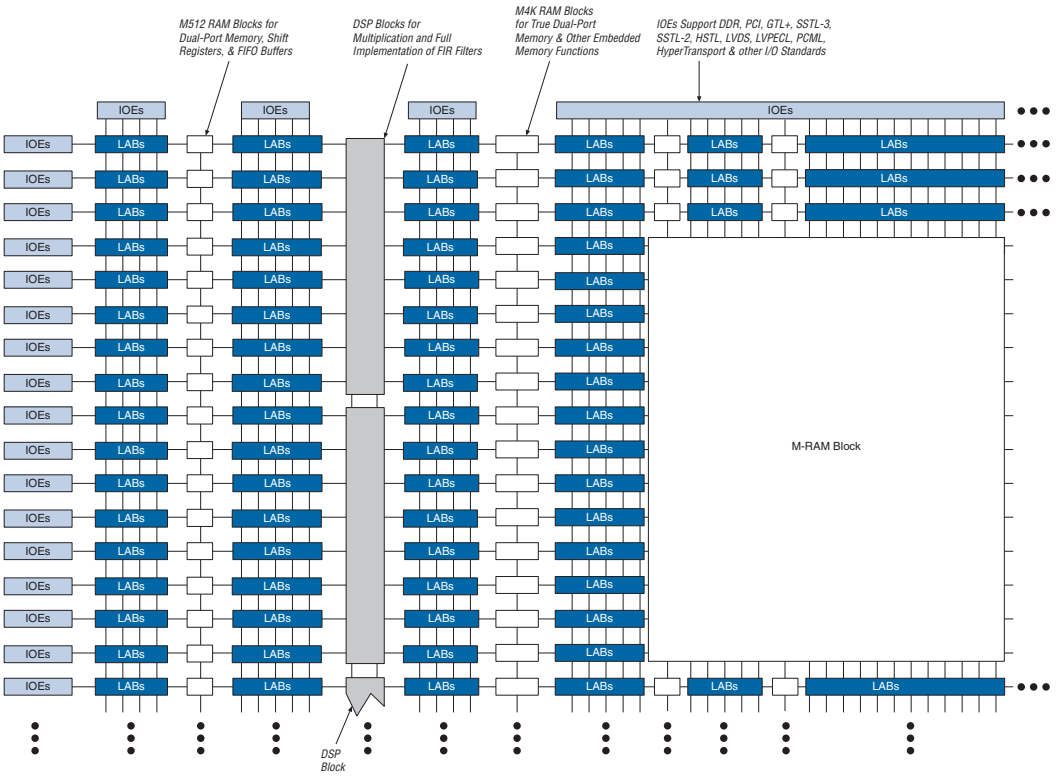
Each Stratix device I/O pin is fed by an I/O element (IOE) located at the end of LAB rows and columns around the periphery of the device. I/O pins support numerous single-ended and differential I/O standards. Each IOE contains a bidirectional I/O buffer and six registers for registering input, output, and output-enable signals. When used with

dedicated clocks, these registers provide exceptional performance and interface support with external memory devices such as DDR SDRAM, FCRAM, ZBT, and QDR SRAM devices.

High-speed serial interface channels support transfers at up to 840 Mbps using LVDS, LVPECL, 3.3-V PCML, or HyperTransport technology I/O standards.

Figure 2–1 shows an overview of the Stratix device.

Figure 2–1. Stratix Block Diagram



The number of M512 RAM, M4K RAM, and DSP blocks varies by device along with row and column numbers and M-RAM blocks. Table 2–1 lists the resources available in Stratix devices.

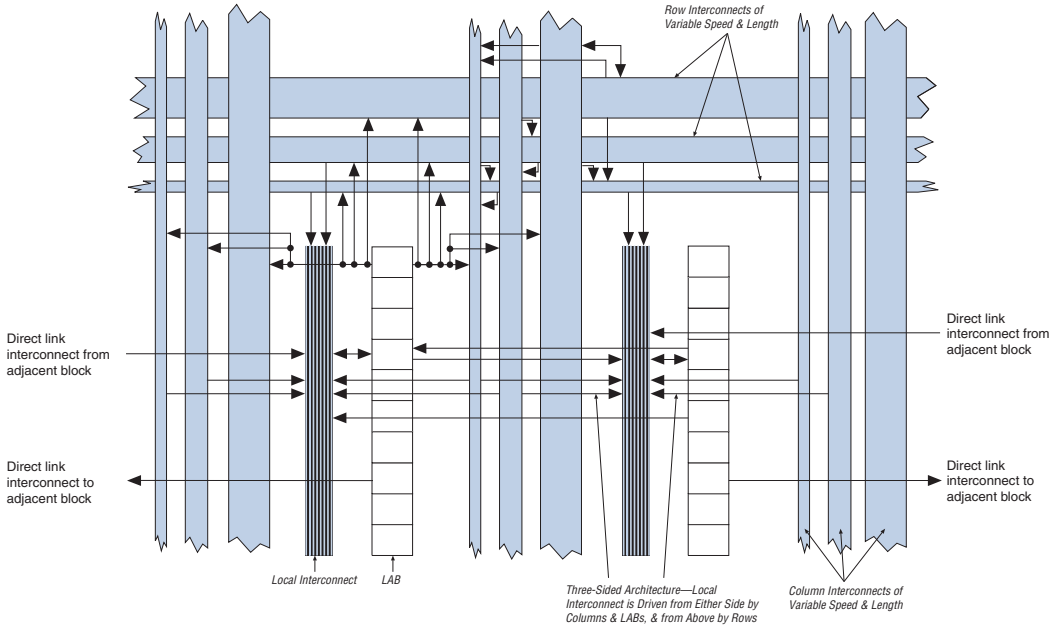
Table 2–1. Stratix Device Resources

Device	M512 RAM Columns/Blocks	M4K RAM Columns/Blocks	M-RAM Blocks	DSP Block Columns/Blocks	LAB Columns	LAB Rows
EP1S10	4 / 94	2 / 60	1	2 / 6	40	30
EP1S20	6 / 194	2 / 82	2	2 / 10	52	41
EP1S25	6 / 224	3 / 138	2	2 / 10	62	46
EP1S30	7 / 295	3 / 171	4	2 / 12	67	57
EP1S40	8 / 384	3 / 183	4	2 / 14	77	61
EP1S60	10 / 574	4 / 292	6	2 / 18	90	73
EP1S80	11 / 767	4 / 364	9	2 / 22	101	91

Logic Array Blocks

Each LAB consists of 10 LEs, LE carry chains, LAB control signals, local interconnect, LUT chain, and register chain connection lines. The local interconnect transfers signals between LEs in the same LAB. LUT chain connections transfer the output of one LE's LUT to the adjacent LE for fast sequential LUT connections within the same LAB. Register chain connections transfer the output of one LE's register to the adjacent LE's register within an LAB. The Quartus® II Compiler places associated logic within an LAB or adjacent LABs, allowing the use of local, LUT chain, and register chain connections for performance and area efficiency. Figure 2–2 shows the Stratix LAB.

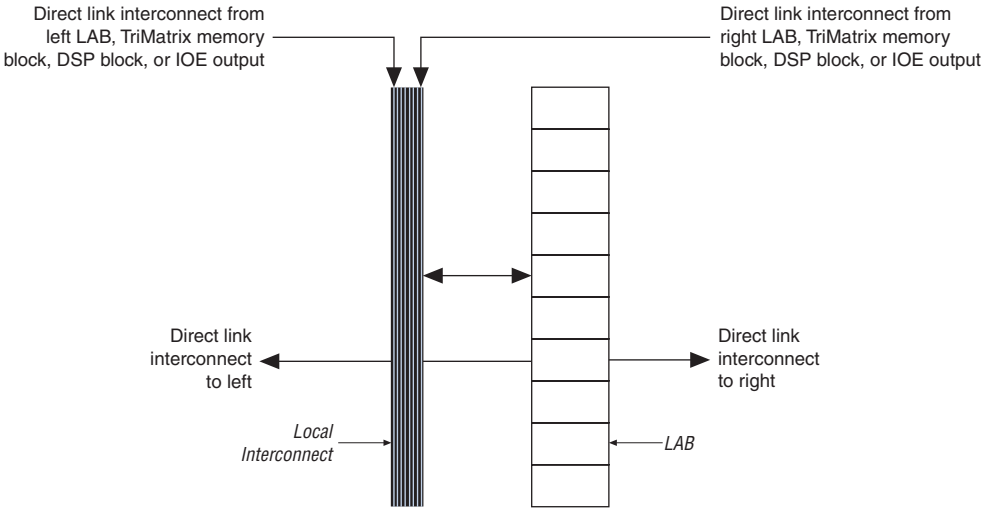
Figure 2–2. Stratix LAB Structure



LAB Interconnects

The LAB local interconnect can drive LEs within the same LAB. The LAB local interconnect is driven by column and row interconnects and LE outputs within the same LAB. Neighboring LABs, M512 RAM blocks, M4K RAM blocks, or DSP blocks from the left and right can also drive an LAB's local interconnect through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each LE can drive 30 other LEs through fast local and direct link interconnects. Figure 2–3 shows the direct link connection.

Figure 2–3. Direct Link Connection



LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its LEs. The control signals include two clocks, two clock enables, two asynchronous clears, synchronous clear, asynchronous preset/load, synchronous load, and add/subtract control signals. This gives a maximum of 10 control signals at a time. Although synchronous load and clear signals are generally used when implementing counters, they can also be used with other functions.

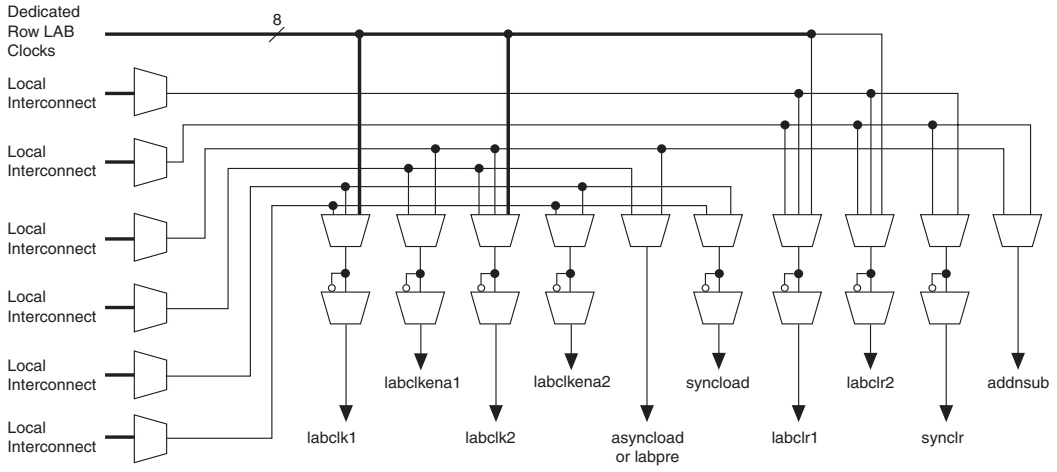
Each LAB can use two clocks and two clock enable signals. Each LAB's clock and clock enable signals are linked. For example, any LE in a particular LAB using the `labclk1` signal will also use `labclkena1`. If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock signals. De-asserting the clock enable signal will turn off the LAB-wide clock.

Each LAB can use two asynchronous clear signals and an asynchronous load/preset signal. The asynchronous load acts as a preset when the asynchronous load data input is tied high.

With the LAB-wide `addnsub` control signal, a single LE can implement a one-bit adder and subtractor. This saves LE resources and improves performance for logic functions such as DSP correlators and signed multipliers that alternate between addition and subtraction depending on data.

The LAB row clocks [7..0] and LAB local interconnect generate the LAB-wide control signals. The MultiTrack™ interconnect's inherent low skew allows clock and control signal distribution in addition to data. Figure 2–4 shows the LAB control signal generation circuit.

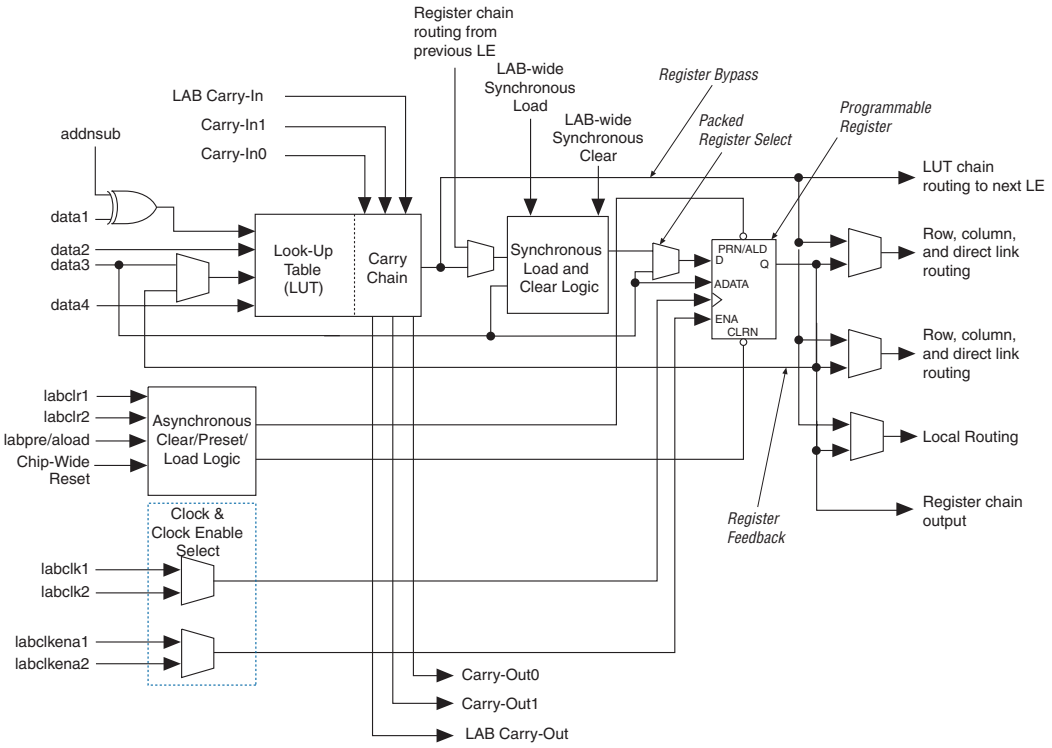
Figure 2–4. LAB-Wide Control Signals



Logic Elements

The smallest unit of logic in the Stratix architecture, the LE, is compact and provides advanced features with efficient logic utilization. Each LE contains a four-input LUT, which is a function generator that can implement any function of four variables. In addition, each LE contains a programmable register and carry chain with carry select capability. A single LE also supports dynamic single bit addition or subtraction mode selectable by an LAB-wide control signal. Each LE drives all types of interconnects: local, row, column, LUT chain, register chain, and direct link interconnects. See Figure 2–5.

Figure 2–5. Stratix LE



Each LE's programmable register can be configured for D, T, JK, or SR operation. Each register has data, true asynchronous load data, clock, clock enable, clear, and asynchronous load/preset inputs. Global signals, general-purpose I/O pins, or any internal logic can drive the register's clock and clear control signals. Either general-purpose I/O pins or internal logic can drive the clock enable, preset, asynchronous load, and asynchronous data. The asynchronous load data input comes from the `data3` input of the LE. For combinatorial functions, the register is bypassed and the output of the LUT drives directly to the outputs of the LE.

Each LE has three outputs that drive the local, row, and column routing resources. The LUT or register output can drive these three outputs independently. Two LE outputs drive column or row and direct link routing connections and one drives local interconnect resources. This allows the LUT to drive one output while the register drives another output. This feature, called register packing, improves device utilization because the device can use the register and the LUT for unrelated

functions. Another special packing mode allows the register output to feed back into the LUT of the same LE so that the register is packed with its own fan-out LUT. This provides another mechanism for improved fitting. The LE can also drive out registered and unregistered versions of the LUT output.

LUT Chain & Register Chain

In addition to the three general routing outputs, the LEs within an LAB have LUT chain and register chain outputs. LUT chain connections allow LUTs within the same LAB to cascade together for wide input functions. Register chain outputs allow registers within the same LAB to cascade together. The register chain output allows an LAB to use LUTs for a single combinatorial function and the registers to be used for an unrelated shift register implementation. These resources speed up connections between LABs while saving local interconnect resources. See "MultiTrack Interconnect" on page 2–14 for more information on LUT chain and register chain connections.

addnsub Signal

The LE's dynamic adder/subtractor feature saves logic resources by using one set of LEs to implement both an adder and a subtractor. This feature is controlled by the LAB-wide control signal `addnsub`. The `addnsub` signal sets the LAB to perform either A + B or A – B. The LUT computes addition, and subtraction is computed by adding the two's complement of the intended subtractor. The LAB-wide signal converts to two's complement by inverting the B bits within the LAB and setting carry-in = 1 to add one to the least significant bit (LSB). The LSB of an adder/subtractor must be placed in the first LE of the LAB, where the LAB-wide `addnsub` signal automatically sets the carry-in to 1. The Quartus II Compiler automatically places and uses the adder/subtractor feature when using adder/subtractor parameterized functions.

LE Operating Modes

The Stratix LE can operate in one of the following modes:

- Normal mode
- Dynamic arithmetic mode

Each mode uses LE resources differently. In each mode, eight available inputs to the LE—the four data inputs from the LAB local interconnect; `carry-in0` and `carry-in1` from the previous LE; the LAB carry-in from the previous carry-chain LAB; and the register chain connection—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear,

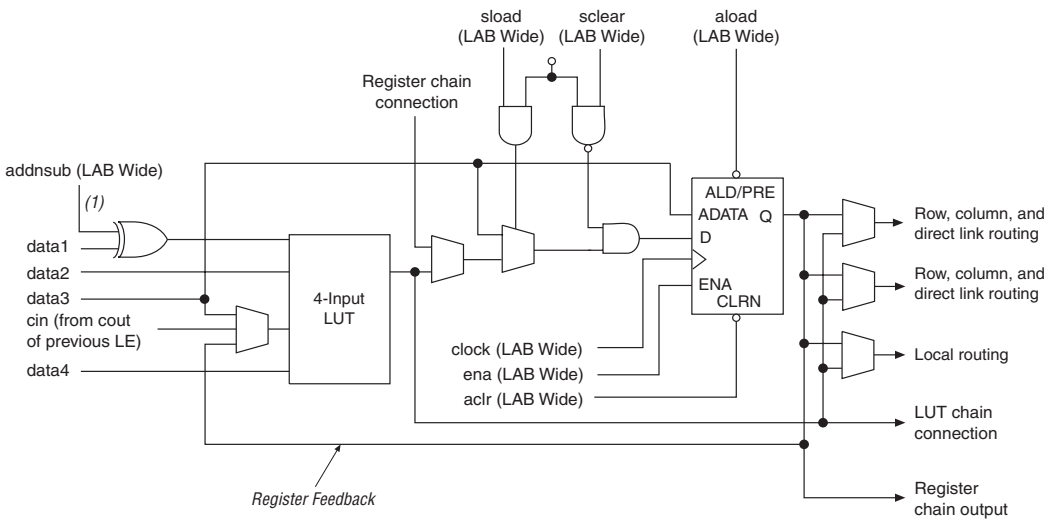
asynchronous preset load, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all LE modes. The addnsub control signal is allowed in arithmetic mode.

The Quartus II software, in conjunction with parameterized functions such as library of parameterized modules (LPM) functions, automatically chooses the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions. If required, you can also create special-purpose functions that specify which LE operating mode to use for optimal performance.

Normal Mode

The normal mode is suitable for general logic applications and combinatorial functions. In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT (see Figure 2–6). The Quartus II Compiler automatically selects the carry-in or the data3 signal as one of the inputs to the LUT. Each LE can use LUT chain connections to drive its combinatorial output directly to the next LE in the LAB. Asynchronous load data for the register comes from the data3 input of the LE. LEs in normal mode support packed registers.

Figure 2–6. LE in Normal Mode



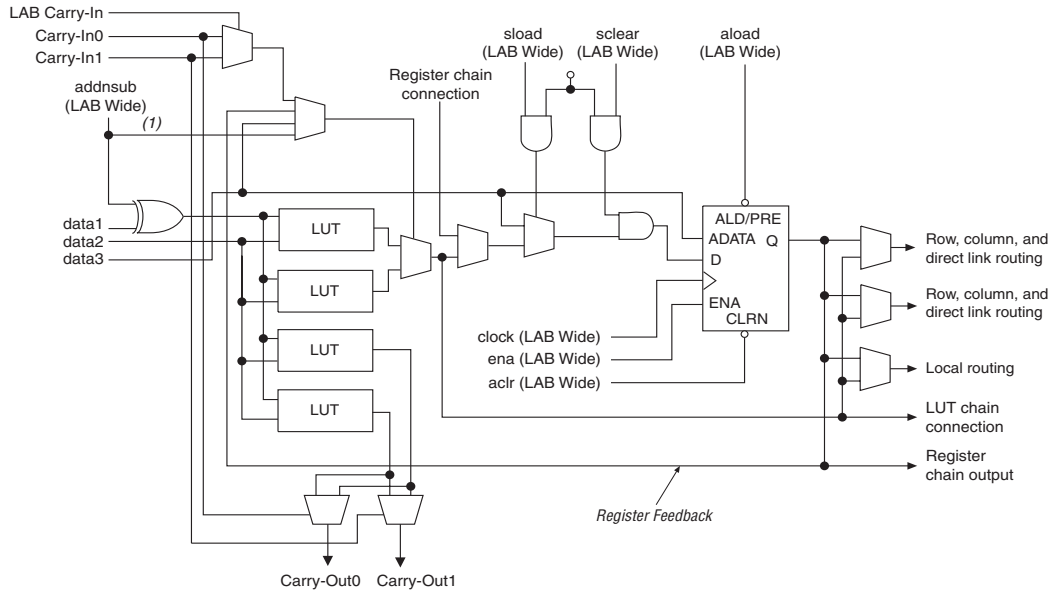
Note to Figure 2–6:
(1) This signal is only allowed in normal mode if the LE is at the end of an adder/subtractor chain.

Dynamic Arithmetic Mode

The dynamic arithmetic mode is ideal for implementing adders, counters, accumulators, wide parity functions, and comparators. An LE in dynamic arithmetic mode uses four 2-input LUTs configurable as a dynamic adder/subtractor. The first two 2-input LUTs compute two summations based on a possible carry-in of 1 or 0; the other two LUTs generate carry outputs for the two chains of the carry select circuitry. As shown in Figure 2–7, the LAB carry-in signal selects either the `carry-in0` or `carry-in1` chain. The selected chain's logic level in turn determines which parallel sum is generated as a combinatorial or registered output. For example, when implementing an adder, the sum output is the selection of two possible calculated sums: `data1` + `data2` + `carry-in0` or `data1` + `data2` + `carry-in1`. The other two LUTs use the `data1` and `data2` signals to generate two possible carry-out signals—one for a carry of 1 and the other for a carry of 0. The `carry-in0` signal acts as the carry select for the `carry-out0` output and `carry-in1` acts as the carry select for the `carry-out1` output. LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output.

The dynamic arithmetic mode also offers clock enable, counter enable, synchronous up/down control, synchronous clear, synchronous load, and dynamic adder/subtractor options. The LAB local interconnect data inputs generate the counter enable and synchronous up/down control signals. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. The Quartus II software automatically places any registers that are not used by the counter into other LABs. The `addnsub` LAB-wide signal controls whether the LE acts as an adder or subtractor.

Figure 2–7. LE in Dynamic Arithmetic Mode



Note to Figure 2–7:
(1) The `addnsub` signal is tied to the carry input for the first LE of a carry chain only.

Carry-Select Chain

The carry-select chain provides a very fast carry-select function between LEs in arithmetic mode. The carry-select chain uses the redundant carry calculation to increase the speed of carry functions. The LE is configured to calculate outputs for a possible carry-in of 1 and carry-in of 0 in parallel. The `carry-in0` and `carry-in1` signals from a lower-order bit feed forward into the higher-order bit via the parallel carry chain and feed into both the LUT and the next portion of the carry chain. Carry-select chains can begin in any LE within an LAB.

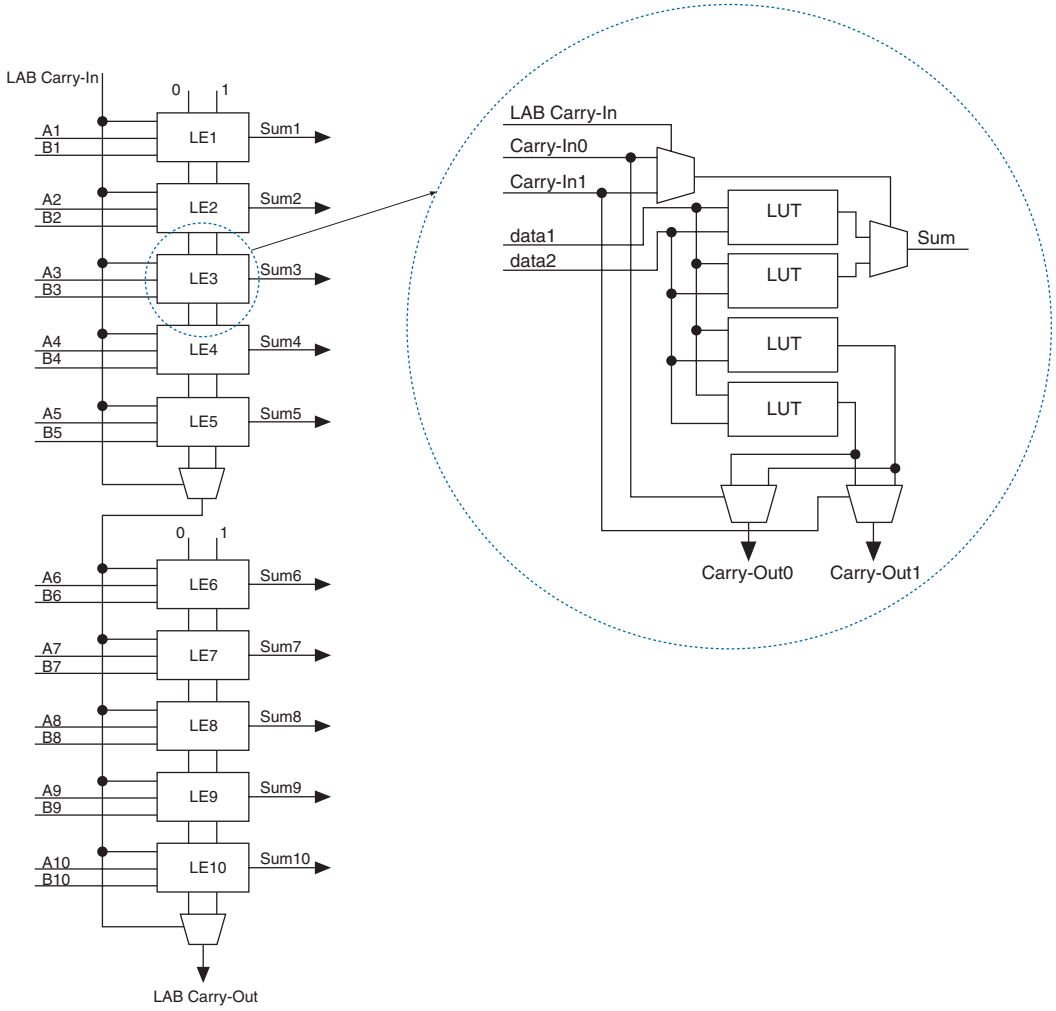
The speed advantage of the carry-select chain is in the parallel pre-computation of carry chains. Since the LAB carry-in selects the precomputed carry chain, not every LE is in the critical path. Only the propagation delay between LAB carry-in generation (LE 5 and LE 10) are now part of the critical path. This feature allows the Stratix architecture to implement high-speed counters, adders, multipliers, parity functions, and comparators of arbitrary width.

Figure 2–8 shows the carry-select circuitry in an LAB for a 10-bit full adder. One portion of the LUT generates the sum of two bits using the input signals and the appropriate carry-in bit; the sum is routed to the output of the LE. The register can be bypassed for simple adders or used for accumulator functions. Another portion of the LUT generates carry-out bits. An LAB-wide carry in bit selects which chain is used for the addition of given inputs. The carry-in signal for each chain, `carry-in0` or `carry-in1`, selects the carry-out to carry forward to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it is fed to local, row, or column interconnects.

The Quartus II Compiler automatically creates carry chain logic during design processing, or you can create it manually during design entry. Parameterized functions such as LPM functions automatically take advantage of carry chains for the appropriate functions.

The Quartus II Compiler creates carry chains longer than 10 LEs by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically allowing fast horizontal connections to TriMatrix™ memory and DSP blocks. A carry chain can continue as far as a full column.

Figure 2–8. Carry Select Chain



Clear & Preset Logic Control

LAB-wide signals control the logic for the register's clear and preset signals. The LE directly supports an asynchronous clear and preset function. The register preset is achieved through the asynchronous load of a logic high. The direct asynchronous preset does not require a NOT-gate push-back technique. Stratix devices support simultaneous preset/

asynchronous load, and clear signals. An asynchronous clear signal takes precedence if both signals are asserted simultaneously. Each LAB supports up to two clears and one preset signal.

In addition to the clear and preset ports, Stratix devices provide a chip-wide reset pin (DEV_CLRn) that resets all registers in the device. An option set before compilation in the Quartus II software controls this pin. This chip-wide reset overrides all other control signals.

MultiTrack Interconnect

In the Stratix architecture, connections between LEs, TriMatrix memory, DSP blocks, and device I/O pins are provided by the MultiTrack interconnect structure with DirectDrive™ technology. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different lengths and speeds used for inter- and intra-design block connectivity. The Quartus II Compiler automatically places critical design paths on faster interconnects to improve design performance.

DirectDrive technology is a deterministic routing technology that ensures identical routing resource usage for any function regardless of placement within the device. The MultiTrack interconnect and DirectDrive technology simplify the integration stage of block-based designing by eliminating the re-optimization cycles that typically follow design changes and additions.

The MultiTrack interconnect consists of row and column interconnects that span fixed distances. A routing structure with fixed length resources for all devices allows predictable and repeatable performance when migrating through different device densities. Dedicated row interconnects route signals to and from LABs, DSP blocks, and TriMatrix memory within the same row. These row resources include:

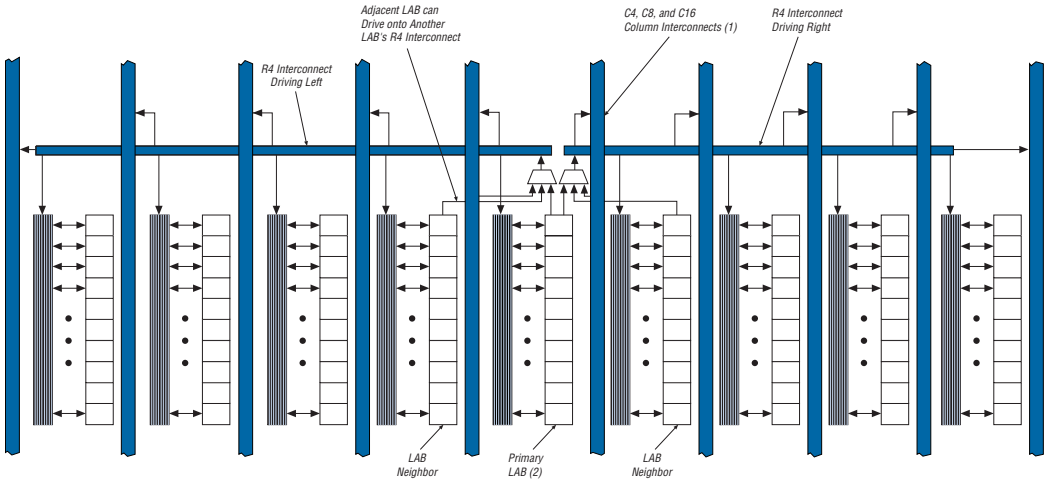
- Direct link interconnects between LABs and adjacent blocks.
- R4 interconnects traversing four blocks to the right or left.
- R8 interconnects traversing eight blocks to the right or left.
- R24 row interconnects for high-speed access across the length of the device.

The direct link interconnect allows an LAB, DSP block, or TriMatrix memory block to drive into the local interconnect of its left and right neighbors and then back into itself. Only one side of a M-RAM block interfaces with direct link and row interconnects. This provides fast communication between adjacent LABs and/or blocks without using row interconnect resources.

The R4 interconnects span four LABs, three LABs and one M512 RAM block, two LABs and one M4K RAM block, or two LABs and one DSP block to the right or left of a source LAB. These resources are used for fast

row connections in a four-LAB region. Every LAB has its own set of R4 interconnects to drive either left or right. Figure 2–9 shows R4 interconnect connections from an LAB. R4 interconnects can drive and be driven by DSP blocks and RAM blocks and horizontal IOEs. For LAB interfacing, a primary LAB or LAB neighbor can drive a given R4 interconnect. For R4 interconnects that drive to the right, the primary LAB and right neighbor can drive on to the interconnect. For R4 interconnects that drive to the left, the primary LAB and its left neighbor can drive on to the interconnect. R4 interconnects can drive other R4 interconnects to extend the range of LABs they can drive. R4 interconnects can also drive C4 and C16 interconnects for connections from one row to another. Additionally, R4 interconnects can drive R24 interconnects.

Figure 2–9. R4 Interconnect Connections



Notes to Figure 2–9:
(1) C4 interconnects can drive R4 interconnects.
(2) This pattern is repeated for every LAB in the LAB row.

The R8 interconnects span eight LABs, M512 or M4K RAM blocks, or DSP blocks to the right or left from a source LAB. These resources are used for fast row connections in an eight-LAB region. Every LAB has its own set of R8 interconnects to drive either left or right. R8 interconnect connections between LABs in a row are similar to the R4 connections shown in Figure 2–9, with the exception that they connect to eight LABs to the right or left, not four. Like R4 interconnects, R8 interconnects can drive and be driven by all types of architecture blocks. R8 interconnects

can drive other R8 interconnects to extend their range as well as C8 interconnects for row-to-row connections. One R8 interconnect is faster than two R4 interconnects connected together.

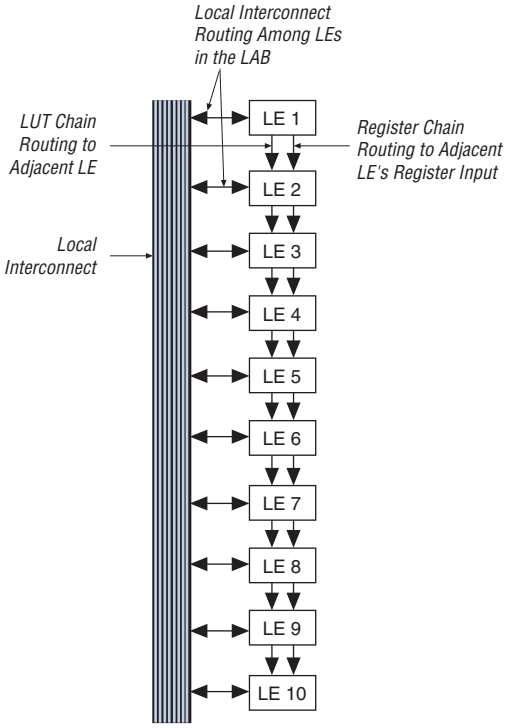
R24 row interconnects span 24 LABs and provide the fastest resource for long row connections between LABs, TriMatrix memory, DSP blocks, and IOEs. The R24 row interconnects can cross M-RAM blocks. R24 row interconnects drive to other row or column interconnects at every fourth LAB and do not drive directly to LAB local interconnects. R24 row interconnects drive LAB local interconnects via R4 and C4 interconnects. R24 interconnects can drive R24, R4, C16, and C4 interconnects.

The column interconnect operates similarly to the row interconnect and vertically routes signals to and from LABs, TriMatrix memory, DSP blocks, and IOEs. Each column of LABs is served by a dedicated column interconnect, which vertically routes signals to and from LABs, TriMatrix memory and DSP blocks, and horizontal IOEs. These column resources include:

- LUT chain interconnects within an LAB
- Register chain interconnects within an LAB
- C4 interconnects traversing a distance of four blocks in up and down direction
- C8 interconnects traversing a distance of eight blocks in up and down direction
- C16 column interconnects for high-speed vertical routing through the device

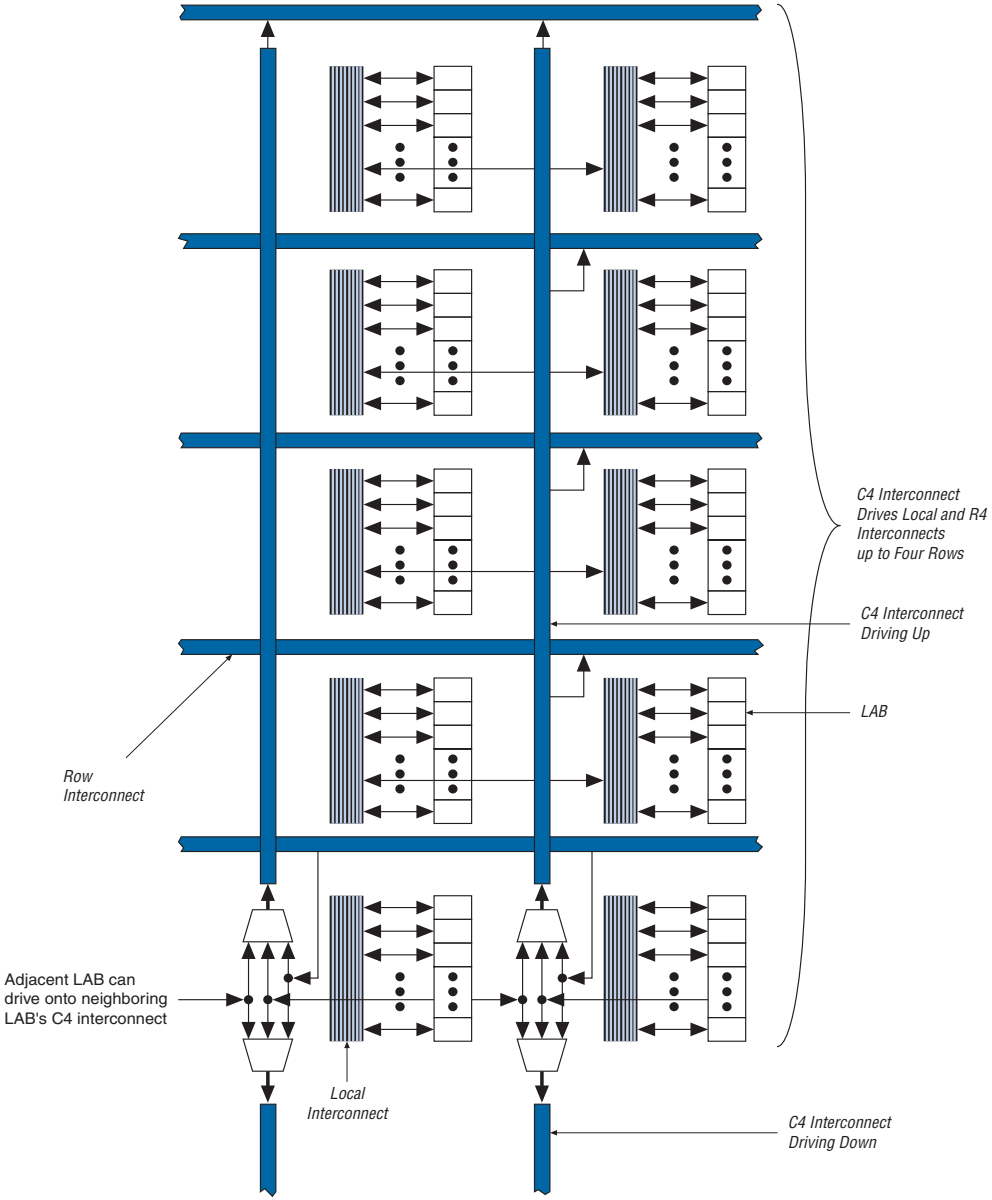
Stratix devices include an enhanced interconnect structure within LABs for routing LE output to LE input connections faster using LUT chain connections and register chain connections. The LUT chain connection allows the combinatorial output of an LE to directly drive the fast input of the LE right below it, bypassing the local interconnect. These resources can be used as a high-speed connection for wide fan-in functions from LE 1 to LE 10 in the same LAB. The register chain connection allows the register output of one LE to connect directly to the register input of the next LE in the LAB for fast shift registers. The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance. Figure 2–10 shows the LUT chain and register chain interconnects.

Figure 2–10. LUT Chain & Register Chain Interconnects



The C4 interconnects span four LABs, M512, or M4K blocks up or down from a source LAB. Every LAB has its own set of C4 interconnects to drive either up or down. Figure 2–11 shows the C4 interconnect connections from an LAB in a column. The C4 interconnects can drive and be driven by all types of architecture blocks, including DSP blocks, TriMatrix memory blocks, and vertical IOEs. For LAB interconnection, a primary LAB or its LAB neighbor can drive a given C4 interconnect.
C4 interconnects can drive each other to extend their range as well as drive row interconnects for column-to-column connections.

Figure 2–11. C4 Interconnect Connections Note (1)



Note to Figure 2–11:
(1) Each C4 interconnect can drive either up or down four rows.

C8 interconnects span eight LABs, M512, or M4K blocks up or down from a source LAB. Every LAB has its own set of C8 interconnects to drive either up or down. C8 interconnect connections between the LABs in a column are similar to the C4 connections shown in Figure 2–11 with the exception that they connect to eight LABs above and below. The C8 interconnects can drive and be driven by all types of architecture blocks similar to C4 interconnects. C8 interconnects can drive each other to extend their range as well as R8 interconnects for column-to-column connections. C8 interconnects are faster than two C4 interconnects.

C16 column interconnects span a length of 16 LABs and provide the fastest resource for long column connections between LABs, TriMatrix memory blocks, DSP blocks, and IOEs. C16 interconnects can cross M-RAM blocks and also drive to row and column interconnects at every fourth LAB. C16 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 similar to LAB-to-LAB interfaces. Each block (i.e., TriMatrix memory 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. All blocks are fed by the row LAB clocks, `labclk[7..0]`.

Table 2–2 shows the Stratix device's routing scheme.

Table 2–2. Stratix Device Routing Scheme

Source	Destination																
	LUT Chain	Register Chain	Local Interconnect	Direct Link Interconnect	R4 Interconnect	R8 Interconnect	R24 Interconnect	C4 Interconnect	C8 Interconnect	C16 Interconnect	LE	M512 RAM Block	M4K RAM Block	M-RAM Block	DSP Blocks	Column IOE	Row IOE
LUT Chain											✓						
Register Chain											✓						
Local Interconnect											✓	✓	✓	✓	✓	✓	✓
Direct Link Interconnect			✓														
R4 Interconnect			✓		✓		✓	✓		✓							
R8 Interconnect			✓			✓			✓								
R24 Interconnect					✓		✓	✓		✓							
C4 Interconnect			✓		✓			✓									
C8 Interconnect			✓			✓			✓								
C16 Interconnect					✓		✓	✓		✓							
LE	✓	✓	✓	✓	✓	✓		✓	✓								
M512 RAM Block			✓	✓	✓	✓		✓	✓								
M4K RAM Block			✓	✓	✓	✓		✓	✓								
M-RAM Block								✓	✓								
DSP Blocks			✓	✓	✓	✓		✓	✓								
Column IOE				✓				✓	✓	✓							
Row IOE				✓		✓	✓	✓	✓	✓							

TriMatrix Memory

TriMatrix memory consists of three types of RAM blocks: M512, M4K, and M-RAM blocks. Although these memory blocks are different, they can all implement various types of memory with or without parity, including true dual-port, simple dual-port, and single-port RAM, ROM, and FIFO buffers. Table 2–3 shows the size and features of the different RAM blocks.

Table 2–3. TriMatrix Memory Features (Part 1 of 2)

Memory Feature	M512 RAM Block (32 × 18 Bits)	M4K RAM Block (128 × 36 Bits)	M-RAM Block (4K × 144 Bits)
Maximum performance	*(1)*	*(1)*	*(1)*
True dual-port memory		✓	✓
Simple dual-port memory	✓	✓	✓
Single-port memory	✓	✓	✓
Shift register	✓	✓	
ROM	✓	✓	*(2)*
FIFO buffer	✓	✓	✓
Byte enable		✓	✓
Parity bits	✓	✓	✓
Mixed clock mode	✓	✓	✓
Memory initialization	✓	✓	
Simple dual-port memory mixed width support	✓	✓	✓
True dual-port memory mixed width support		✓	✓
Power-up conditions	Outputs cleared	Outputs cleared	Outputs unknown
Register clears	Input and output registers	Input and output registers	Output registers
Mixed-port read-during-write	Unknown output/old data	Unknown output/old data	Unknown output

Table 2–3. TriMatrix Memory Features (Part 2 of 2)

Memory Feature	M512 RAM Block (32 × 18 Bits)	M4K RAM Block (128 × 36 Bits)	M-RAM Block (4K × 144 Bits)
Configurations	512 × 1 256 × 2 128 × 4 64 × 8 64 × 9 32 × 16 32 × 18	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 16K × 36 8K × 64 8K × 72 4K × 128 4K × 144

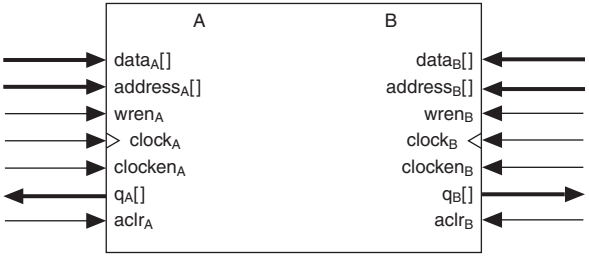
Notes to Table 2–3:
(1) See Table 4–36 for maximum performance information.
(2) The M-RAM block does not support memory initializations. However, the M-RAM block can emulate a ROM function using a dual-port RAM bock. The Stratix device must write to the dual-port memory once and then disable the write-enable ports afterwards.

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

Memory Modes

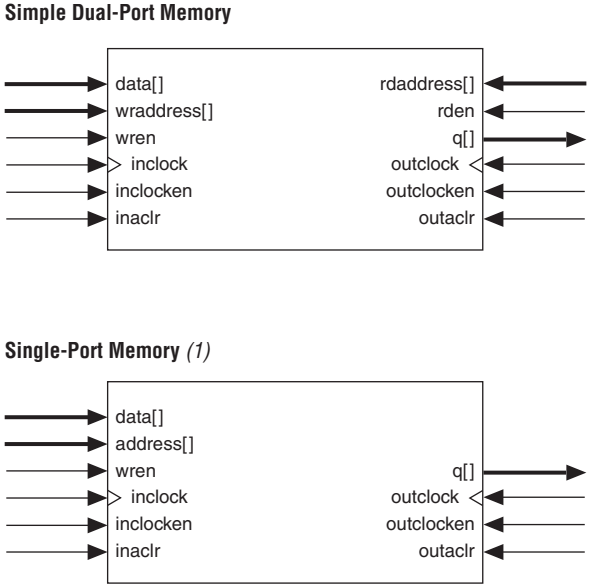
TriMatrix memory blocks include input registers that synchronize writes and output registers to pipeline designs and improve system performance. M4K and M-RAM memory blocks offer a true dual-port mode to support any combination of two-port operations: two reads, two writes, or one read and one write at two different clock frequencies. Figure 2–12 shows true dual-port memory.

Figure 2–12. True Dual-Port Memory Configuration



In addition to true dual-port memory, the memory blocks support simple dual-port and single-port RAM. Simple dual-port memory supports a simultaneous read and write and can either read old data before the write occurs or just read the don't care bits. Single-port memory supports non-simultaneous reads and writes, but the q[] port will output the data once it has been written to the memory (if the outputs are not registered) or after the next rising edge of the clock (if the outputs are registered). For more information, see Chapter 2, TriMatrix Embedded Memory Blocks in Stratix & Stratix GX Devices of the *Stratix Device Handbook, Volume 2*. Figure 2–13 shows these different RAM memory port configurations for TriMatrix memory.

Figure 2–13. Simple Dual-Port & Single-Port Memory Configurations



Note to Figure 2–13:

(1) Two single-port memory blocks can be implemented in a single M4K block as long as each of the two independent block sizes is equal to or less than half of the M4K block size.

The memory blocks also enable mixed-width data ports for reading and writing to the RAM ports in dual-port RAM configuration. For example, the memory block can be written in ×1 mode at port A and read out in ×16 mode from port B.

TriMatrix memory architecture can implement pipelined RAM by registering both the input and output signals to the RAM block. All TriMatrix memory block inputs are registered providing synchronous write cycles. In synchronous operation, the memory block generates its own self-timed strobe write enable (WREN) signal derived from the global or regional clock. In contrast, a circuit using asynchronous RAM must generate the RAM WREN signal while ensuring its data and address signals meet setup and hold time specifications relative to the WREN signal. The output registers can be bypassed. Flow-through reading is possible in the simple dual-port mode of M512 and M4K RAM blocks by clocking the read enable and read address registers on the negative clock edge and bypassing the output registers.

Two single-port memory blocks can be implemented in a single M4K block as long as each of the two independent block sizes is equal to or less than half of the M4K block size.

The Quartus II software automatically implements larger memory by combining multiple TriMatrix memory blocks. For example, two 256 × 16-bit RAM blocks can be combined to form a 256 × 32-bit RAM block. Memory performance does not degrade for memory blocks using the maximum number of words available in one memory block. Logical memory blocks using less than the maximum number of words use physical blocks in parallel, eliminating any external control logic that would increase delays. To create a larger high-speed memory block, the Quartus II software automatically combines memory blocks with LE control logic.

Clear Signals

When applied to input registers, the asynchronous clear signal for the TriMatrix embedded memory immediately clears the input registers. However, the output of the memory block does not show the effects until the next clock edge. When applied to output registers, the asynchronous clear signal clears the output registers and the effects are seen immediately.

Parity Bit Support

The memory blocks support a parity bit for each byte. The parity bit, along with internal LE logic, can implement parity checking for error detection to ensure data integrity. You can also use parity-size data words to store user-specified control bits. In the M4K and M-RAM blocks, byte enables are also available for data input masking during write operations.

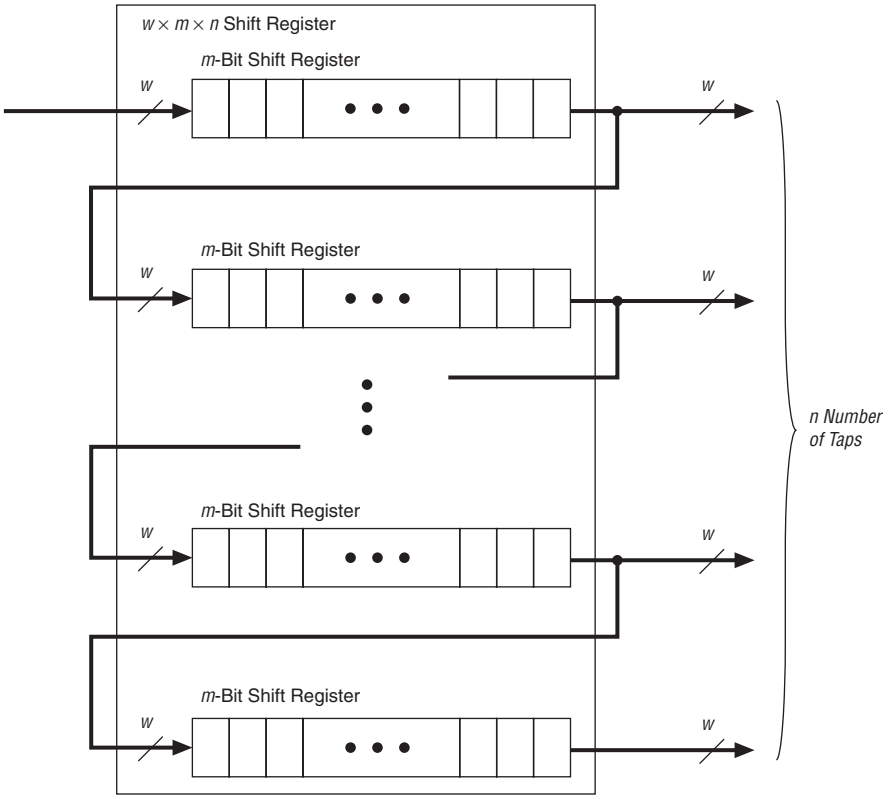
Shift Register Support

You can configure embedded memory blocks to implement shift registers for DSP applications such as pseudo-random number generators, multi-channel filtering, auto-correlation, and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flip-flops, which can quickly consume many logic cells and routing resources for large shift registers. A more efficient alternative is to use embedded memory as a shift register block, which saves logic cell and routing resources and provides a more efficient implementation with the dedicated circuitry.

The size of a $w \times m \times n$ shift register is determined by the input data width (w), the length of the taps (m), and the number of taps (n). The size of a $w \times m \times n$ shift register must be less than or equal to the maximum number of memory bits in the respective block: 576 bits for the M512 RAM block and 4,608 bits for the M4K RAM block. The total number of shift register outputs (number of taps $n \times$ width w) must be less than the maximum data width of the RAM block (18 for M512 blocks, 36 for M4K blocks). To create larger shift registers, the memory blocks are cascaded together.

Data is written into each address location at the falling edge of the clock and read from the address at the rising edge of the clock. The shift register mode logic automatically controls the positive and negative edge clocking to shift the data in one clock cycle. Figure 2–14 shows the TriMatrix memory block in the shift register mode.

Figure 2–14. Shift Register Memory Configuration



Memory Block Size

TriMatrix memory provides three different memory sizes for efficient application support. The large number of M512 blocks are ideal for designs with many shallow first-in first-out (FIFO) buffers. M4K blocks provide additional resources for channelized functions that do not require large amounts of storage. The M-RAM blocks provide a large single block of RAM ideal for data packet storage. The different-sized blocks allow Stratix devices to efficiently support variable-sized memory in designs.

The Quartus II software automatically partitions the user-defined memory into the embedded memory blocks using the most efficient size combinations. You can also manually assign the memory to a specific block size or a mixture of block sizes.

M512 RAM Block

The M512 RAM block is a simple dual-port memory block and is useful for implementing small FIFO buffers, DSP, and clock domain transfer applications. Each block contains 576 RAM bits (including parity bits). M512 RAM blocks can be configured in the following modes:

- Simple dual-port RAM
- Single-port RAM
- FIFO
- ROM
- Shift register

When configured as RAM or ROM, you can use an initialization file to pre-load the memory contents.

The memory address depths and output widths can be configured as 512 × 1, 256 × 2, 128 × 4, 64 × 8 (64 × 9 bits with parity), and 32 × 16 (32 × 18 bits with parity). Mixed-width configurations are also possible, allowing different read and write widths. Table 2–4 summarizes the possible M512 RAM block configurations.

Table 2–4. M512 RAM Block Configurations (Simple Dual-Port RAM)

Read Port	Write Port						
	512 × 1	256 × 2	128 × 4	64 × 8	32 × 16	64 × 9	32 × 18
512 × 1	✓	✓	✓	✓	✓		
256 × 2	✓	✓	✓	✓	✓		
128 × 4	✓	✓	✓		✓		
64 × 8	✓	✓		✓			
32 × 16	✓	✓	✓		✓		
64 × 9						✓	
32 × 18							✓

When the M512 RAM block is configured as a shift register block, a shift register of size up to 576 bits is possible.

The M512 RAM block can also be configured to support serializer and deserializer applications. By using the mixed-width support in combination with DDR I/O standards, the block can function as a SERDES to support low-speed serial I/O standards using global or regional clocks. See "I/O Structure" on page 2–104 for details on dedicated SERDES in Stratix devices.

M512 RAM blocks can have different clocks on its inputs and outputs. The wren, datain, and write address registers are all clocked together from one of the two clocks feeding the block. The read address, rden, and output registers can be clocked by either of the two clocks driving the block. This allows the RAM block to operate in read/write or input/output clock modes. Only the output register can be bypassed. The eight labclk signals or local interconnect can drive the inclock, outclock, wren, rden, inclr, and outclr signals. Because of the advanced interconnect between the LAB and M512 RAM blocks, LEs can also control the wren and rden signals and the RAM clock, clock enable, and asynchronous clear signals. Figure 2–15 shows the M512 RAM block control signal generation logic.

The RAM blocks within Stratix devices have local interconnects to allow LEs and interconnects to drive into RAM blocks. The M512 RAM block local interconnect is driven by the R4, R8, C4, C8, and direct link interconnects from adjacent LABs. The M512 RAM blocks can communicate with LABs on either the left or right side through these row interconnects or with LAB columns on the left or right side with the column interconnects. Up to 10 direct link input connections to the M512 RAM block are possible from the left adjacent LABs and another 10 possible from the right adjacent LAB. M512 RAM outputs can also connect to left and right LABs through 10 direct link interconnects. The M512 RAM block has equal opportunity for access and performance to and from LABs on either its left or right side. Figure 2–16 shows the M512 RAM block to logic array interface.

Figure 2–15. M512 RAM Block Control Signals

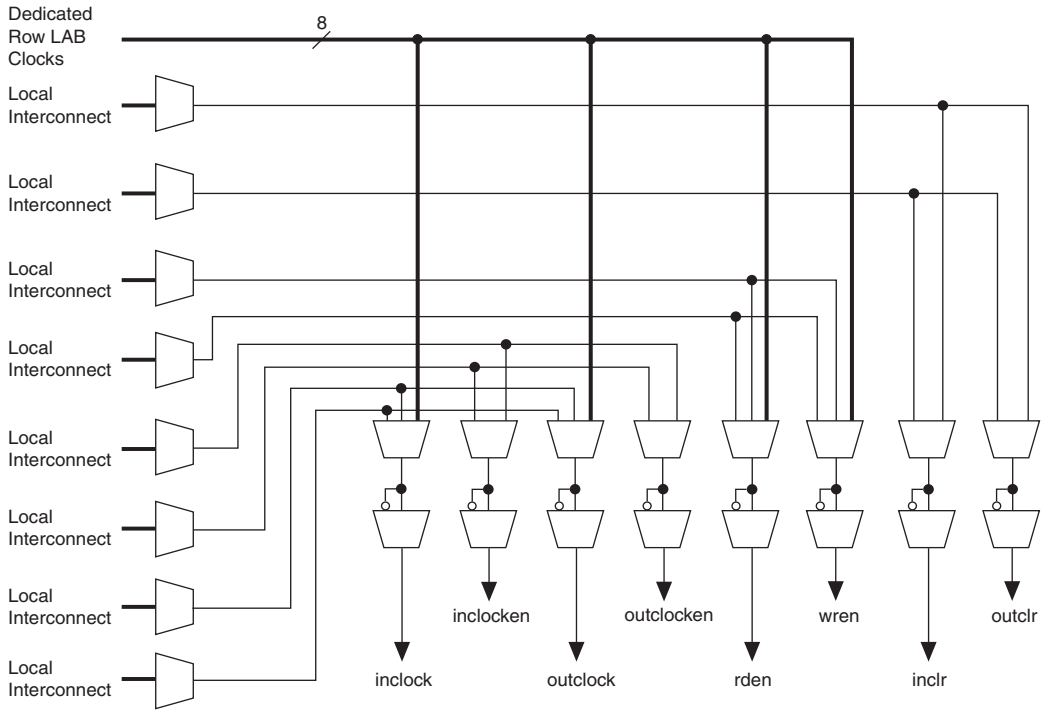
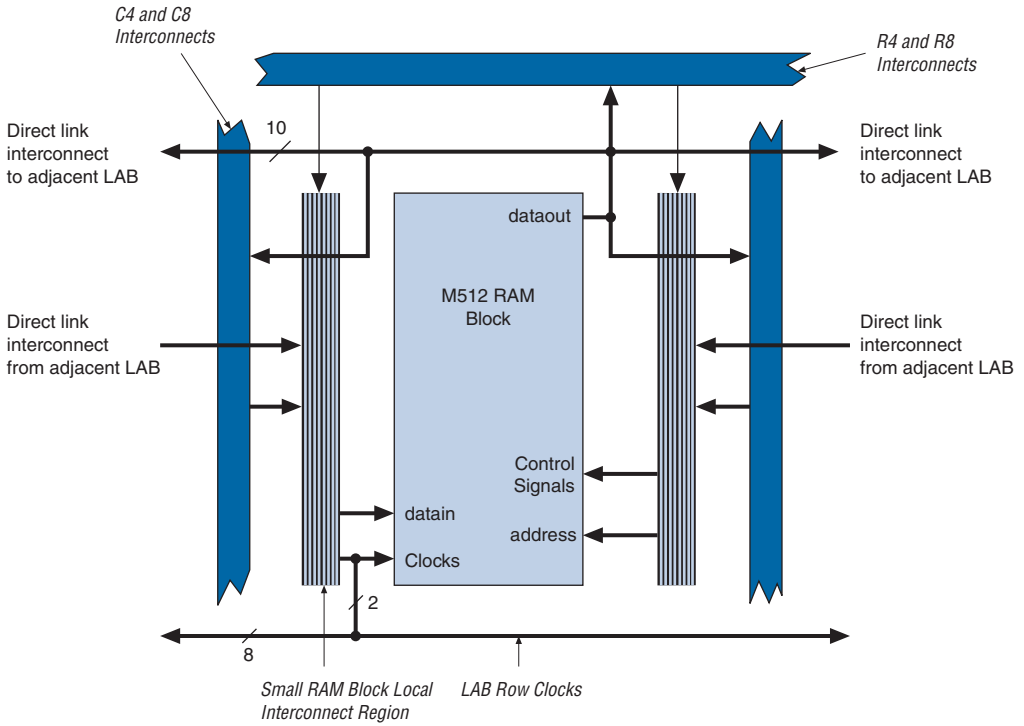


Figure 2–16. M512 RAM Block LAB Row Interface



M4K RAM Blocks

The M4K RAM block includes support for true dual-port RAM. The M4K RAM block is used to implement buffers for a wide variety of applications such as storing processor code, implementing lookup schemes, and implementing larger memory applications. Each block contains 4,608 RAM bits (including parity bits). M4K RAM blocks can be configured in the following modes:

- True dual-port RAM
- Simple dual-port RAM
- Single-port RAM
- FIFO
- ROM
- Shift register

When configured as RAM or ROM, you can use an initialization file to pre-load the memory contents.

The memory address depths and output widths can be configured as 4,096 × 1, 2,048 × 2, 1,024 × 4, 512 × 8 (or 512 × 9 bits), 256 × 16 (or 256 × 18 bits), and 128 × 32 (or 128 × 36 bits). The 128 × 32- or 36-bit configuration is not available in the true dual-port mode. Mixed-width configurations are also possible, allowing different read and write widths. Tables 2–5 and 2–6 summarize the possible M4K RAM block configurations.

Table 2–5. M4K RAM Block Configurations (Simple Dual-Port)

Read Port	Write Port								
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	128 × 32	512 × 9	256 × 18	128 × 36
4K × 1	✓	✓	✓	✓	✓	✓			
2K × 2	✓	✓	✓	✓	✓	✓			
1K × 4	✓	✓	✓	✓	✓	✓			
512 × 8	✓	✓	✓	✓	✓	✓			
256 × 16	✓	✓	✓	✓	✓	✓			
128 × 32	✓	✓	✓	✓	✓	✓			
512 × 9							✓	✓	✓
256 × 18							✓	✓	✓
128 × 36							✓	✓	✓

Table 2–6. M4K RAM Block Configurations (True Dual-Port)

Port A	Port B						
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	512 × 9	256 × 18
4K × 1	✓	✓	✓	✓	✓		
2K × 2	✓	✓	✓	✓	✓		
1K × 4	✓	✓	✓	✓	✓		
512 × 8	✓	✓	✓	✓	✓		
256 × 16	✓	✓	✓	✓	✓		
512 × 9						✓	✓
256 × 18						✓	✓

When the M4K RAM block is configured as a shift register block, you can create a shift register up to 4,608 bits ($w \times m \times n$).

M4K RAM blocks support byte writes when the write port has a data width of 16, 18, 32, or 36 bits. The byte enables allow the input data to be masked so the device can write to specific bytes. The unwritten bytes retain the previous written value. Table 2–7 summarizes the byte selection.

Table 2–7. Byte Enable for M4K Blocks Notes (1), (2)

byteena[3..0]	datain ×18	datain ×36
[0] = 1	[8..0]	[8..0]
[1] = 1	[17..9]	[17..9]
[2] = 1	–	[26..18]
[3] = 1	–	[35..27]

Notes to Table 2–7:
(1) Any combination of byte enables is possible.
(2) Byte enables can be used in the same manner with 8-bit words, i.e., in ×16 and ×32 modes.

The M4K RAM blocks allow for different clocks on their inputs and outputs. Either of the two clocks feeding the block can clock M4K RAM block registers (`renwe`, address, byte enable, `datain`, and output registers). Only the output register can be bypassed. The eight `labclk` signals or local interconnects can drive the control signals for the A and B ports of the M4K RAM block. LEs can also control the `clock_a`, `clock_b`, `renwe_a`, `renwe_b`, `clr_a`, `clr_b`, `clocken_a`, and `clocken_b` signals, as shown in Figure 2–17.

The R4, R8, C4, C8, and direct link interconnects from adjacent LABs drive the M4K RAM block local interconnect. The M4K RAM blocks can communicate with LABs on either the left or right side through these row resources or with LAB columns on either the right or left with the column resources. Up to 10 direct link input connections to the M4K RAM Block are possible from the left adjacent LABs and another 10 possible from the right adjacent LAB. M4K RAM block outputs can also connect to left and right LABs through 10 direct link interconnects each. Figure 2–18 shows the M4K RAM block to logic array interface.

Figure 2–17. M4K RAM Block Control Signals

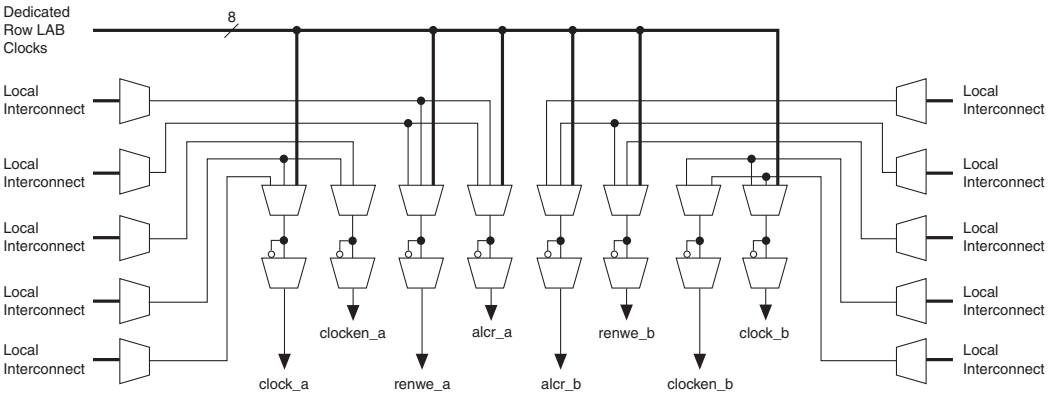
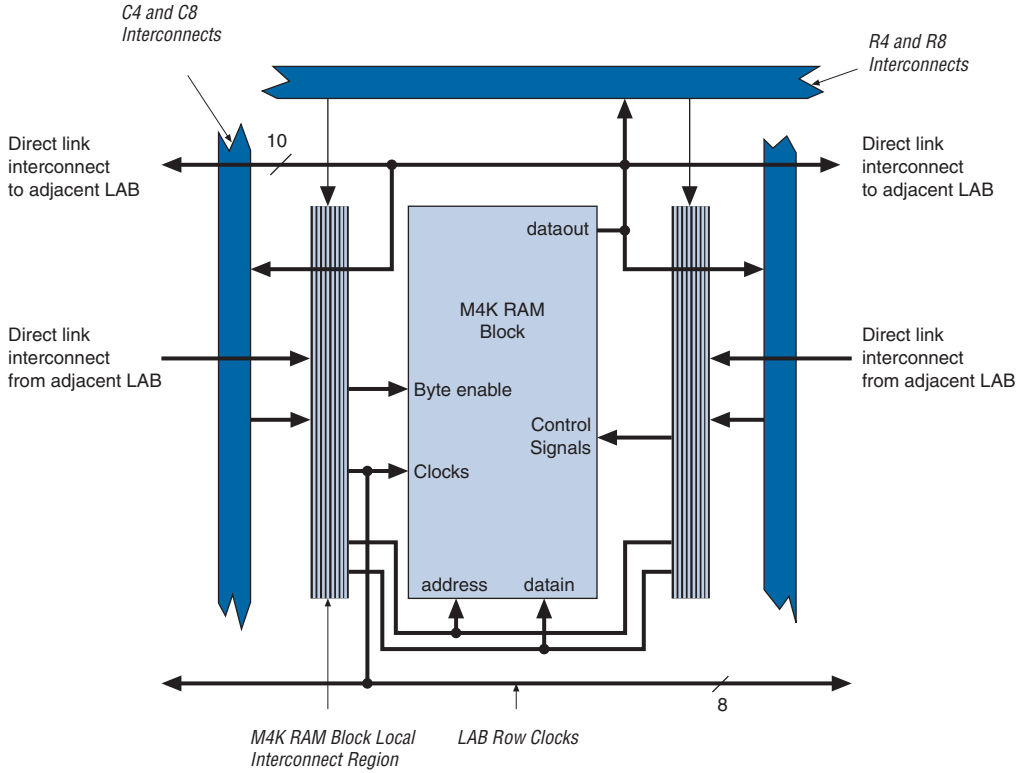


Figure 2–18. M4K RAM Block LAB Row Interface



M-RAM Block

The largest TriMatrix memory block, the M-RAM block, is useful for applications where a large volume of data must be stored on-chip. Each block contains 589,824 RAM bits (including parity bits). The M-RAM block can be configured in the following modes:

- True dual-port RAM
- Simple dual-port RAM
- Single-port RAM
- FIFO RAM

You cannot use an initialization file to initialize the contents of a M-RAM block. All M-RAM block contents power up to an undefined value. Only synchronous operation is supported in the M-RAM block, so all inputs are registered. Output registers can be bypassed. The memory address and output width can be configured as 64K × 8 (or 64K × 9 bits), 32K × 16 (or 32K × 18 bits), 16K × 32 (or 16K × 36 bits), 8K × 64 (or 8K × 72 bits), and 4K × 128 (or 4K × 144 bits). The 4K × 128 configuration is unavailable in true dual-port mode because there are a total of 144 data output drivers in the block. Mixed-width configurations are also possible, allowing different read and write widths. Tables 2–8 and 2–9 summarize the possible M-RAM block configurations:

Table 2–8. M-RAM Block Configurations (Simple Dual-Port)

Read Port	Write Port				
	64K × 9	32K × 18	16K × 36	8K × 72	4K × 144
64K × 9	✓	✓	✓	✓	
32K × 18	✓	✓	✓	✓	
16K × 36	✓	✓	✓	✓	
8K × 72	✓	✓	✓	✓	
4K × 144					✓

Table 2–9. M-RAM Block Configurations (True Dual-Port)				
Port A	Port B			
	64K × 9	32K × 18	16K × 36	8K × 72
64K × 9	✓	✓	✓	✓
32K × 18	✓	✓	✓	✓
16K × 36	✓	✓	✓	✓
8K × 72	✓	✓	✓	✓

The read and write operation of the memory is controlled by the WREN signal, which sets the ports into either read or write modes. There is no separate read enable (RE) signal.

Writing into RAM is controlled by both the WREN and byte enable (byteena) signals for each port. The default value for the byteena signal is high, in which case writing is controlled only by the WREN signal. The byte enables are available for the ×18, ×36, and ×72 modes. In the ×144 simple dual-port mode, the two sets of byteena signals (byteena_a and byteena_b) are combined to form the necessary 16 byte enables. Tables 2–10 and 2–11 summarize the byte selection.

Table 2–10. Byte Enable for M-RAM Blocks *Notes (1), (2)*			
byteena[3..0]	datain ×18	datain ×36	datain ×72
[0] = 1	[8..0]	[8..0]	[8..0]
[1] = 1	[17..9]	[17..9]	[17..9]
[2] = 1	–	[26..18]	[26..18]
[3] = 1	–	[35..27]	[35..27]
[4] = 1	–	–	[44..36]
[5] = 1	–	–	[53..45]
[6] = 1	–	–	[62..54]
[7] = 1	–	–	[71..63]

Table 2–11. M-RAM Combined Byte Selection for ×144 Mode Notes (1), (2)	
byteena[15..0]	**datain ×144**
[0] = 1	[8..0]
[1] = 1	[17..9]
[2] = 1	[26..18]
[3] = 1	[35..27]
[4] = 1	[44..36]
[5] = 1	[53..45]
[6] = 1	[62..54]
[7] = 1	[71..63]
[8] = 1	[80..72]
[9] = 1	[89..81]
[10] = 1	[98..90]
[11] = 1	[107..99]
[12] = 1	[116..108]
[13] = 1	[125..117]
[14] = 1	[134..126]
[15] = 1	[143..135]

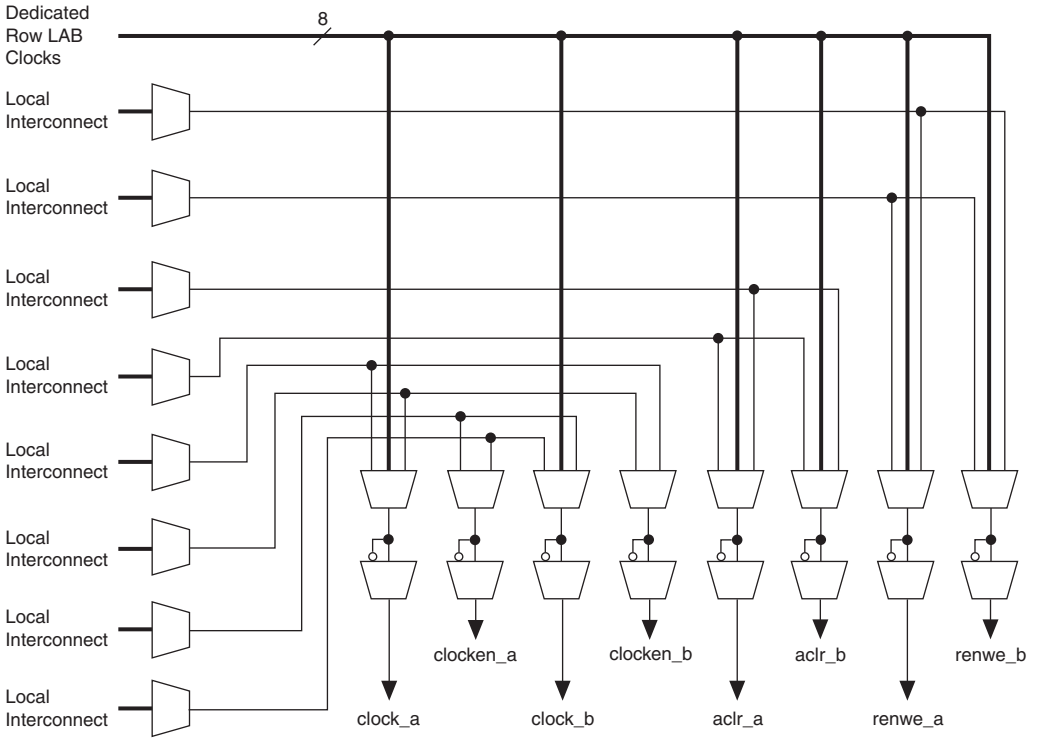
Notes to Tables 2–10 and 2–11:

(1) Any combination of byte enables is possible.

(2) Byte enables can be used in the same manner with 8-bit words, i.e., in ×16, ×32, ×64, and ×128 modes.

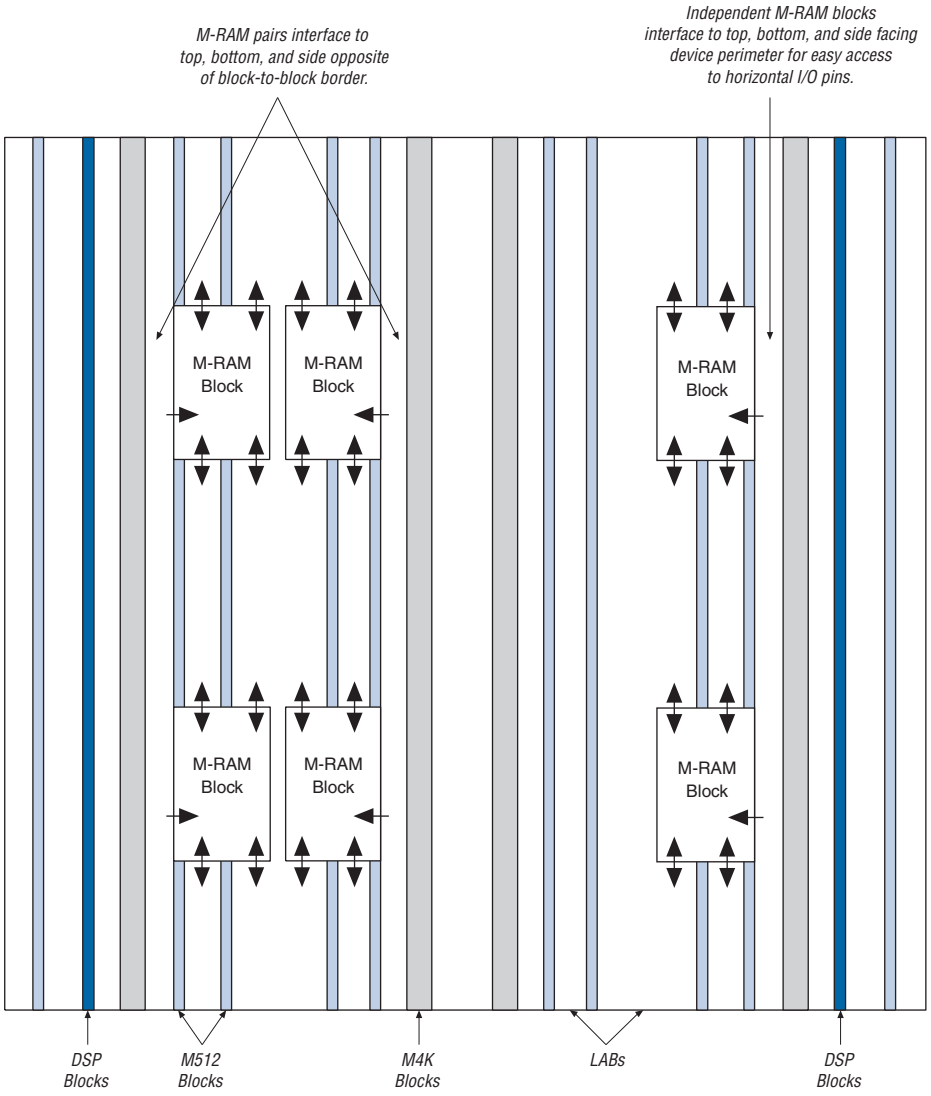
Similar to all RAM blocks, M-RAM blocks can have different clocks on their inputs and outputs. All input registers—`renwe`, `datain`, address, and byte enable registers—are clocked together from either of the two clocks feeding the block. The output register can be bypassed. The eight `labclk` signals or local interconnect can drive the control signals for the A and B ports of the M-RAM block. LEs can also control the `clock_a`, `clock_b`, `renwe_a`, `renwe_b`, `clr_a`, clr_b, `clocken_a`, and `clocken_b` signals as shown in Figure 2–19.

Figure 2–19. M-RAM Block Control Signals



One of the M-RAM block's horizontal sides drive the address and control signal (clock, renwe, byteena, etc.) inputs. Typically, the horizontal side closest to the device perimeter contains the interfaces. The one exception is when two M-RAM blocks are paired next to each other. In this case, the side of the M-RAM block opposite the common side of the two blocks contains the input interface. The top and bottom sides of any M-RAM block contain data input and output interfaces to the logic array. The top side has 72 data inputs and 72 data outputs for port B, and the bottom side has another 72 data inputs and 72 data outputs for port A. Figure 2–20 shows an example floorplan for the EP1S60 device and the location of the M-RAM interfaces.

Figure 2–20. EP1S60 Device with M-RAM Interface Locations Note (1)

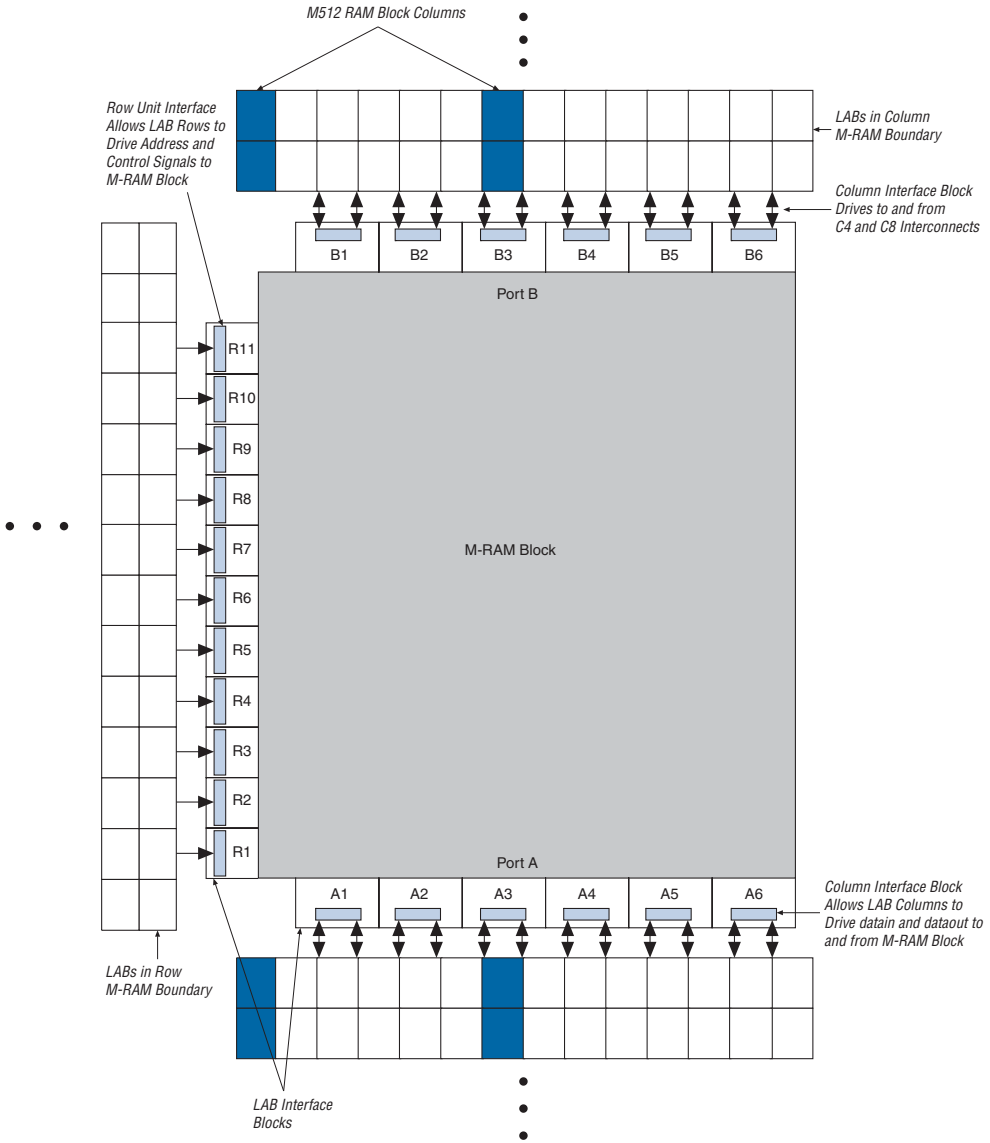


Note to Figure 2–20:
(1) Device shown is an EP1S60 device. The number and position of M-RAM blocks varies in other devices.

The M-RAM block local interconnect is driven by the R4, R8, C4, C8, and direct link interconnects from adjacent LABs. For independent M-RAM blocks, up to 10 direct link address and control signal input connections to the M-RAM block are possible from the left adjacent LABs for M-RAM

blocks facing to the left, and another 10 possible from the right adjacent LABs for M-RAM blocks facing to the right. For column interfacing, every M-RAM column unit connects to the right and left column lines, allowing each M-RAM column unit to communicate directly with three columns of LABs. Figures 2–21 through 2–23 show the interface between the M-RAM block and the logic array.

Figure 2–21. Left-Facing M-RAM to Interconnect Interface Notes (1), (2)



Notes to Figure 2–21:

(1) Only R24 and C16 interconnects cross the M-RAM block boundaries.

(2) The right-facing M-RAM block has interface blocks on the right side, but none on the left. B1 to B6 and A1 to A6 orientation is clipped across the vertical axis for right-facing M-RAM blocks.

Figure 2–22. M-RAM Row Unit Interface to Interconnect

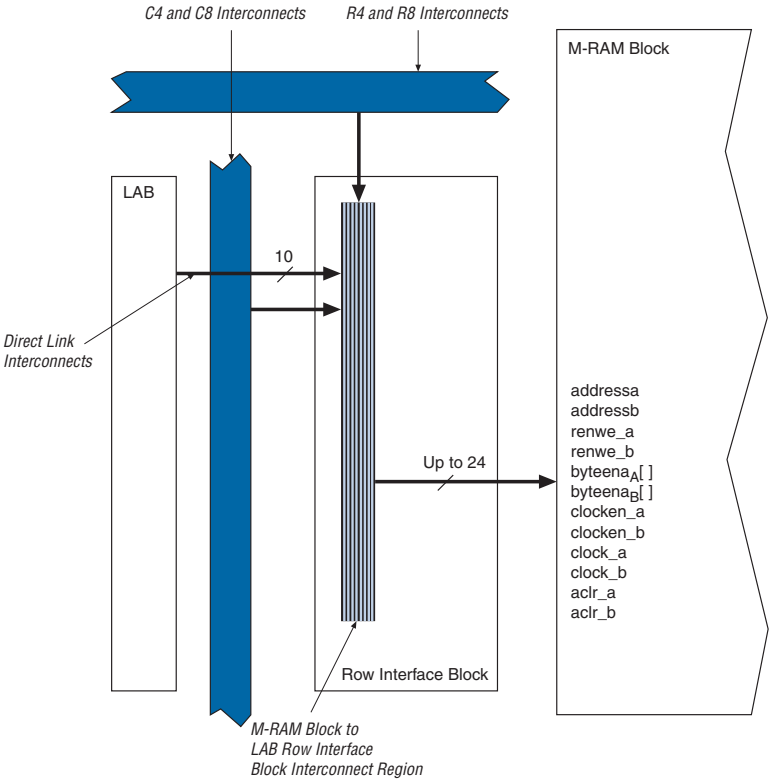


Figure 2–23. M-RAM Column Unit Interface to Interconnect

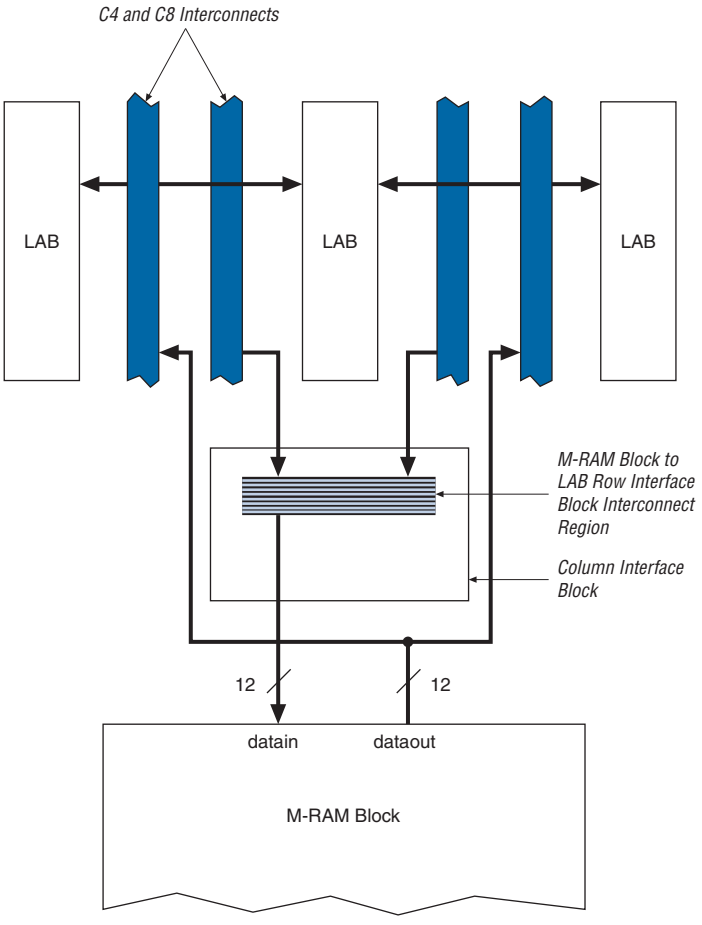


Table 2–12 shows the input and output data signal connections for the column units (B1 to B6 and A1 to A6). It also shows the address and control signal input connections to the row units (R1 to R11).

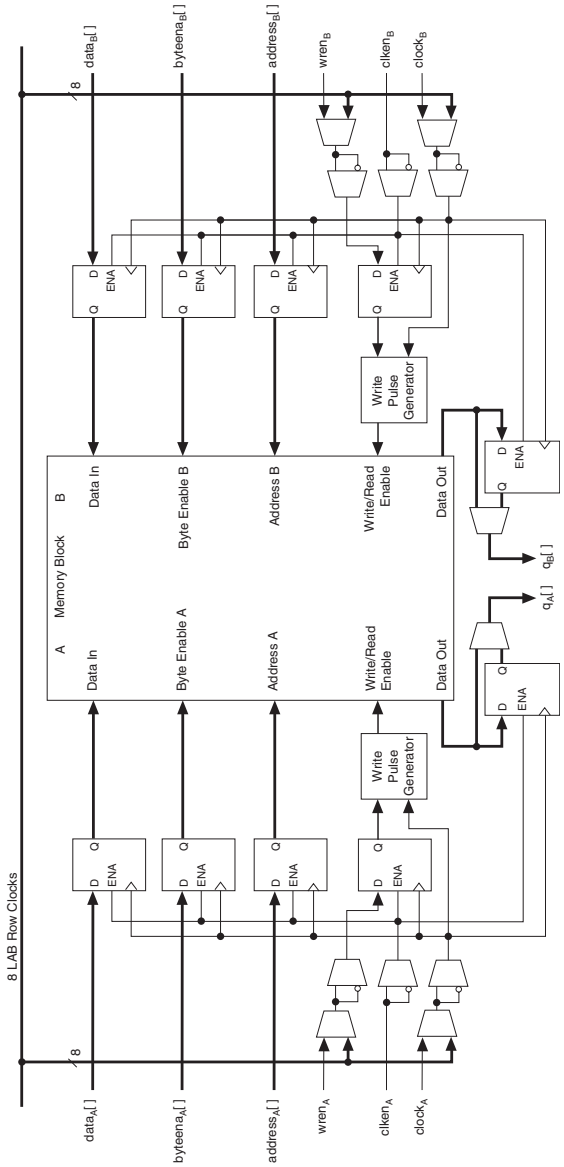
Table 2–12. M-RAM Row & Column Interface Unit Signals

Unit Interface Block	Input SIgnals	Output Signals
R1	addressa[7..0]	
R2	addressa[15..8]	
R3	byte_enable_a[7..0] renwe_a	
R4	-	
R5	-	
R6	clock_a clocken_a clock_b clocken_b	
R7	-	
R8	-	
R9	byte_enable_b[7..0] renwe_b	
R10	addressb[15..8]	
R11	addressb[7..0]	
B1	datain_b[71..60]	dataout_b[71..60]
B2	datain_b[59..48]	dataout_b[59..48]
B3	datain_b[47..36]	dataout_b[47..36]
B4	datain_b[35..24]	dataout_b[35..24]
B5	datain_b[23..12]	dataout_b[23..12]
B6	datain_b[11..0]	dataout_b[11..0]
A1	datain_a[71..60]	dataout_a[71..60]
A2	datain_a[59..48]	dataout_a[59..48]
A3	datain_a[47..36]	dataout_a[47..36]
A4	datain_a[35..24]	dataout_a[35..24]
A5	datain_a[23..12]	dataout_a[23..12]
A6	datain_a[11..0]	dataout_a[11..0]

Independent Clock Mode

The memory blocks implement independent clock mode for true dual-port memory. In this mode, a separate clock is available for each port (ports 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, A and B, also supports independent clock enables and asynchronous clear signals for port A and B registers. Figure 2–24 shows a TriMatrix memory block in independent clock mode.

Figure 2–24. Independent Clock Mode Notes (1), (2)



Notes to Figure 2–24

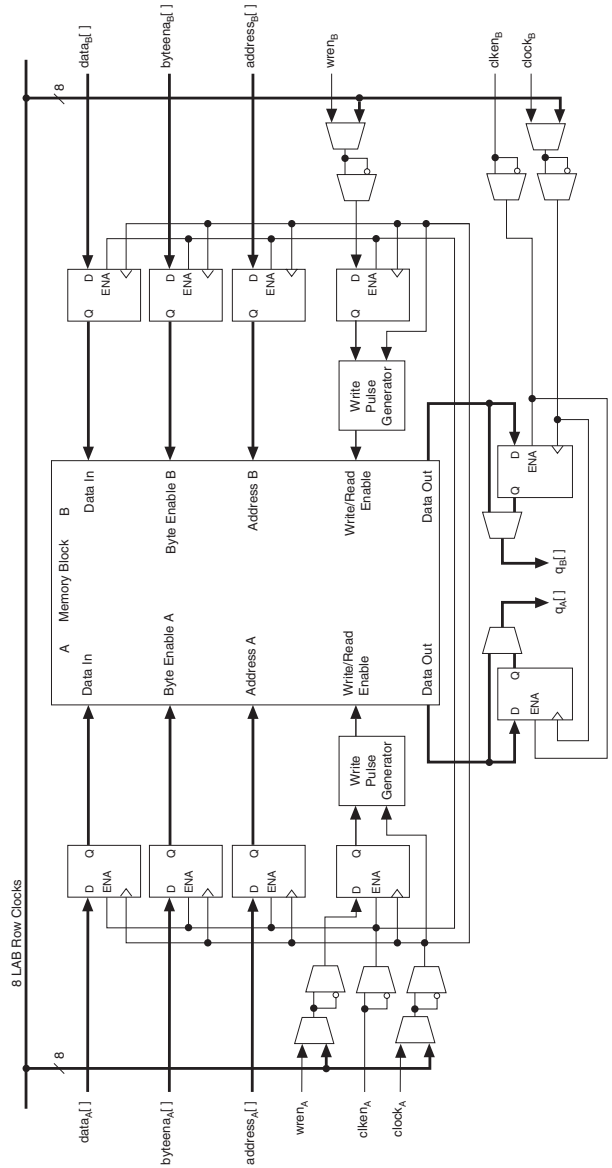
(1) All registers shown have asynchronous clear ports.

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

Input/Output Clock Mode

Input/output clock mode can be implemented for both the true and simple dual-port memory modes. On each of the two ports, A or B, one clock controls all registers for inputs into the memory block: data input, `wren`, and address. The other clock controls the block's data output registers. Each memory block port, A or B, also supports independent clock enables and asynchronous clear signals for input and output registers. Figures 2–25 and 2–26 show the memory block in input/output clock mode.

Figure 2–25. Input/Output Clock Mode in True Dual-Port Mode Notes (1), (2)

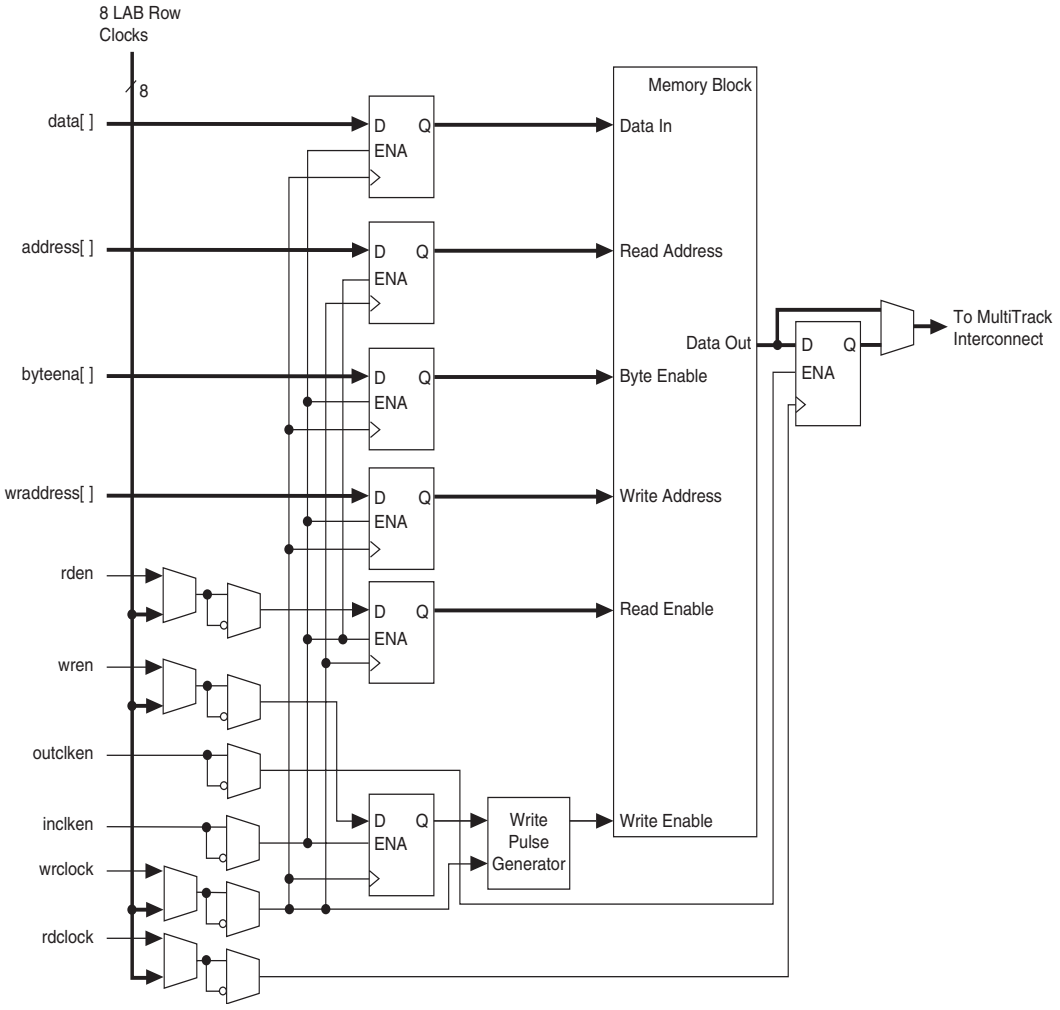


Notes to Figure 2–25:

(1) All registers shown have asynchronous clear ports.

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

Figure 2–26. Input/Output Clock Mode in Simple Dual-Port Mode Notes (1), (2)



Notes to Figure 2–26:

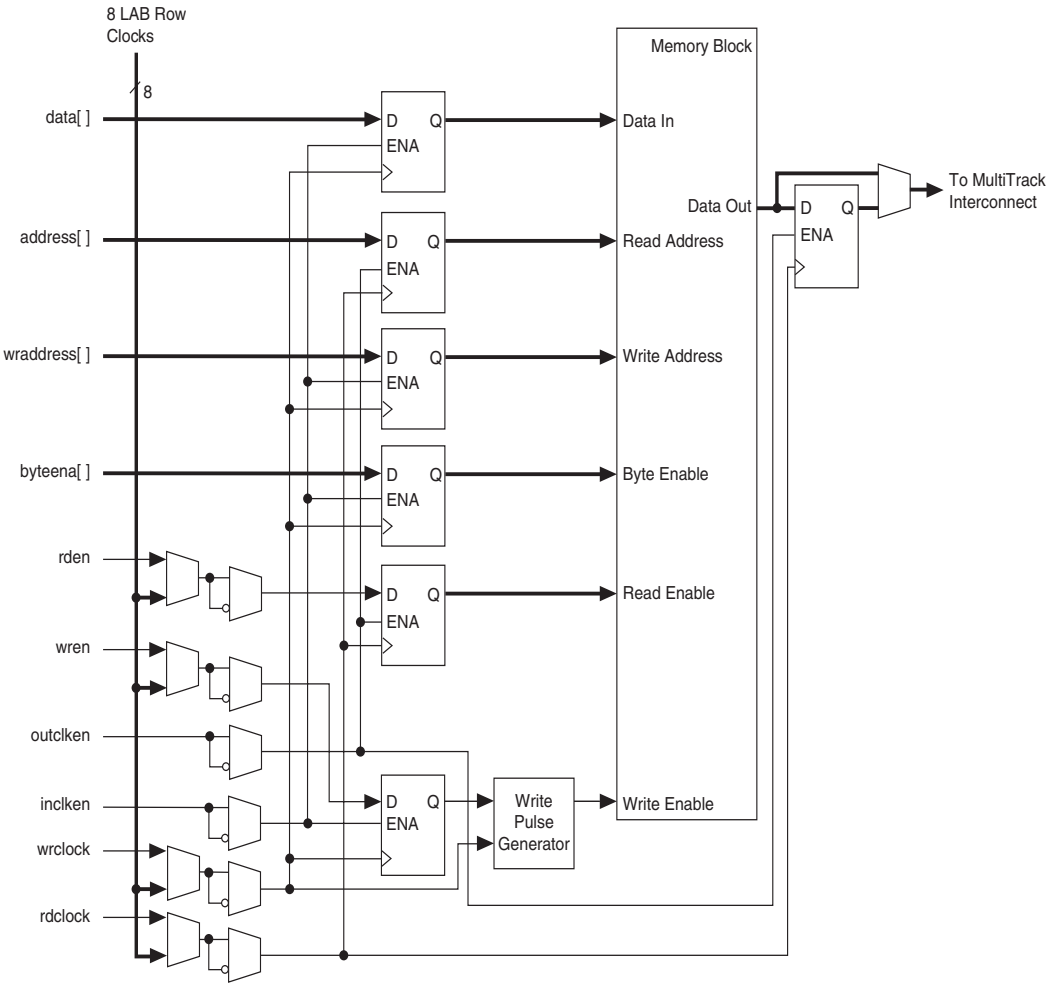
(1) All registers shown except the `rden` register have asynchronous clear ports.

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

Read/Write Clock Mode

The memory blocks implement read/write clock mode for simple dual-port memory. You can use up to two clocks in this mode. The write clock controls the block's data inputs, `wraddress`, and `wren`. The read clock controls the data output, `rdaddress`, and `rden`. The memory blocks support independent clock enables for each clock and asynchronous clear signals for the read- and write-side registers. Figure 2–27 shows a memory block in read/write clock mode.

Figure 2–27. Read/Write Clock Mode in Simple Dual-Port Mode Notes (1), (2)



Notes to Figure 2–27:

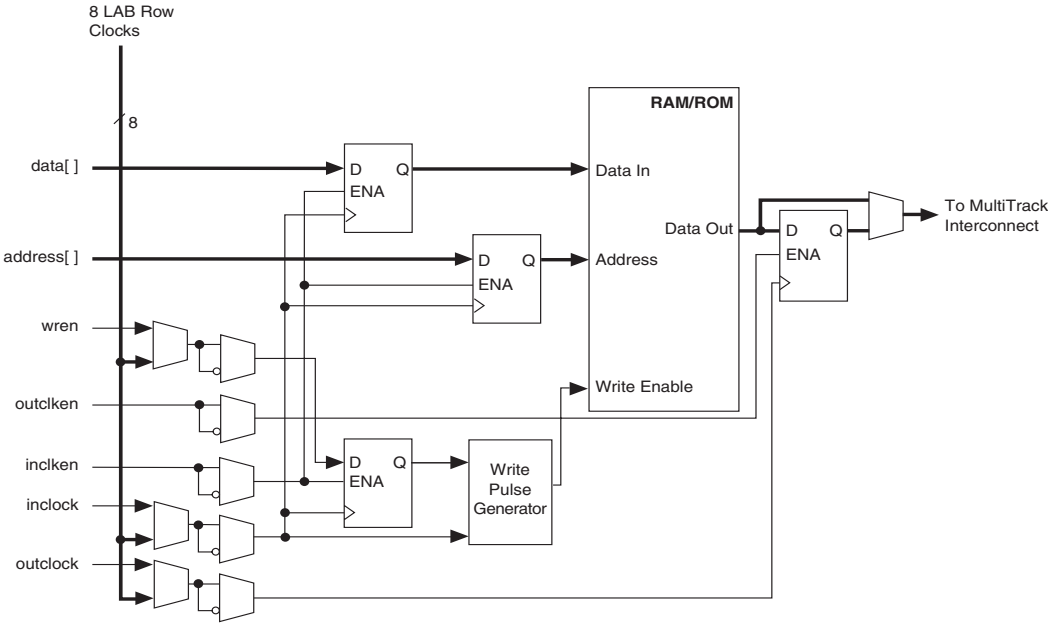
(1) All registers shown except the `rden` register have asynchronous clear ports.

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

Single-Port Mode

The memory blocks also support single-port mode, used when simultaneous reads and writes are not required. See Figure 2–28. A single block in a memory block can support up to two single-port mode RAM blocks in the M4K RAM blocks if each RAM block is less than or equal to 2K bits in size.

Figure 2–28. Single-Port Mode *Note (1)*



Note to Figure 2–28:

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

Digital Signal Processing Block

The most commonly used DSP functions are finite impulse response (FIR) filters, complex FIR filters, infinite impulse response (IIR) filters, fast Fourier transform (FFT) functions, direct cosine transform (DCT) functions, and correlators. All of these blocks have the same fundamental building block: the multiplier. Additionally, some applications need specialized operations such as multiply-add and multiply-accumulate operations. Stratix devices provide DSP blocks to meet the arithmetic requirements of these functions.

Each Stratix device has two columns of DSP blocks to efficiently implement DSP functions faster than LE-based implementations. Larger Stratix devices have more DSP blocks per column (see Table 2–13). Each DSP block can be configured to support up to:

- Eight 9 × 9-bit multipliers
- Four 18 × 18-bit multipliers
- One 36 × 36-bit multiplier

As indicated, the Stratix DSP block can support one 36 × 36-bit multiplier in a single DSP block. This is true for any matched sign multiplications (either unsigned by unsigned or signed by signed), but the capabilities for dynamic and mixed sign multiplications are handled differently. The following list provides the largest functions that can fit into a single DSP block.

- 36 × 36-bit unsigned by unsigned multiplication
- 36 × 36-bit signed by signed multiplication
- 35 × 36-bit unsigned by signed multiplication
- 36 × 35-bit signed by unsigned multiplication
- 36 × 35-bit signed by dynamic sign multiplication
- 35 × 36-bit dynamic sign by signed multiplication
- 35 × 36-bit unsigned by dynamic sign multiplication
- 36 × 35-bit dynamic sign by unsigned multiplication
- 35 × 35-bit dynamic sign multiplication when the sign controls for each operand are different
- 36 × 36-bit dynamic sign multiplication when the same sign control is used for both operands

This list only shows functions that can fit into a single DSP block. Multiple DSP blocks can support larger multiplication functions.

Figure 2–29 shows one of the columns with surrounding LAB rows.

Figure 2–29. DSP Blocks Arranged in Columns

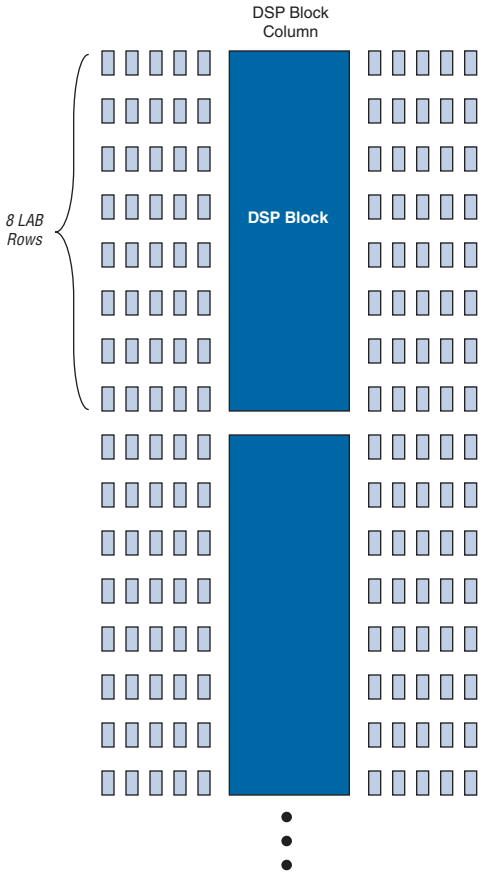


Table 2–13 shows the number of DSP blocks in each Stratix device.

Table 2–13. DSP Blocks in Stratix Devices Notes (1), (2)

Device	DSP Blocks	Total 9 × 9 Multipliers	Total 18 × 18 Multipliers	Total 36 × 36 Multipliers
EP1S10	6	48	24	6
EP1S20	10	80	40	10
EP1S25	10	80	40	10
EP1S30	12	96	48	12
EP1S40	14	112	56	14
EP1S60	18	144	72	18
EP1S80	22	176	88	22

Notes to Table 2–13:

(1) Each device has either the number of 9 × 9-, 18 × 18-, or 36 × 36-bit multipliers shown. The total number of multipliers for each device is not the sum of all the multipliers.

(2) The number of supported multiply functions shown is based on signed/signed or unsigned/unsigned implementations.

DSP block multipliers can optionally feed an adder/subtractor or accumulator within the block depending on the configuration. This makes routing to LEs easier, saves LE routing resources, and increases performance, because all connections and blocks are within the DSP block. Additionally, the DSP block input registers can efficiently implement shift registers for FIR filter applications.

Figure 2–30 shows the top-level diagram of the DSP block configured for 18 × 18-bit multiplier mode. Figure 2–31 shows the 9 × 9-bit multiplier configuration of the DSP block.

Figure 2–30. DSP Block Diagram for 18 × 18-Bit Configuration

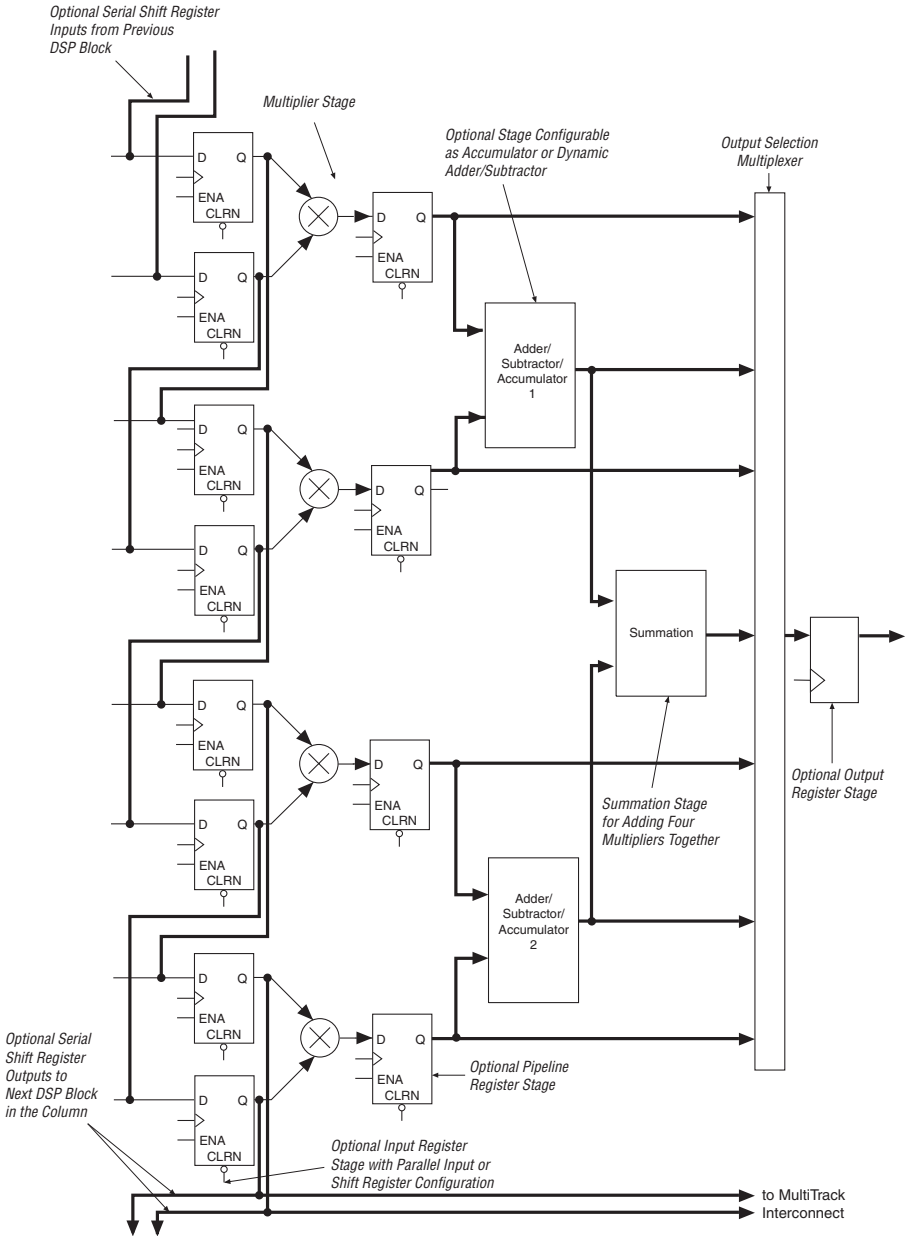
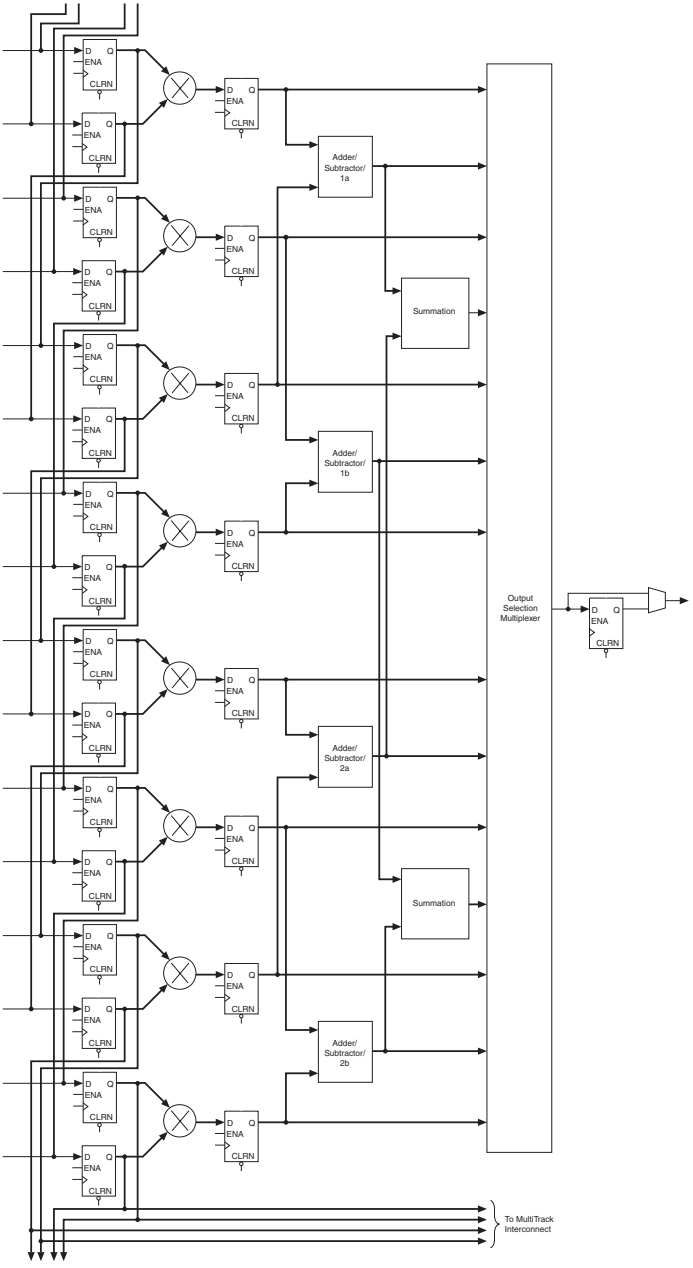


Figure 2–31. DSP Block Diagram for 9 × 9-Bit Configuration



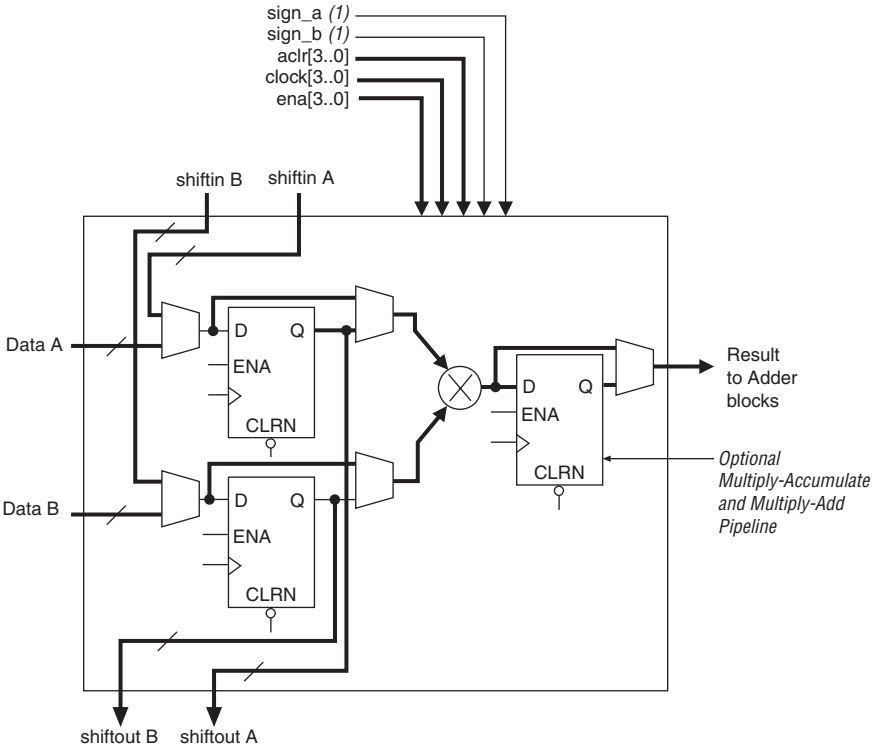
The DSP block consists of the following elements:

- Multiplier block
- Adder/output block

Multiplier Block

The DSP block multiplier block consists of the input registers, a multiplier, and pipeline register for pipelining multiply-accumulate and multiply-add/subtract functions as shown in Figure 2–32.

Figure 2–32. Multiplier Sub-Block within Stratix DSP Block



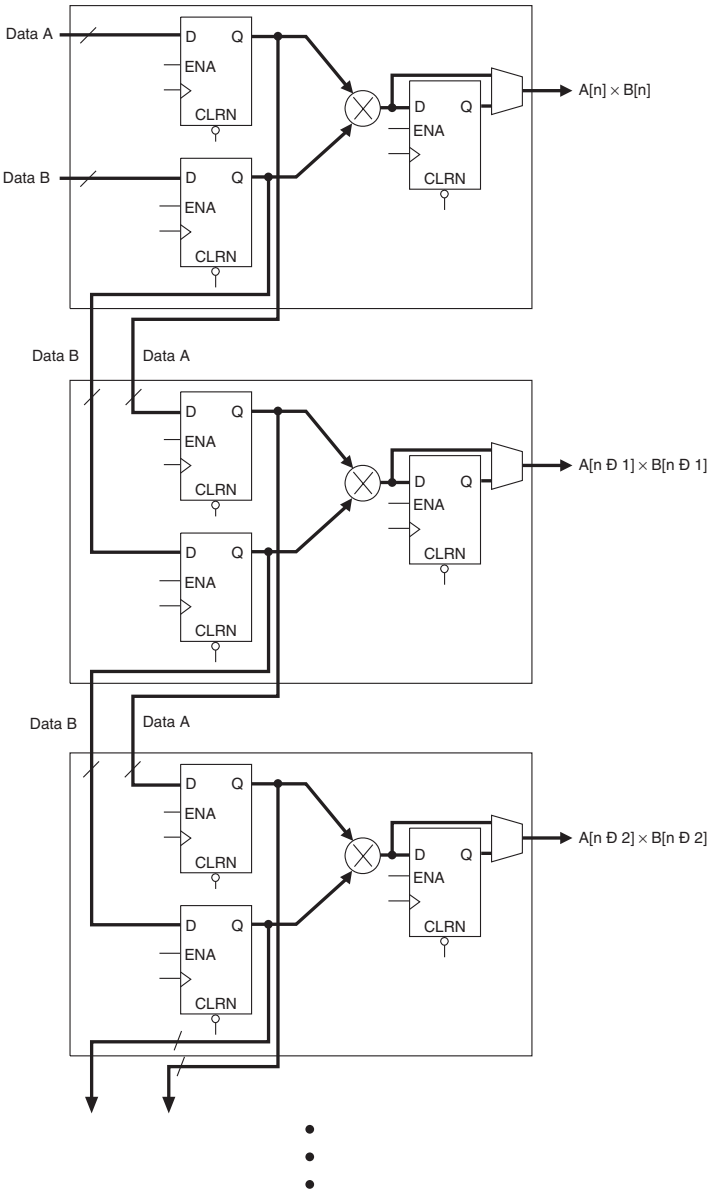
Note to Figure 2–32:
(1) These signals can be unregistered or registered once to match data path pipelines if required.

Input Registers

A bank of optional input registers is located at the input of each multiplier and multiplicand inputs to the multiplier. When these registers are configured for parallel data inputs, they are driven by regular routing resources. You can use a clock signal, asynchronous clear signal, and a clock enable signal to independently control each set of A and B inputs for each multiplier in the DSP block. You select these control signals from a set of four different `clock[3..0]`, `aclr[3..0]`, and `ena[3..0]` signals that drive the entire DSP block.

You can also configure the input registers for a shift register application. In this case, the input registers feed the multiplier and drive two dedicated shift output lines: `shiftoutA` and `shiftoutB`. The shift outputs of one multiplier block directly feed the adjacent multiplier block in the same DSP block (or the next DSP block) as shown in Figure 2–33, to form a shift register chain. This chain can terminate in any block, that is, you can create any length of shift register chain up to 224 registers. You can use the input shift registers for FIR filter applications. One set of shift inputs can provide data for a filter, and the other are coefficients that are optionally loaded in serial or parallel. When implementing 9 × 9- and 18 × 18-bit multipliers, you do not need to implement external shift registers in LAB LEs. You implement all the filter circuitry within the DSP block and its routing resources, saving LE and general routing resources for general logic. External registers are needed for shift register inputs when using 36 × 36-bit multipliers.

Figure 2–33. Multiplier Sub-Blocks Using Input Shift Register Connections
Note (1)



Note to Figure 2–33:
(1) Either Data A or Data B input can be set to a parallel input for constant coefficient multiplication.

Table 2–14 shows the summary of input register modes for the DSP block.

Table 2–14. Input Register Modes

Register Input Mode	9 × 9	18 × 18	36 × 36
Parallel input	✓	✓	✓
Shift register input	✓	✓	

Multiplier

The multiplier supports 9 × 9-, 18 × 18-, or 36 × 36-bit multiplication. Each DSP block supports eight possible 9 × 9-bit or smaller multipliers. There are four multiplier blocks available for multipliers larger than 9 × 9 bits but smaller than 18 × 18 bits. There is one multiplier block available for multipliers larger than 18 × 18 bits but smaller than or equal to 36 × 36 bits. The ability to have several small multipliers is useful in applications such as video processing. Large multipliers greater than 18 × 18 bits are useful for applications such as the mantissa multiplication of a single-precision floating-point number.

The multiplier operands can be signed or unsigned numbers, where the result is signed if either input is signed as shown in Table 2–15. The `sign_a` and `sign_b` signals provide dynamic control of each operand's representation: a logic 1 indicates the operand is a signed number, a logic 0 indicates the operand is an unsigned number. These sign signals affect all multipliers and adders within a single DSP block and you can register them to match the data path pipeline. The multipliers are full precision (that is, 18 bits for the 18-bit multiply, 36-bits for the 36-bit multiply, and so on) regardless of whether `sign_a` or `sign_b` set the operands as signed or unsigned numbers.

Table 2–15. Multiplier Signed Representation

Data A	Data B	Result
Unsigned	Unsigned	Unsigned
Unsigned	Signed	Signed
Signed	Unsigned	Signed
Signed	Signed	Signed

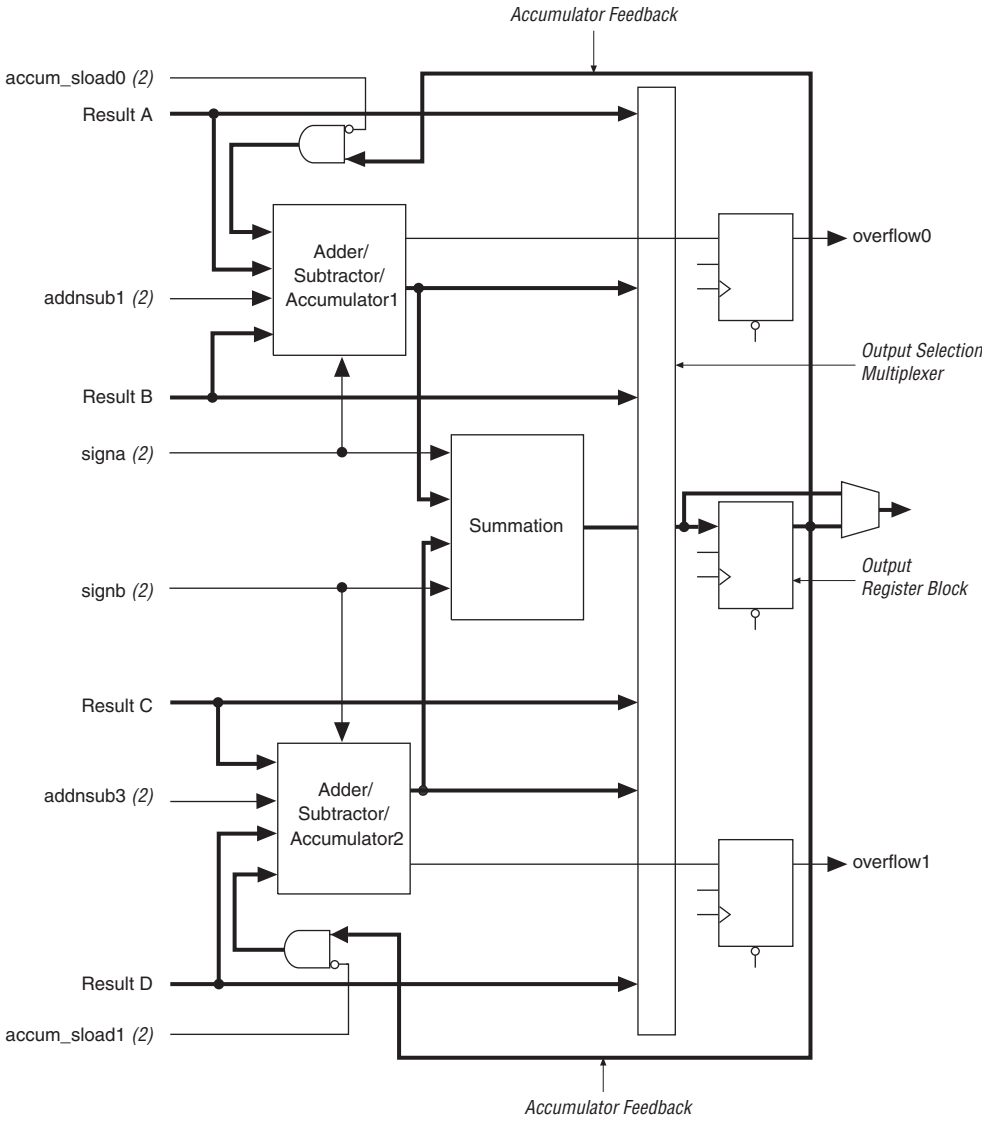
Pipeline/Post Multiply Register

The output of 9 × 9- or 18 × 18-bit multipliers can optionally feed a register to pipeline multiply-accumulate and multiply-add/subtract functions. For 36 × 36-bit multipliers, this register will pipeline the multiplier function.

Adder/Output Blocks

The result of the multiplier sub-blocks are sent to the adder/output block which consist of an adder/subtractor/accumulator unit, summation unit, output select multiplexer, and output registers. The results are used to configure the adder/output block as a pure output, accumulator, a simple two-multiplier adder, four-multiplier adder, or final stage of the 36-bit multiplier. You can configure the adder/output block to use output registers in any mode, and must use output registers for the accumulator. The system cannot use adder/output blocks independently of the multiplier. Figure 2–34 shows the adder and output stages.

Figure 2–34. Adder/Output Blocks Note (1)



Notes to Figure 2–34:

(1) Adder/output block shown in Figure 2–34 is in 18 × 18-bit mode. In 9 × 9-bit mode, there are four adder/subtractor blocks and two summation blocks.

(2) These signals are either not registered, registered once, or registered twice to match the data path pipeline.

Adder/Subtractor/Accumulator

The adder/subtractor/accumulator is the first level of the adder/output block and can be used as an accumulator or as an adder/subtractor.

Adder/Subtractor
Each adder/subtractor/accumulator block can perform addition or subtraction using the `addnsub` independent control signal for each first-level adder in 18 × 18-bit mode. There are two `addnsub[1..0]` signals available in a DSP block for any configuration. For 9 × 9-bit mode, one `addnsub[1..0]` signal controls the top two one-level adders and another `addnsub[1..0]` signal controls the bottom two one-level adders. A high `addnsub` signal indicates addition, and a low signal indicates subtraction. The `addnsub` control signal can be unregistered or registered once or twice when feeding the adder blocks to match data path pipelines.

The `signa` and `signb` signals serve the same function as the multiplier block `signa` and `signb` signals. The only difference is that these signals can be registered up to two times. These signals are tied to the same `signa` and `signb` signals from the multiplier and must be connected to the same clocks and control signals.

Accumulator
When configured for accumulation, the adder/output block output feeds back to the accumulator as shown in Figure 2–34. The `accum_sload[1..0]` signal synchronously loads the multiplier result to the accumulator output. This signal can be unregistered or registered once or twice. Additionally, the `overflow` signal indicates the accumulator has overflowed or underflowed in accumulation mode. This signal is always registered and must be externally latched in LEs if the design requires a latched `overflow` signal.

Summation

The output of the adder/subtractor/accumulator block feeds to an optional summation block. This block sums the outputs of the DSP block multipliers. In 9 × 9-bit mode, there are two summation blocks providing the sums of two sets of four 9 × 9-bit multipliers. In 18 × 18-bit mode, there is one summation providing the sum of one set of four 18 × 18-bit multipliers.

Output Selection Multiplexer

The outputs from the various elements of the adder/output block are routed through an output selection multiplexer. Based on the DSP block operational mode and user settings, the multiplexer selects whether the output from the multiplier, the adder/subtractor/accumulator, or summation block feeds to the output.

Output Registers

Optional output registers for the DSP block outputs are controlled by four sets of control signals: `clock[3..0]`, `aclr[3..0]`, and `ena[3..0]`. Output registers can be used in any mode.

Modes of Operation

The adder, subtractor, and accumulate functions of a DSP block have four modes of operation:

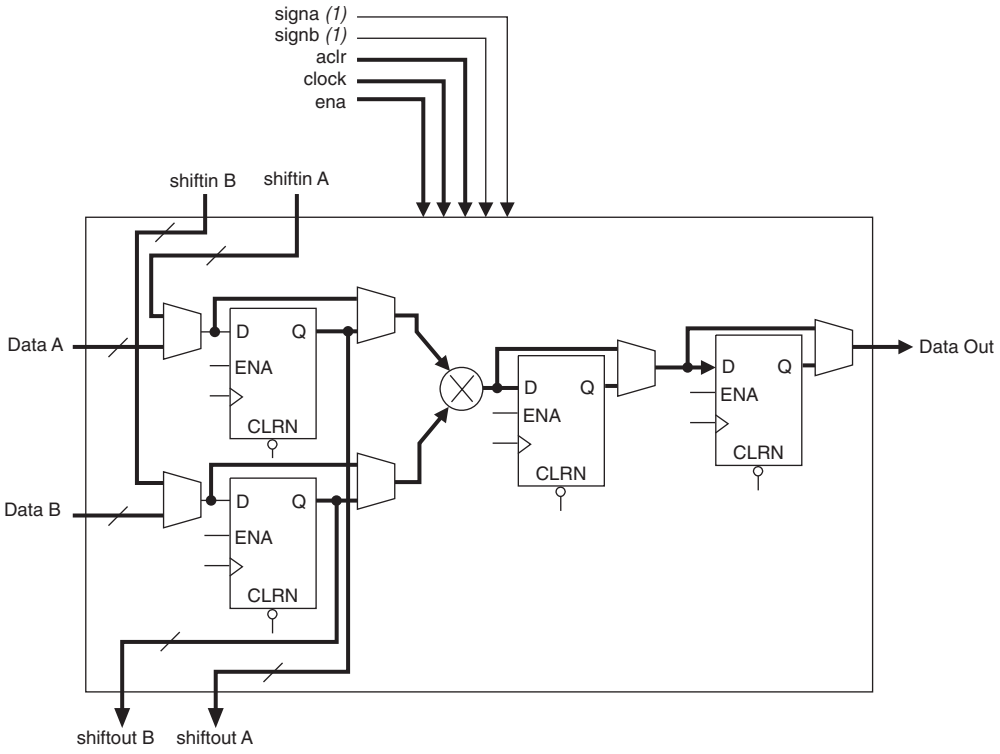
- Simple multiplier
- Multiply-accumulator
- Two-multipliers adder
- Four-multipliers adder

Each DSP block can only support one mode. Mixed modes in the same DSP block is not supported.

Simple Multiplier Mode

In simple multiplier mode, the DSP block drives the multiplier sub-block result directly to the output with or without an output register. Up to four 18 × 18-bit multipliers or eight 9 × 9-bit multipliers can drive their results directly out of one DSP block. See Figure 2–35.

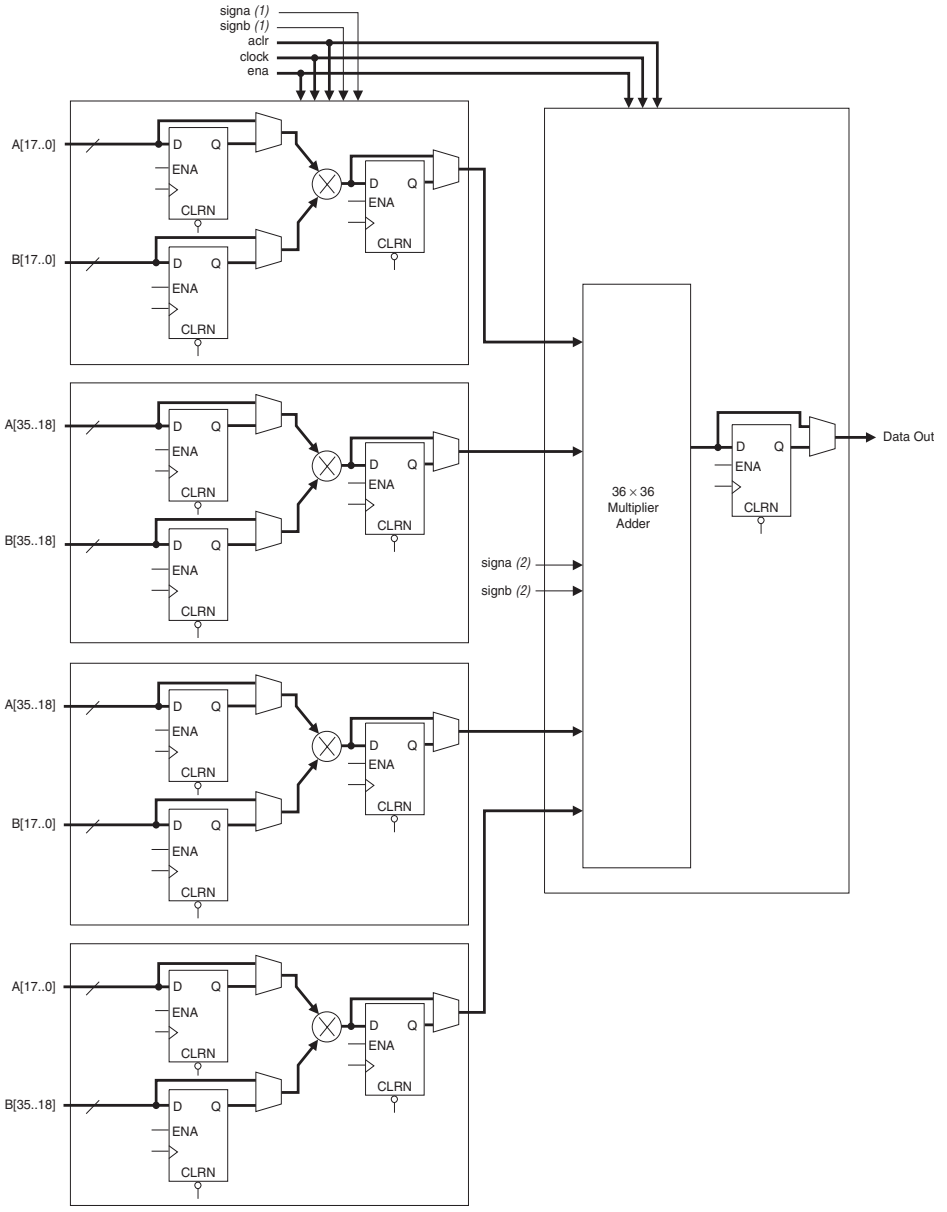
Figure 2–35. Simple Multiplier Mode



Note to Figure 2–35:
(1) These signals are not registered or registered once to match the data path pipeline.

DSP blocks can also implement one 36 × 36-bit multiplier in multiplier mode. DSP blocks use four 18 × 18-bit multipliers combined with dedicated adder and internal shift circuitry to achieve 36-bit multiplication. The input shift register feature is not available for the 36 × 36-bit multiplier. In 36 × 36-bit mode, the device can use the register that is normally a multiplier-result-output register as a pipeline stage for the 36 × 36-bit multiplier. Figure 2–36 shows the 36 × 36-bit multiply mode.

Figure 2–36. 36 × 36 Multiply Mode



Notes to Figure 2–36:

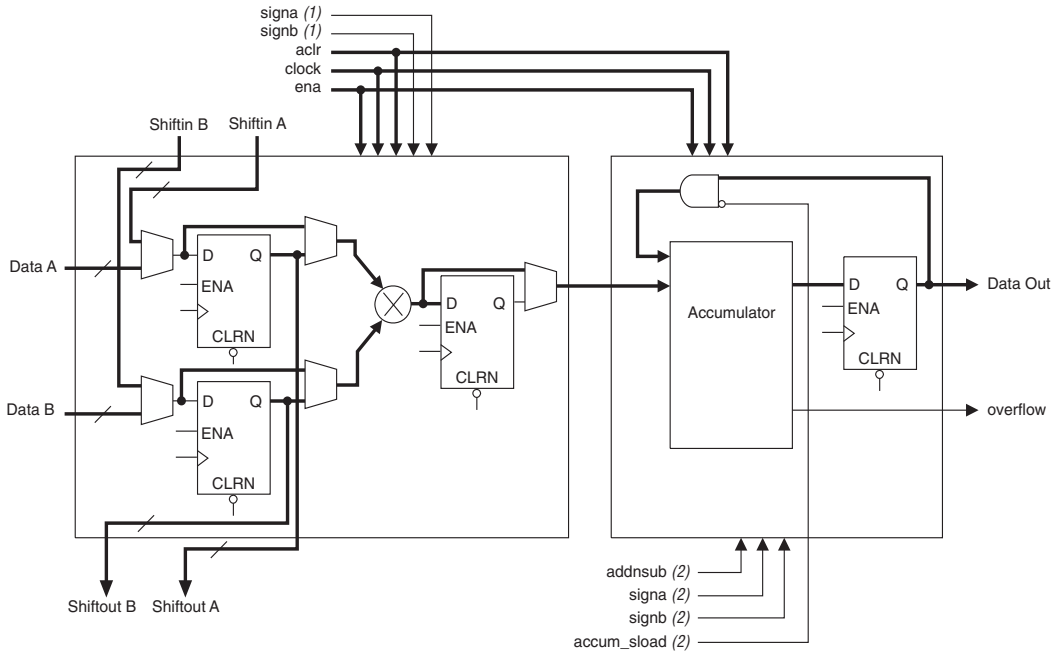
(1) These signals are not registered or registered once to match the pipeline.

(2) These signals are not registered, registered once, or registered twice for latency to match the pipeline.

Multiply-Accumulator Mode

In multiply-accumulator mode (see Figure 2–37), the DSP block drives multiplied results to the adder/subtractor/accumulator block configured as an accumulator. You can implement one or two multiply-accumulators up to 18 × 18 bits in one DSP block. The first and third multiplier sub-blocks are unused in this mode, because only one multiplier can feed one of two accumulators. The multiply-accumulator output can be up to 52 bits—a maximum of a 36-bit result with 16 bits of accumulation. The `accum_sload` and `overflow` signals are only available in this mode. The `addnsub` signal can set the accumulator for decimation and the `overflow` signal indicates underflow condition.

Figure 2–37. Multiply-Accumulate Mode



Notes to Figure 2–37:
(1) These signals are not registered or registered once to match the data path pipeline.
(2) These signals are not registered, registered once, or registered twice for latency to match the data path pipeline.

Two-Multipliers Adder Mode

The two-multipliers adder mode uses the adder/subtractor/accumulator block to add or subtract the outputs of the multiplier block, which is useful for applications such as FFT functions and complex FIR filters. A

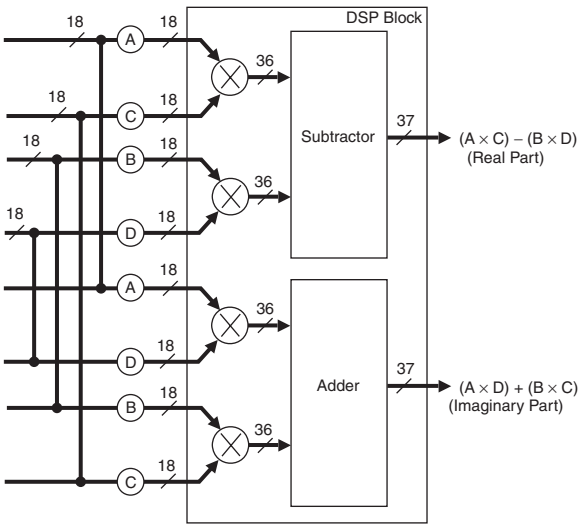
single DSP block can implement two sums or differences from two 18 × 18-bit multipliers each or four sums or differences from two 9 × 9-bit multipliers each.

You can use the two-multipliers adder mode for complex multiplications, which are written as:

$$(a + jb) \times (c + jd) = [(a \times c) - (b \times d)] + j \times [(a \times d) + (b \times c)]$$

The two-multipliers adder mode allows a single DSP block to calculate the real part [(a × c) – (b × d)] using one subtractor and the imaginary part [(a × d) + (b × c)] using one adder, for data widths up to 18 bits. Two complex multiplications are possible for data widths up to 9 bits using four adder/subtractor/accumulator blocks. Figure 2–38 shows an 18-bit two-multipliers adder.

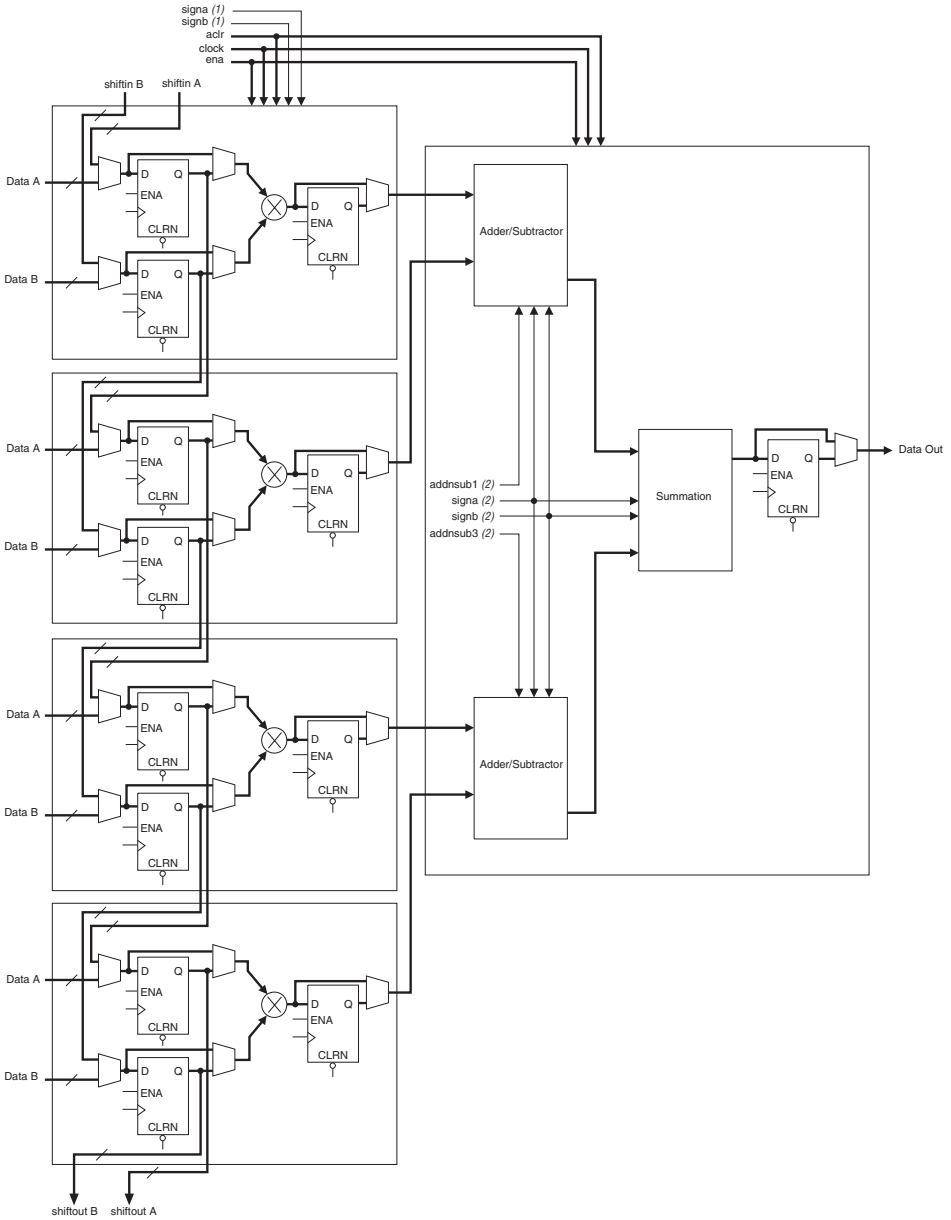
Figure 2–38. Two-Multipliers Adder Mode Implementing Complex Multiply



Four-Multipliers Adder Mode

In the four-multipliers adder mode, the DSP block adds the results of two first -stage adder/subtractor blocks. One sum of four 18 × 18-bit multipliers or two different sums of two sets of four 9 × 9-bit multipliers can be implemented in a single DSP block. The product width for each multiplier must be the same size. The four-multipliers adder mode is useful for FIR filter applications. Figure 2–39 shows the four multipliers adder mode.

Figure 2–39. Four-Multipliers Adder Mode



Notes to Figure 2–39:

(1) These signals are not registered or registered once to match the data path pipeline.

(2) These signals are not registered, registered once, or registered twice for latency to match the data path pipeline.

For FIR filters, the DSP block combines the four-multipliers adder mode with the shift register inputs. One set of shift inputs contains the filter data, while the other holds the coefficients loaded in serial or parallel. The input shift register eliminates the need for shift registers external to the DSP block (i.e., implemented in LEs). This architecture simplifies filter design since the DSP block implements all of the filter circuitry.

One DSP block can implement an entire 18-bit FIR filter with up to four taps. For FIR filters larger than four taps, DSP blocks can be cascaded with additional adder stages implemented in LEs.

Table 2–16 shows the different number of multipliers possible in each DSP block mode according to size. These modes allow the DSP blocks to implement numerous applications for DSP including FFTs, complex FIR, FIR, and 2D FIR filters, equalizers, IIR, correlators, matrix multiplication and many other functions.

Table 2–16. Multiplier Size & Configurations per DSP block

DSP Block Mode	9 × 9	18 × 18	36 × 36 *(1)*
Multiplier	Eight multipliers with eight product outputs	Four multipliers with four product outputs	One multiplier with one product output
Multiply-accumulator	Two multiply and accumulate (52 bits)	Two multiply and accumulate (52 bits)	–
Two-multipliers adder	Four sums of two multiplier products each	Two sums of two multiplier products each	–
Four-multipliers adder	Two sums of four multiplier products each	One sum of four multiplier products each	–

Note to Table 2–16:
(1) The number of supported multiply functions shown is based on signed/signed or unsigned/unsigned implementations.

DSP Block Interface

Stratix device DSP block outputs can cascade down within the same DSP block column. Dedicated connections between DSP blocks provide fast connections between the shift register inputs to cascade the shift register chains. You can cascade DSP blocks for 9 × 9- or 18 × 18-bit FIR filters larger than four taps, with additional adder stages implemented in LEs. If the DSP block is configured as 36 × 36 bits, the adder, subtractor, or accumulator stages are implemented in LEs. Each DSP block can route the shift register chain out of the block to cascade two full columns of DSP blocks.

The DSP block is divided into eight block units that interface with eight LAB rows on the left and right. Each block unit can be considered half of an 18 × 18-bit multiplier sub-block with 18 inputs and 18 outputs. A local interconnect region is associated with each DSP block. Like an LAB, this interconnect region can be fed with 10 direct link interconnects from the LAB to the left or right of the DSP block in the same row. All row and column routing resources can access the DSP block's local interconnect region. The outputs also work similarly to LAB outputs as well. Nine outputs from the DSP block can drive to the left LAB through direct link interconnects and nine can drive to the right LAB though direct link interconnects. All 18 outputs can drive to all types of row and column routing. Outputs can drive right- or left-column routing. Figures 2–40 and 2–41 show the DSP block interfaces to LAB rows.

Figure 2–40. DSP Block Interconnect Interface

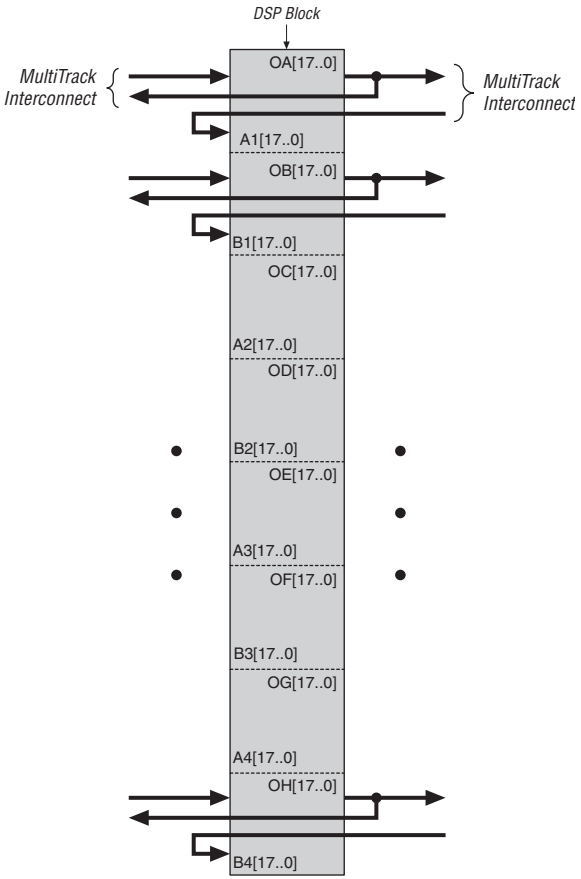
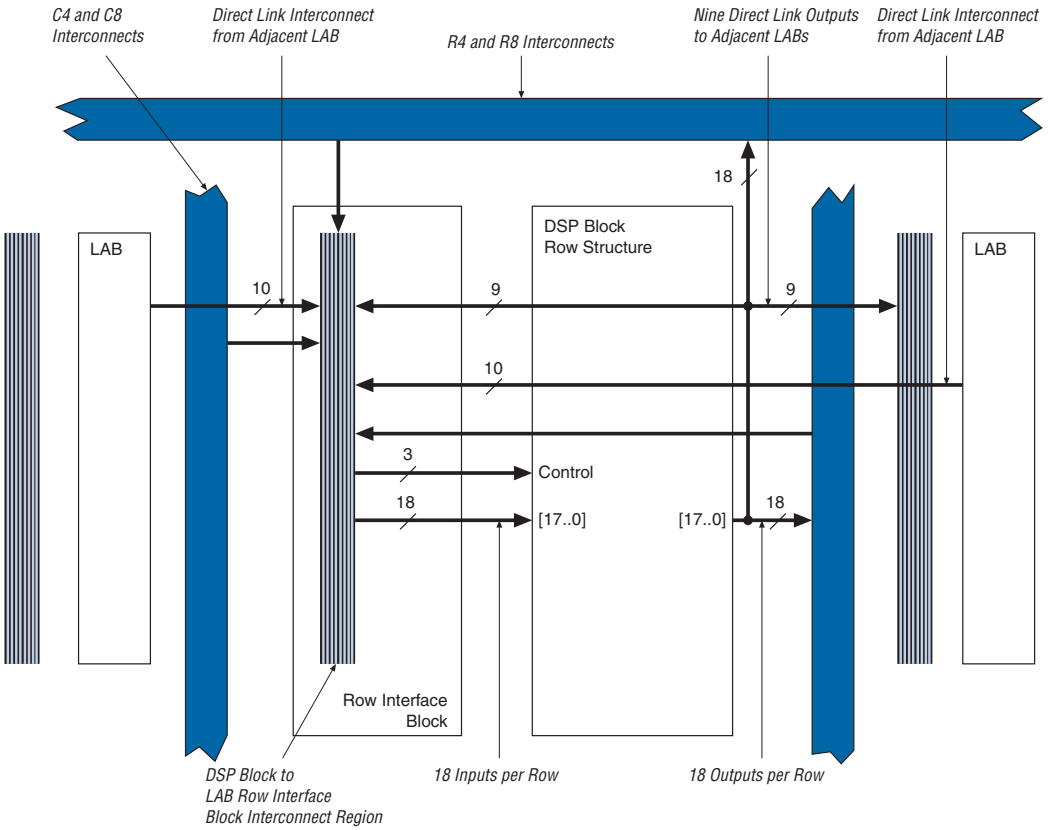


Figure 2–41. DSP Block Interface to Interconnect



A bus of 18 control signals feeds the entire DSP block. These signals include `clock[0..3]` clocks, `aclr[0..3]` asynchronous clears, `ena[1..4]` clock enables, `signa`, `signb` signed/unsigned control signals, `addnsub1` and `addnsub3` addition and subtraction control signals, and `accum_sload[0..1]` accumulator synchronous loads. The

clock signals are routed from LAB row clocks and are generated from specific LAB rows at the DSP block interface. The LAB row source for control signals, data inputs, and outputs is shown in Table 2–17.

Table 2–17. DSP Block Signal Sources & Destinations

LAB Row at Interface	Control Signals Generated	Data Inputs	Data Outputs
1	signa	A1[17..0]	OA[17..0]
2	aclr0 accum_sload0	B1[17..0]	OB[17..0]
3	addnsub1 clock0 ena0	A2[17..0]	OC[17..0]
4	aclr1 clock1 ena1	B2[17..0]	OD[17..0]
5	aclr2 clock2 ena2	A3[17..0]	OE[17..0]
6	sign_b clock3 ena3	B3[17..0]	OF[17..0]
7	clear3 accum_sload1	A4[17..0]	OG[17..0]
8	addnsub3	B4[17..0]	OH[17..0]

PLLs & Clock Networks

Stratix devices provide a hierarchical clock structure and multiple PLLs with advanced features. The large number of clocking resources in combination with the clock synthesis precision provided by enhanced and fast PLLs provides a complete clock management solution.

Global & Hierarchical Clocking

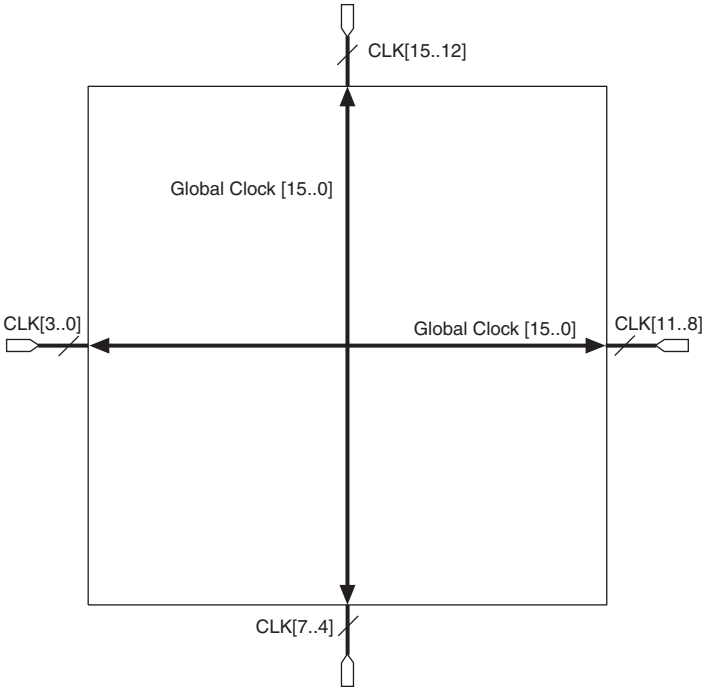
Stratix devices provide 16 dedicated global clock networks, 16 regional clock networks (four per device quadrant), and 8 dedicated fast regional clock networks (for EP1S10, EP1S20, and EP1S25 devices), and 16 dedicated fast regional clock networks (for EP1S30 EP1S40, and EP1S60, and EP1S80 devices). These clocks are organized into a hierarchical clock structure that allows for up to 22 clocks per device region with low skew and delay. This hierarchical clocking scheme provides up to 48 unique clock domains within Stratix devices.

There are 16 dedicated clock pins (CLK[15..0]) to drive either the global or regional clock networks. Four clock pins drive each side of the device, as shown in Figure 2–42. Enhanced and fast PLL outputs can also drive the global and regional clock networks.

Global Clock Network

These clocks drive throughout the entire device, feeding all device quadrants. The global clock networks can be used as clock sources for all resources within the device—IOEs, LEs, DSP blocks, and all memory blocks. These resources can also be used for control signals, such as clock enables and synchronous or asynchronous clears fed from the external pin. The global clock networks can also be driven by internal logic for internally generated global clocks and asynchronous clears, clock enables, or other control signals with large fanout. Figure 2–42 shows the 16 dedicated CLK pins driving global clock networks.

Figure 2–42. Global Clocking Note (1)



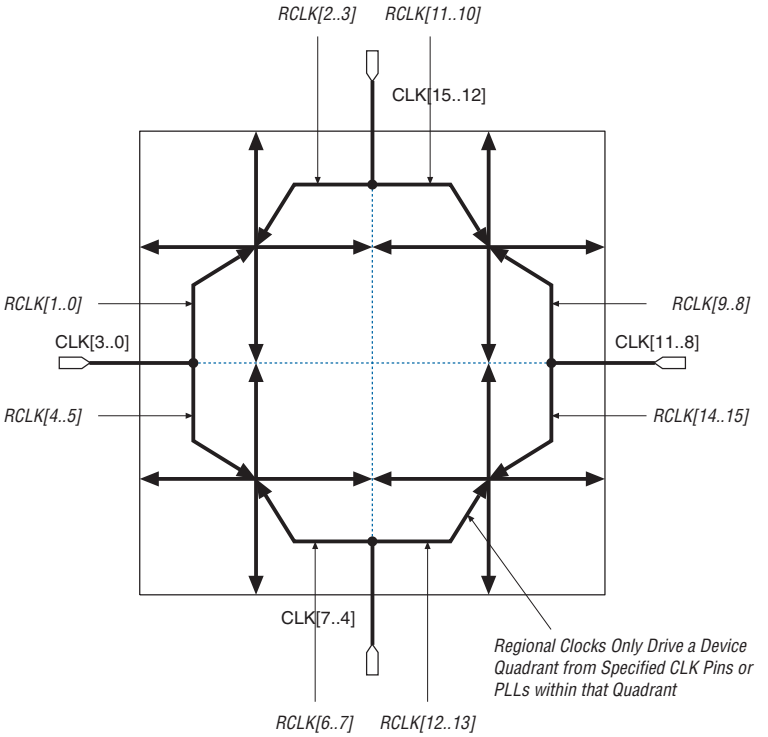
Note to Figure 2–42:

(1) The corner fast PLLs can also be driven through the global or regional clock networks. The global or regional clock input to the fast PLL can be driven by an output from another PLL, a pin-driven global or regional clock, or internally-generated global signals.

Regional Clock Network

There are four regional clock networks within each quadrant of the Stratix device that are driven by the same dedicated CLK[15..0] input pins or from PLL outputs. From a top view of the silicon, RCLK[0..3] are in the top left quadrant, RCLK[8..11] are in the top-right quadrant, RCLK[4..7] are in the bottom-left quadrant, and RCLK[12..15] are in the bottom-right quadrant. The regional clock networks only pertain to the quadrant they drive into. The regional clock networks provide the lowest clock delay and skew for logic contained within a single quadrant. RCLK cannot be driven by internal logic. The CLK clock pins symmetrically drive the RCLK networks within a particular quadrant, as shown in Figure 2–43. See Figures 2–50 and 2–51 for RCLK connections from PLLs and CLK pins.

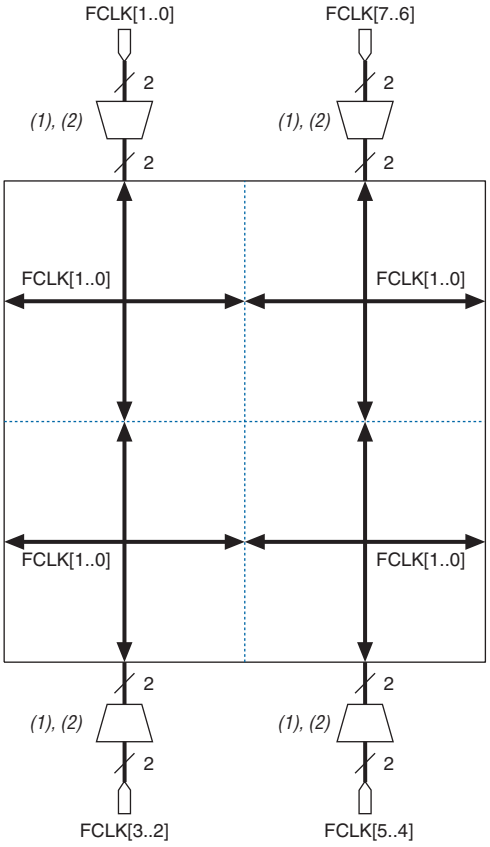
Figure 2–43. Regional Clocks



Fast Regional Clock Network

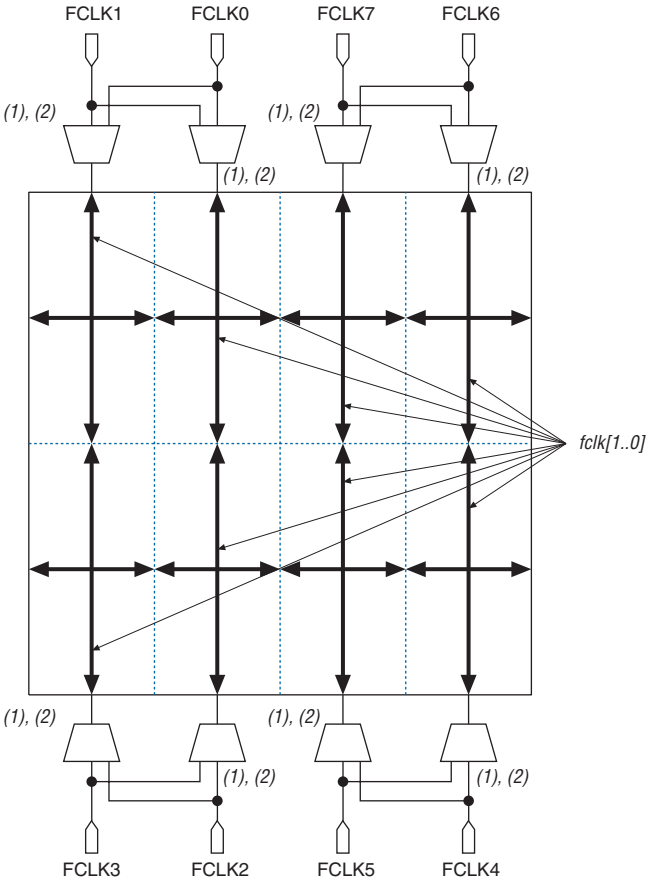
In EP1S25, EP1S20, and EP1S10 devices, there are two fast regional clock networks, FCLK[1..0], within each quadrant, fed by input pins that can connect to fast regional clock networks (see Figure 2–44). In EP1S30 and larger devices, there are two fast regional clock networks within each half-quadrant (see Figure 2–45). Dual-purpose FCLK pins drive the fast clock networks. All devices have eight FCLK pins to drive fast regional clock networks. Any I/O pin can drive a clock or control signal onto any fast regional clock network with the addition of a delay. This signal is driven via the I/O interconnect. The fast regional clock networks can also be driven from internal logic elements.

Figure 2–44. EP1S25, EP1S20 & EP1S10 Device Fast Clock Pin Connections to Fast Regional Clocks



Notes to Figure 2–44:
(1) This is a set of two multiplexers.
(2) In addition to the FCLK pin inputs, there is also an input from the I/O interconnect.

Figure 2–45. EP1S30 Device Fast Regional Clock Pin Connections to Fast Regional Clocks

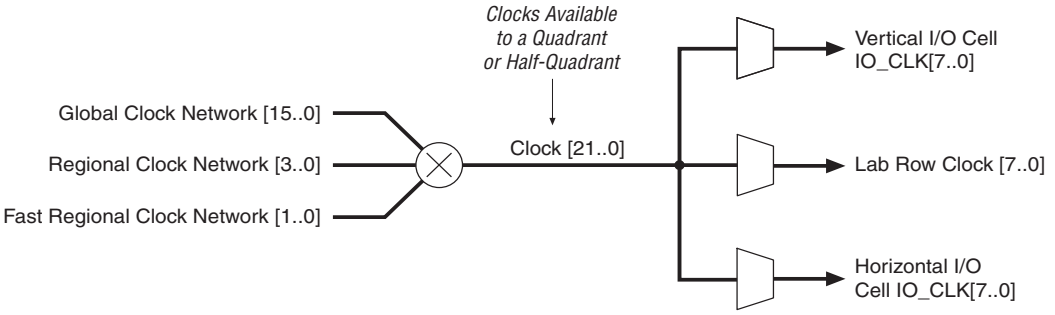


Notes to Figure 2–45:
(1) This is a set of two multiplexers.
(2) In addition to the FCLK pin inputs, there is also an input from the I/O interconnect.

Combined Resources

Within each region, there are 22 distinct dedicated clocking resources consisting of 16 global clock lines, four regional clock lines, and two fast regional clock lines. Multiplexers are used with these clocks to form eight bit busses to drive LAB row clocks, column IOE clocks, or row IOE clocks. Another multiplexer is used at the LAB level to select two of the eight row clocks to feed the LE registers within the LAB. See Figure 2–46.

Figure 2–46. Regional Clock Bus



IOE clocks have horizontal and vertical block regions that are clocked by eight I/O clock signals chosen from the 22 quadrant or half-quadrant clock resources. Figures 2–47 and 2–48 show the quadrant and half-quadrant relationship to the I/O clock regions, respectively. The vertical regions (column pins) have less clock delay than the horizontal regions (row pins).

Figure 2–47. EP1S10, EP1S20 & EP1S25 Device I/O Clock Groups

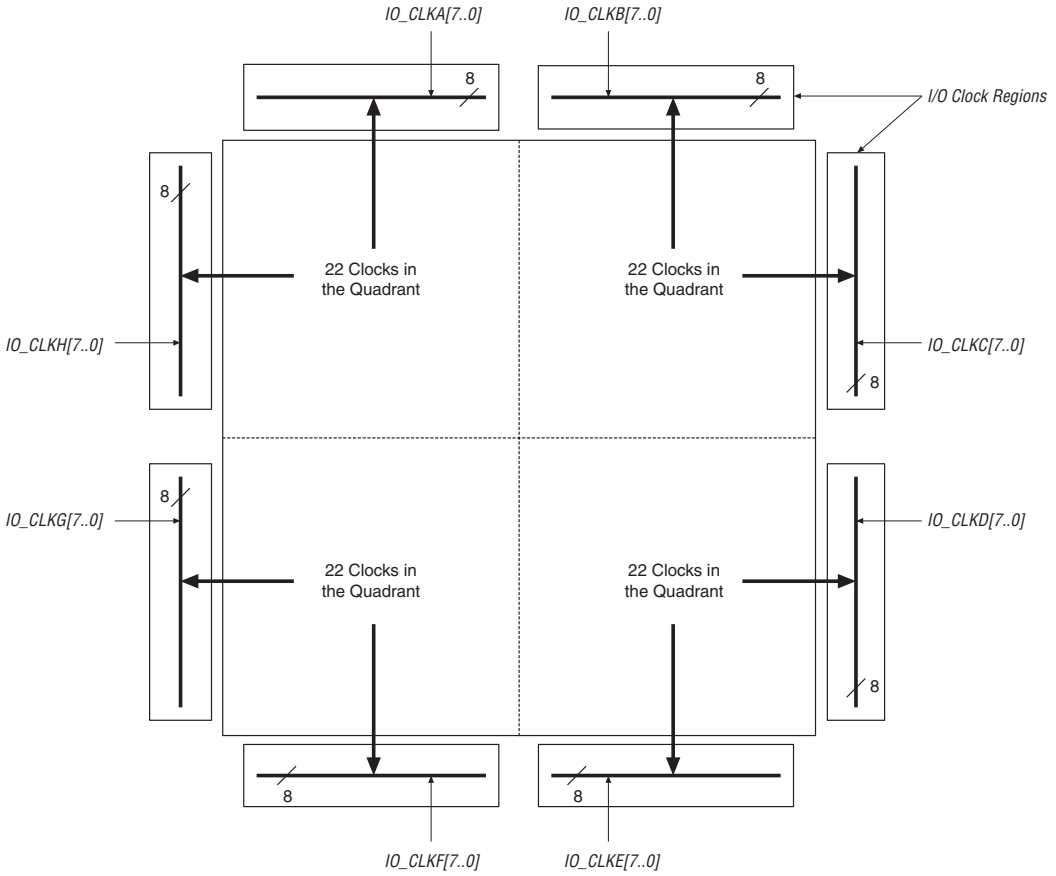
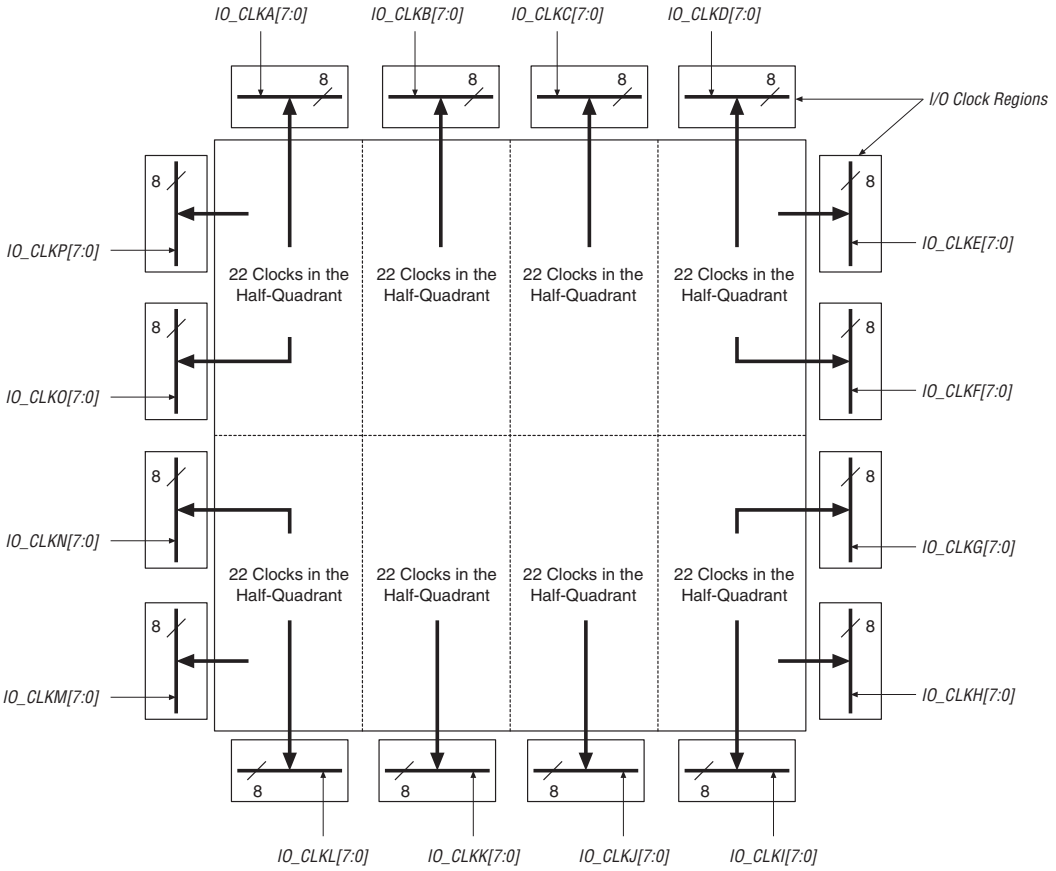


Figure 2–48. EP1S30, EP1S40, EP1S60, EP1S80 Device I/O Clock Groups



You can use the Quartus II software to control whether a clock input pin is either global, regional, or fast regional. The Quartus II software automatically selects the clocking resources if not specified.

Enhanced & Fast PLLs

Stratix devices provide robust clock management and synthesis using up to four enhanced PLLs and eight fast PLLs. These PLLs increase performance and provide advanced clock interfacing and clock-frequency synthesis. With features such as clock switchover, spread spectrum clocking, programmable bandwidth, phase and delay control, and PLL reconfiguration, the Stratix device's enhanced PLLs provide you with complete control of your clocks and system timing. The fast PLLs

provide general purpose clocking with multiplication and phase shifting as well as high-speed outputs for high-speed differential I/O support. Enhanced and fast PLLs work together with the Stratix high-speed I/O and advanced clock architecture to provide significant improvements in system performance and bandwidth.

The Quartus II software enables the PLLs and their features without requiring any external devices. Table 2–18 shows the PLLs available for each Stratix device.

Table 2–18. Stratix Device PLL Availability

Device	Fast PLLs								Enhanced PLLs			
	1	2	3	4	7	8	9	10	5 *(1)*	6 *(1)*	11 *(2)*	12 *(2)*
EP1S10	✓	✓	✓	✓					✓	✓		
EP1S20	✓	✓	✓	✓					✓	✓		
EP1S25	✓	✓	✓	✓					✓	✓		
EP1S30	✓	✓	✓	✓	✓ *(3)*	✓ *(3)*	✓ *(3)*	✓ *(3)*	✓	✓		
EP1S40	✓	✓	✓	✓	✓ *(3)*	✓ *(3)*	✓ *(3)*	✓ *(3)*	✓	✓	✓ *(3)*	✓ *(3)*
EP1S60	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP1S80	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notes to Table 2–18:

(1) PLLs 5 and 6 each have eight single-ended outputs or four differential outputs.

(2) PLLs 11 and 12 each have one single-ended output.

(3) EP1S30 and EP1S40 devices do not support these PLLs in the 780-pin FineLine BGA® package.

Table 2–19 shows the enhanced PLL and fast PLL features in Stratix devices.

Table 2–19. Stratix PLL Features

Feature	Enhanced PLL	Fast PLL
Clock multiplication and division	*m*/(*n* × post-scale counter) *(1)*	*m*/(post-scale counter) *(2)*
Phase shift	Down to 156.25-ps increments *(3)*, *(4)*	Down to 125-ps increments *(3)*, *(4)*
Delay shift	250-ps increments for ±3 ns	
Clock switchover	✓	
PLL reconfiguration	✓	
Programmable bandwidth	✓	
Spread spectrum clocking	✓	
Programmable duty cycle	✓	✓
Number of internal clock outputs	6	3 *(5)*
Number of external clock outputs	Four differential/eight singled-ended or one single-ended *(6)*	*(7)*
Number of feedback clock inputs	2 *(8)*	

Notes to Table 2–19:

(1) For enhanced PLLs, *m*, *n*, range from 1 to 512 and post-scale counters *g*, *l*, *e* range from 1 to 1024 with 50% duty cycle. With a non-50% duty cycle the post-scale counters *g*, *l*, *e* range from 1 to 512.
(2) For fast PLLs, *m* and post-scale counters range from 1 to 32.
(3) The smallest phase shift is determined by the voltage controlled oscillator (VCO) period divided by 8.
(4) For degree increments, Stratix devices can shift all output frequencies in increments of at least 45° . Smaller degree increments are possible depending on the frequency and divide parameters.
(5) PLLs 7, 8, 9, and 10 have two output ports per PLL. PLLs 1, 2, 3, and 4 have three output ports per PLL.
(6) Every Stratix device has two enhanced PLLs (PLLs 5 and 6) with either eight single-ended outputs or four differential outputs each. Two additional enhanced PLLs (PLLs 11 and 12) in EP1S80, EP1S60, and EP1S40 devices each have one single-ended output. Devices in the 780 pin FineLine BGA packages do not support PLLs 11 and 12.
(7) Fast PLLs can drive to any I/O pin as an external clock. For high-speed differential I/O pins, the device uses a data channel to generate `txclkout`.
(8) Every Stratix device has two enhanced PLLs with one single-ended or differential external feedback input per PLL.

Figure 2–49 shows a top-level diagram of the Stratix device and PLL floorplan.

Figure 2–49. PLL Locations

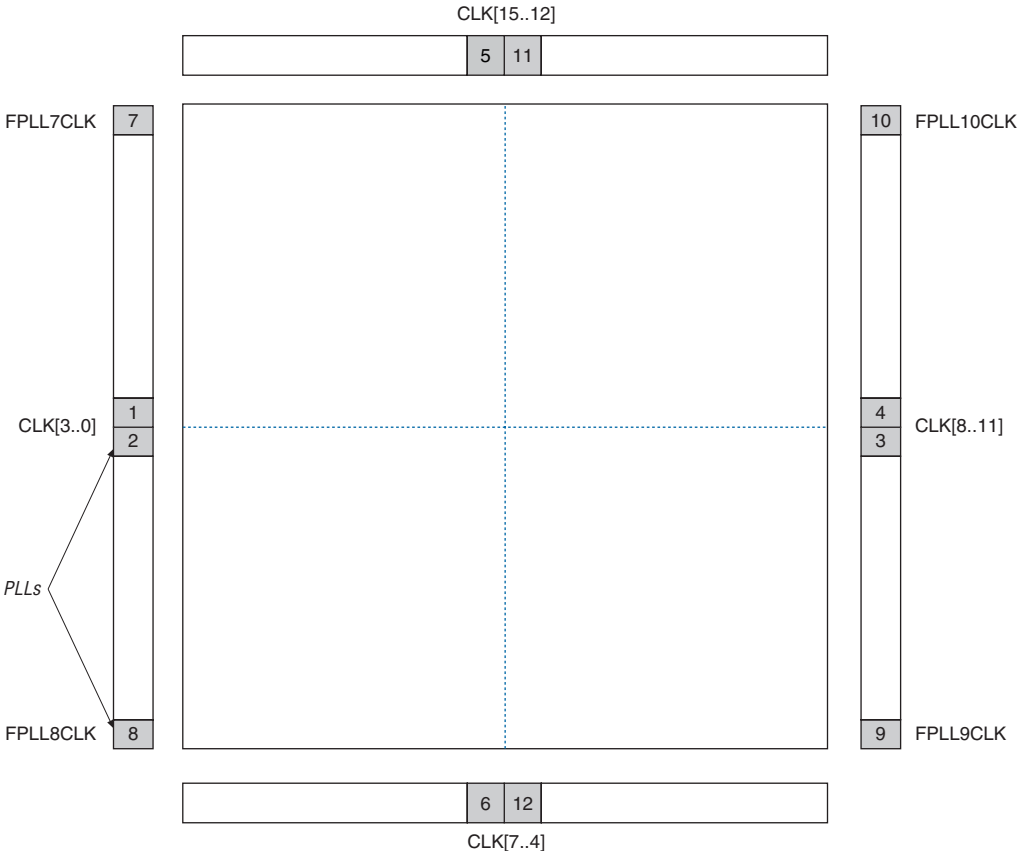
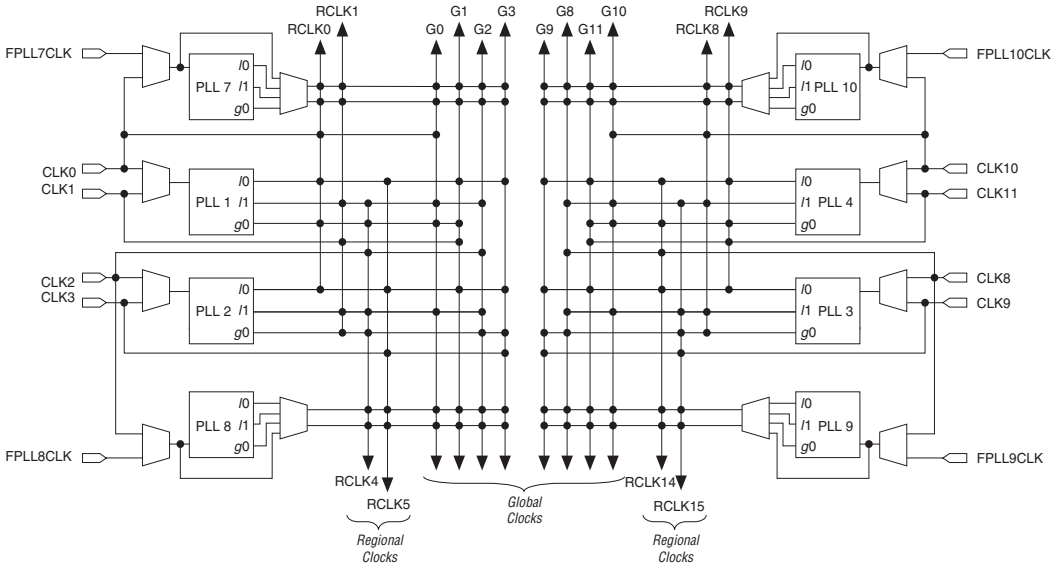


Figure 2–50 shows the global and regional clocking from the PLL outputs and the CLK pins.

Figure 2–50. Global & Regional Clock Connections from Side Pins & Fast PLL Outputs *Note (1), (2)*



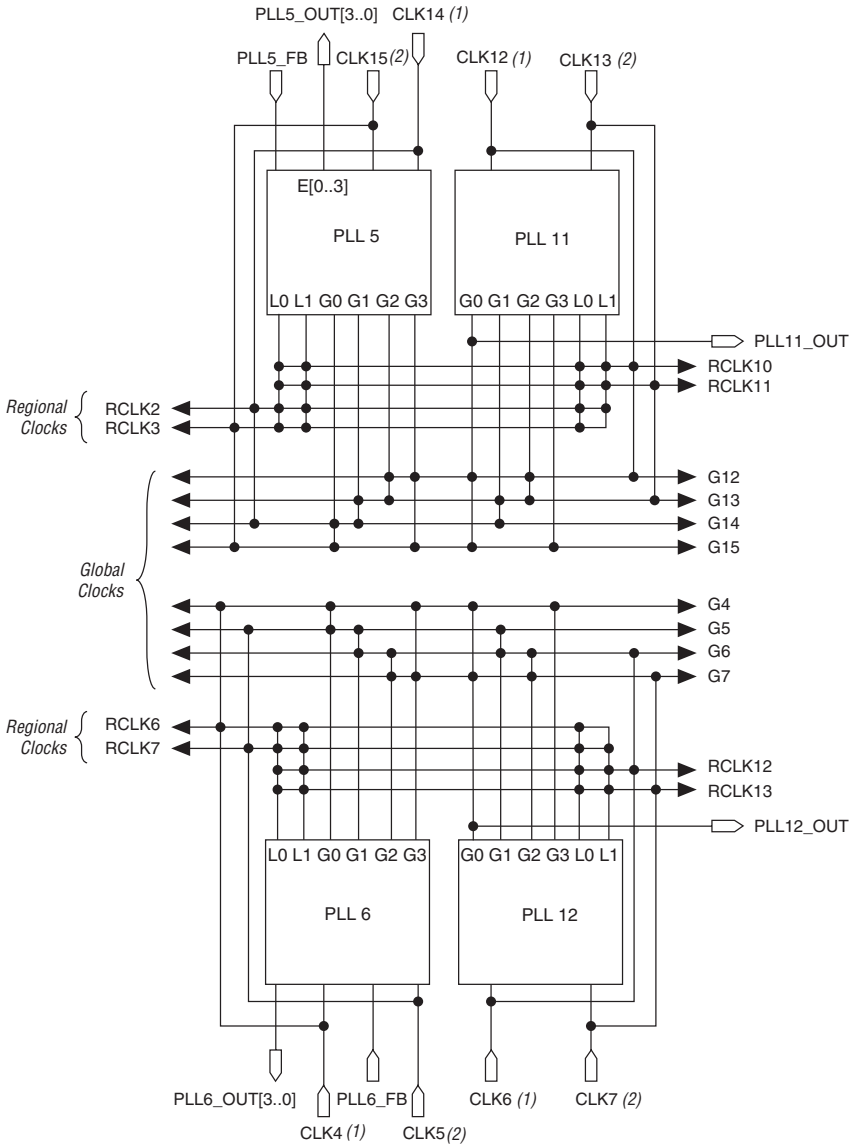
Notes to Figure 2–50:

(1) PLLs 1 to 4 and 7 to 10 are fast PLLs. PLLs 5, 6, 11, and 12 are enhanced PLLs.

(2) The global or regional clocks in a fast PLL's quadrant can drive the fast PLL input. A pin or other PLL must drive the global or regional source. The source cannot be driven by internally generated logic before driving the fast PLL.

Figure 2–51 shows the global and regional clocking from enhanced PLL outputs and top `CLK` pins.

Figure 2–51. Global & Regional Clock Connections from Top Clock Pins & Enhanced PLL Outputs Note (1)



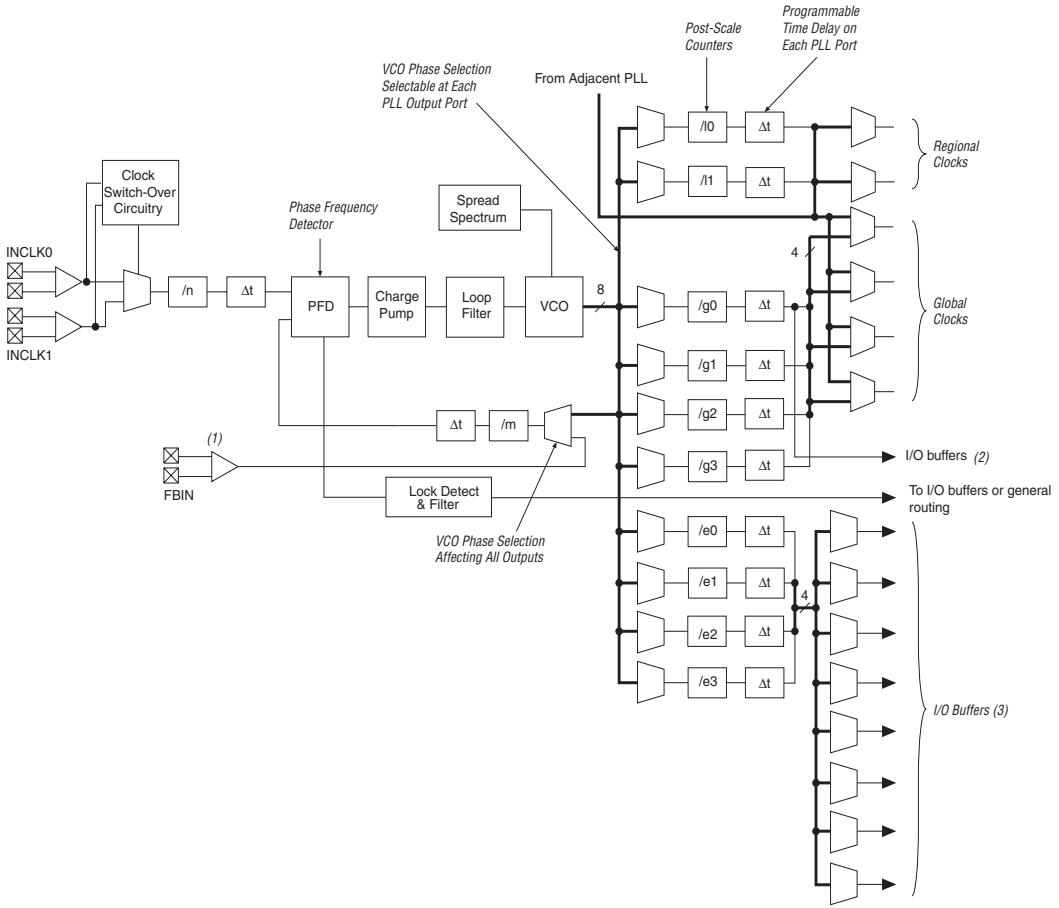
Notes to Figure 2–51:

(1) PLLs 1 to 4 and 7 to 10 are fast PLLs. PLLs 5, 6, 11, and 12 are enhanced PLLs.
(2) `CLK4`, `CLK6`, `CLK12`, and `CLK14` feed the corresponding PLL's `inclk0` port.
(3) `CLK5`, `CLK7`, `CLK13`, and `CLK15` feed the corresponding PLL's `inclk1` port.
(4) The EP1S40 device in the 780-pin FineLine BGA package does not support PLLs 11 and 12.

Enhanced PLLs

Stratix devices contain up to four enhanced PLLs with advanced clock management features. Figure 2–52 shows a diagram of the enhanced PLL.

Figure 2–52. Stratix Enhanced PLL



Notes to Figure 2–52:

(1) External feedback is available in PLLs 5 and 6.
(2) This single-ended external output is available from the *g*0 counter for PLLs 11 and 12.
(3) These four counters and external outputs are available in PLLs 5 and 6.
(4) This connection is only available on EP1S40 and larger Stratix devices. For example, PLLs 5 and 11 are adjacent and PLLs 6 and 12 are adjacent. The EP1S40 device in the 780-pin FineLine BGA package does not support PLLs 11 and 12.

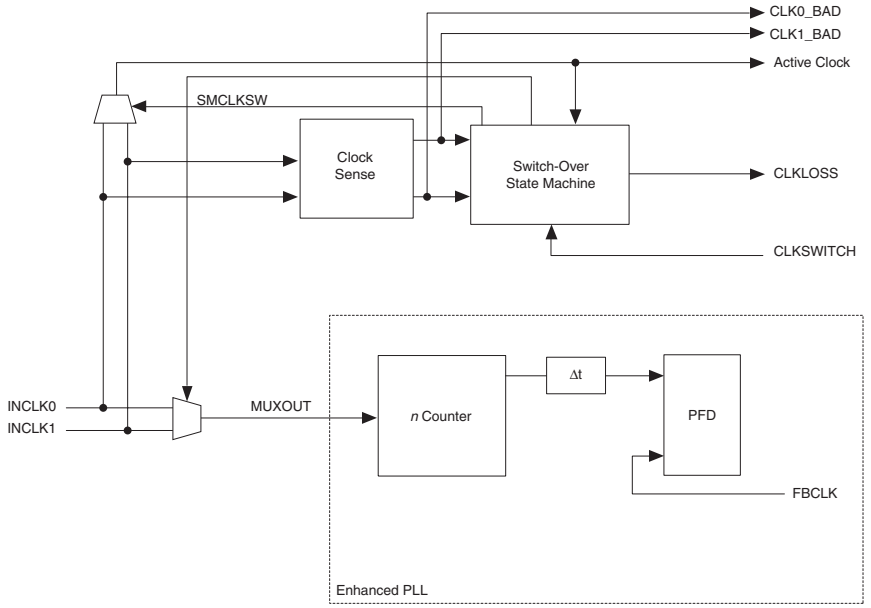
Clock Multiplication & Division

Each Stratix device enhanced PLL provides clock synthesis for PLL output ports using $m/(n \times$ post-scale counter) scaling factors. The input clock is divided by a pre-scale divider, n, and is then multiplied by the m feedback factor. The control loop drives the VCO to match $f_{IN} \times (m/n)$. Each output port has a unique post-scale counter that divides down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO is set to the least common multiple of the output frequencies that meets its frequency specifications. Then, the post-scale dividers scale down the output frequency for each output port. For example, if output frequencies required from one PLL are 33 and 66 MHz, set the VCO to 330 MHz (the least common multiple in the VCO's range). There is one pre-scale counter, n, and one multiply counter, m, per PLL, with a range of 1 to 512 on each. There are two post-scale counters (l) for regional clock output ports, four counters (g) for global clock output ports, and up to four counters (e) for external clock outputs, all ranging from 1 to 1024 with a 50% duty cycle setting. The post-scale counters range from 1 to 512 with any non-50% duty cycle setting. The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered.

Clock Switchover

To effectively develop high-reliability network systems, clocking schemes must support multiple clocks to provide redundancy. For this reason, Stratix device enhanced PLLs support a flexible clock switchover capability. Figure 2–53 shows a block diagram of the switchover circuit.The switchover circuit is configurable, so you can define how to implement it. Clock-sense circuitry automatically switches from the primary to secondary clock for PLL reference when the primary clock signal is not present.

Figure 2–53. Clock Switchover Circuitry



There are two possible ways to use the clock switchover feature.

- Use automatic switchover circuitry for switching between inputs of the same frequency. For example, in applications that require a redundant clock with the same frequency as the primary clock, the switchover state machine generates a signal that controls the multiplexer select input on the bottom of Figure 2–53. In this case, the secondary clock becomes the reference clock for the PLL.
- Use the `clkswitch` input for user- or system-controlled switch conditions. This is possible for same-frequency switchover or to switch between inputs of different frequencies. For example, if `inclk0` is 66 MHz and `inclk1` is 100 MHz, you must control the switchover because the automatic clock-sense circuitry cannot monitor primary and secondary clock frequencies with a frequency difference of more than ±20%. This feature is useful when clock sources can originate from multiple cards on the backplane, requiring a system-controlled switchover between frequencies of operation. You can use `clkswitch` together with the lock signal to trigger the switch from a clock that is running but becomes unstable and cannot be locked onto.

During switchover, the PLL VCO continues to run and will either slow down or speed up, generating frequency drift on the PLL outputs. The clock switchover transitions without any glitches. After the switch, there is a finite resynchronization period to lock onto new clock as the VCO ramps up. The exact amount of time it takes for the PLL to relock relates to the PLL configuration and may be adjusted by using the programmable bandwidth feature of the PLL. The specification for the maximum time to relock is 100 µs.

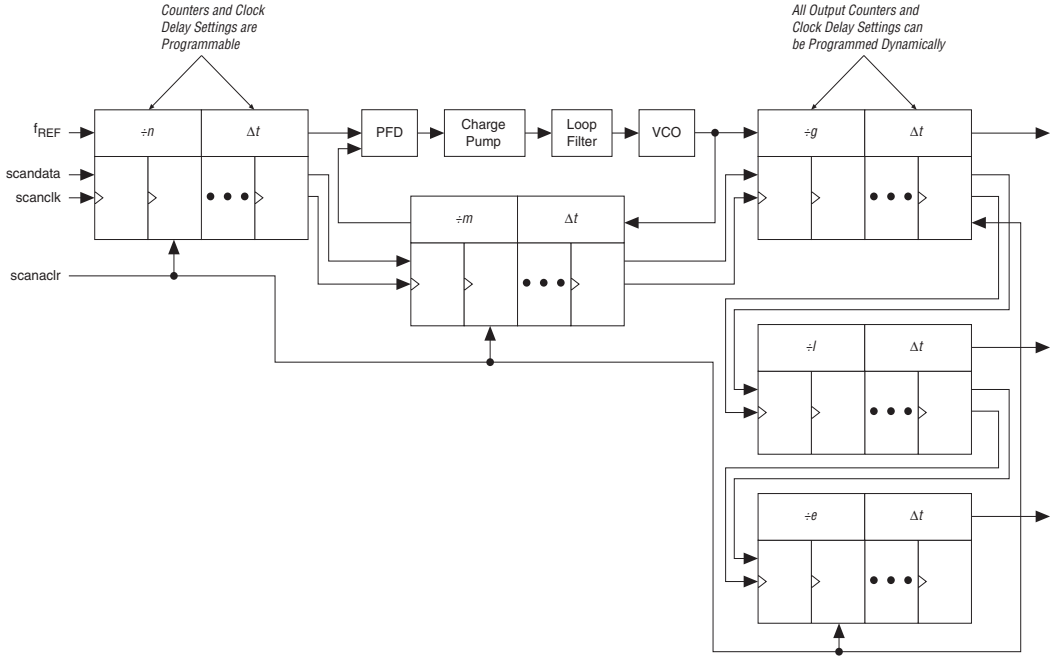
For more information on clock switchover, see *AN 313, Implementing Clock Switchover in Stratix & Stratix GX Devices*.

PLL Reconfiguration

The PLL reconfiguration feature enables system logic to change Stratix device enhanced PLL counters and delay elements without reloading a Programmer Object File (**.pof**). This provides considerable flexibility for frequency synthesis, allowing real-time PLL frequency and output clock delay variation. You can sweep the PLL output frequencies and clock delay in prototype environments. The PLL reconfiguration feature can also dynamically or intelligently control system clock speeds or t_{CO} delays in end systems.

Clock delay elements at each PLL output port implement variable delay. Figure 2–54 shows a diagram of the overall dynamic PLL control feature for the counters and the clock delay elements. The configuration time is less than 20 µs for the enhanced PLL using a input shift clock rate of 22 MHz. The charge pump, loop filter components, and phase shifting using VCO phase taps cannot be dynamically adjusted.

Figure 2–54. Dynamically Programmable Counters & Delays in Stratix Device Enhanced PLLs



PLL reconfiguration data is shifted into serial registers from the logic array or external devices. The PLL input shift data uses a reference input shift clock. Once the last bit of the serial chain is clocked in, the register chain is synchronously loaded into the PLL configuration bits. The shift circuitry also provides an asynchronous clear for the serial registers.

For more information on PLL reconfiguration, see *AN 282: Implementing PLL Reconfiguration in Stratix & Stratix GX Devices.*

Programmable Bandwidth

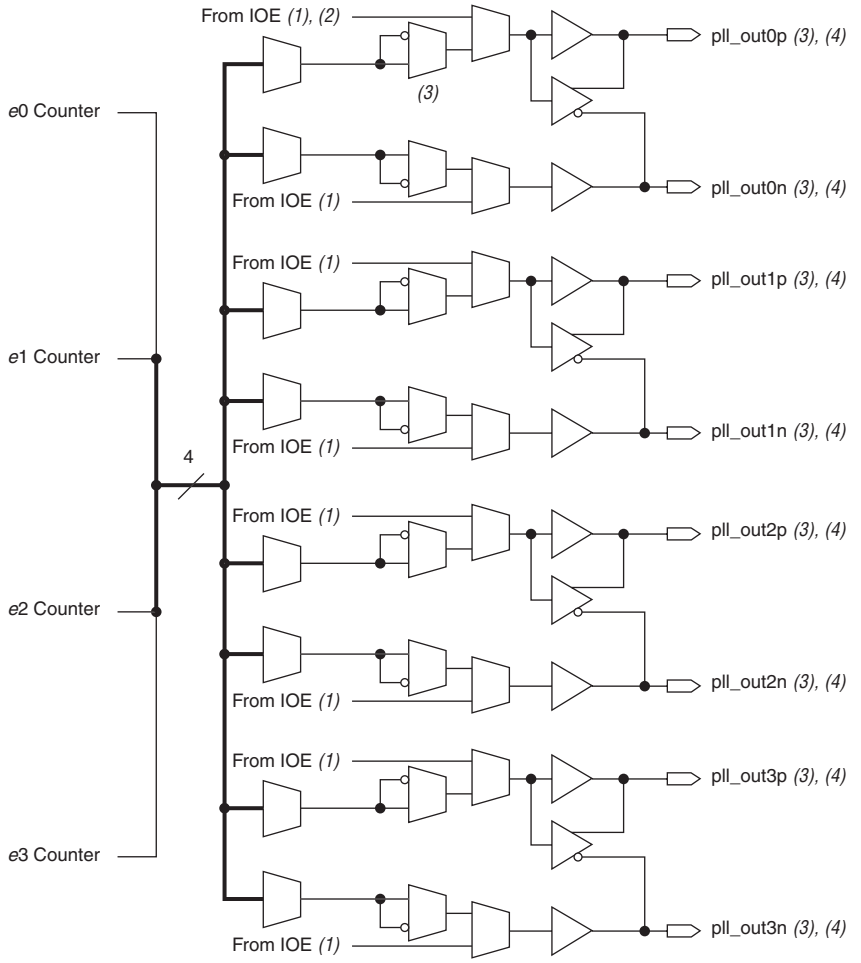
You have advanced control of the PLL bandwidth using the programmable control of the PLL loop characteristics, including loop filter and charge pump. The PLL's bandwidth is a measure of its ability to track the input clock and jitter. A high-bandwidth PLL can quickly lock onto a reference clock and react to any changes in the clock. It also will allow a wide band of input jitter spectrum to pass to the output. A low-bandwidth PLL will take longer to lock, but it will attenuate all high-frequency jitter components. The Quartus II software can adjust PLL characteristics to achieve the desired bandwidth. The programmable

bandwidth is tuned by varying the charge pump current, loop filter resistor value, high frequency capacitor value, and *m* counter value. You can manually adjust these values if desired. Bandwidth is programmable from 200 kHz to 1.5 MHz.

External Clock Outputs

Enhanced PLLs 5 and 6 each support up to eight single-ended clock outputs (or four differential pairs). Differential SSTL and HSTL outputs are implemented using 2 single-ended output buffers which are programmed to have opposite polarity. In Quartus II software, simply assign the appropriate differential I/O standard and the software will implement the inversion. See Figure 2–55.

Figure 2–55. External Clock Outputs for PLLs 5 & 6



Notes to Figure 2–55:

(1) The design can use each external clock output pin as a general-purpose output pin from the logic array. These pins are multiplexed with IOE outputs.

(2) Two single-ended outputs are possible per output counter—either two outputs of the same frequency and phase or one shifted 180° .

(3) EP1S10, EP1S20, and EP1S25 devices in 672-pin BGA and 484- and 672-pin FineLine BGA packages only have two pairs of external clocks (i.e., `pll_out0p`, `pll_out0n`, `pll_out1p`, and `pll_out1n`).

(4) Differential SSTL and HSTL outputs are implemented using two single-ended output buffers, which are programmed to have opposite polarity.

Any of the four external output counters can drive the single-ended or differential clock outputs for PLLs 5 and 6. This means one counter or frequency can drive all output pins available from PLL 5 or PLL 6. Each pair of output pins (four pins total) has dedicated VCC and GND pins to reduce the output clock's overall jitter by providing improved isolation from switching I/O pins.

For PLLs 5 and 6, each pin of a single-ended output pair can either be in phase or 180° out of phase. 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, 3.3-V PCML, HyperTransport technology, differential HSTL, and differential SSTL. Table 2–20 shows which I/O standards the enhanced PLL clock pins support. When in single-ended or differential mode, the two outputs operate off the same power supply. Both outputs use the same standards in single-ended mode to maintain performance. You can also use the external clock output pins as user output pins if external enhanced PLL clocking is not needed.

Table 2–20. I/O Standards Supported for Enhanced PLL Pins (Part 1 of 2)

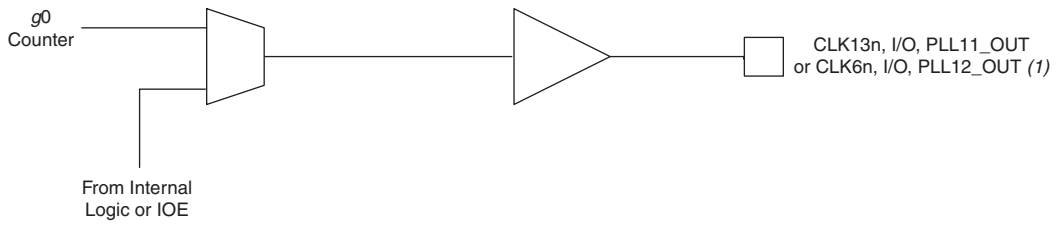
I/O Standard	Input			Output
	INCLK	FBIN	PLLENABLE	EXTCLK
LVTTL	✓	✓	✓	✓
LVCMOS	✓	✓	✓	✓
2.5 V	✓	✓		✓
1.8 V	✓	✓		✓
1.5 V	✓	✓		✓
3.3-V PCI	✓	✓		✓
3.3-V PCI-X 1.0	✓	✓		✓
LVPECL	✓	✓		✓
3.3-V PCML	✓	✓		✓
LVDS	✓	✓		✓
HyperTransport technology	✓	✓		✓
Differential HSTL	✓			✓
Differential SSTL				✓
3.3-V GTL	✓	✓		✓
3.3-V GTL+	✓	✓		✓
1.5-V HSTL Class I	✓	✓		✓

Table 2–20. I/O Standards Supported for Enhanced PLL Pins (Part 2 of 2)

I/O Standard	Input			Output
	INCLK	FBIN	PLLENABLE	EXTCLK
1.5-V HSTL Class II	✓	✓		✓
1.8-V HSTL Class I	✓	✓		✓
1.8-V HSTL Class II	✓	✓		✓
SSTL-18 Class I	✓	✓		✓
SSTL-18 Class II	✓	✓		✓
SSTL-2 Class I	✓	✓		✓
SSTL-2 Class II	✓	✓		✓
SSTL-3 Class I	✓	✓		✓
SSTL-3 Class II	✓	✓		✓
AGP (1× and 2×)	✓	✓		✓
CTT	✓	✓		✓

Enhanced PLLs 11 and 12 support one single-ended output each (see Figure 2–56). These outputs do not have their own VCC and GND signals. Therefore, to minimize jitter, do not place switching I/O pins next to this output pin.

Figure 2–56. External Clock Outputs for Enhanced PLLs 11 & 12



Note to Figure 2–56:
(1) For PLL 11, this pin is CLK13n; for PLL 12 this pin is CLK7n.

Stratix devices can drive any enhanced PLL driven through the global clock or regional clock network to any general I/O pin as an external output clock. The jitter on the output clock is not guaranteed for these cases.

Clock Feedback

The following four feedback modes in Stratix device enhanced PLLs allow multiplication and/or phase and delay shifting:

- Zero delay buffer: The external clock output pin is phase-aligned with the clock input pin for zero delay. Altera recommends using the same I/O standard on the input clock and the output clocks for optimum performance.
- External feedback: The external feedback input pin, FBIN, is phase-aligned with the clock input, CLK, pin. Aligning these clocks allows you to remove clock delay and skew between devices. This mode is only possible for PLLs 5 and 6. PLLs 5 and 6 each support feedback for one of the dedicated external outputs, either one single-ended or one differential pair. In this mode, one *e* counter feeds back to the PLL FBIN input, becoming part of the feedback loop. Altera recommends using the same I/O standard on the input clock, the FBIN pin, and the output clocks for optimum performance.
- Normal mode: If an internal clock is used in this mode, it is phase-aligned to the input clock pin. The external clock output pin will have a phase delay relative to the clock input pin if connected in this mode. You define which internal clock output from the PLL should be phase-aligned to the internal clock pin.
- No compensation: In this mode, the PLL will not compensate for any clock networks or external clock outputs.

Phase & Delay Shifting

Stratix device enhanced PLLs provide advanced programmable phase and clock delay shifting. These parameters are set in the Quartus II software.

Phase Delay

The Quartus II software automatically sets the phase taps and counter settings according to the phase shift entry. You enter a desired phase shift and the Quartus II software automatically sets the closest setting achievable. This type of phase shift is not reconfigurable during system operation. For phase shifting, enter a phase shift (in degrees or time units) for each PLL clock output port or for all outputs together in one shift. You can select phase-shifting values in time units with a resolution of 156.25 to 416.66 ps. This resolution is a function of frequency input and the multiplication and division factors (that is, it is a function of the VCO period), with the finest step being equal to an eighth (×0.125) of the VCO period. Each clock output counter can choose a different phase of the

VCO period from up to eight taps for individual fine step selection. Also, each clock output counter can use a unique initial count setting to achieve individual coarse shift selection in steps of one VCO period. The combination of coarse and fine shifts allows phase shifting for the entire input clock period.

The equation to determine the precision of the phase shifting in degrees is: 45° ÷ post-scale counter value. Therefore, the maximum step size is 45°, and smaller steps are possible depending on the multiplication and division ratio necessary on the output counter port.

This type of phase shift provides the highest precision since it is the least sensitive to process, supply, and temperature variation.

Clock Delay

In addition to the phase shift feature, the ability to fine tune the Δt clock delay provides advanced time delay shift control on each of the four PLL outputs. There are time delays for each post-scale counter (*e*, *g*, or *l*) from the PLL, the *n* counter, and *m* counter. Each of these can shift in 250-ps increments for a range of 3.0 ns. The *m* delay shifts all outputs earlier in time, while *n* delay shifts all outputs later in time. Individual delays on post-scale counters (*e*, *g*, and *l*) provide positive delay for each output. Table 2–21 shows the combined delay for each output for normal or zero delay buffer mode where Δt_e, Δt_g, or Δt_l is unique for each PLL output.

The t_{OUTPUT} for a single output can range from –3 ns to +6 ns. The total delay shift difference between any two PLL outputs, however, must be less than ±3 ns. For example, shifts on two outputs of –1 and +2 ns is allowed, but not –1 and +2.5 ns because these shifts would result in a difference of 3.5 ns. If the design uses external feedback, the Δt_e delay will remove delay from outputs, represented by a negative sign (see Table 2–21). This effect occurs because the Δt_e delay is then part of the feedback loop.

Table 2–21. Output Clock Delay for Enhanced PLLs

Normal or Zero Delay Buffer Mode	External Feedback Mode
$\Delta t_{eOUTPUT} = \Delta t_n - \Delta t_m + \Delta t_e$ $\Delta t_{gOUTPUT} = \Delta t_n - \Delta t_m + \Delta t_g$ $\Delta t_{lOUTPUT} = \Delta t_n - \Delta t_m + \Delta t_l$	$\Delta t_{eOUTPUT} = \Delta t_n - \Delta t_m - \Delta t_e$ *(1)* $\Delta t_{gOUTPUT} = \Delta t_n - \Delta t_m + \Delta t_g$ $\Delta t_{lOUTPUT} = \Delta t_n - \Delta t_m + \Delta t_l$

Note to Table 2–21:

(1) Δt_e removes delay from outputs in external feedback mode.

The variation due to process, voltage, and temperature is about ±15% on the delay settings. PLL reconfiguration can control the clock delay shift elements, but not the VCO phase shift multiplexers, during system operation.

Spread-Spectrum Clocking

Stratix device enhanced PLLs use spread-spectrum technology to reduce electromagnetic interference generation from a system by distributing the energy over a broader frequency range. The enhanced PLL typically provides 0.5% down spread modulation using a triangular profile. The modulation frequency is programmable. Enabling spread-spectrum for a PLL affects all of its outputs.

Lock Detect

The lock output indicates that there is a stable clock output signal in phase with the reference clock. Without any additional circuitry, the lock signal may toggle as the PLL begins tracking the reference clock. You may need to gate the lock signal for use as a system control. The lock signal from the locked port can drive the logic array or an output pin.

Whenever the PLL loses lock (for example, `inclk` jitter, clock switchover, PLL reconfiguration, power supply noise, and so on), the PLL must be reset with the `areset` signal to guarantee correct phase relationship between the PLL output clocks. If the phase relationship between the input clock versus output clock, and between different output clocks from the PLL is not important in the design, then the PLL need not be reset.

See the *Stratix FPGA Errata Sheet* for more information on implementing the gated lock signal in a design.

Programmable Duty Cycle

The programmable duty cycle allows enhanced PLLs to generate clock outputs with a variable duty cycle. This feature is supported on each enhanced PLL post-scale counter (*g*0..*g*3, *l*0..*l*3, *e*0..*e*3). The duty cycle setting is achieved by a low and high time count setting for the post-scale dividers. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices.

Advanced Clear & Enable Control

There are several control signals for clearing and enabling PLLs and their outputs. You can use these signals to control PLL resynchronization and gate PLL output clocks for low-power applications.

The pllenable pin is a dedicated pin that enables/disables PLLs. When the pllenable pin is low, the clock output ports are driven by GND and all the PLLs go out of lock. When the pllenable pin goes high again, the PLLs relock and resynchronize to the input clocks. You can choose which PLLs are controlled by the pllenable signal by connecting the pllenable input port of the altpll megafunction to the common pllenable input pin.

The areset signals are reset/resynchronization inputs for each PLL. The areset signal should be asserted every time the PLL loses lock to guarantee correct phase relationship between the PLL output clocks. Users should include the areset signal in designs if any of the following conditions are true:

- PLL Reconfiguration or Clock switchover enables in the design.
- Phase relationships between output clocks need to be maintained after a loss of lock condition

The device input pins or logic elements (LEs) can drive these input signals. When driven high, the PLL counters will reset, clearing the PLL output and placing the PLL out of lock. The VCO will set back to its nominal setting (~700 MHz). When driven low again, the PLL will resynchronize to its input as it relocks. If the target VCO frequency is below this nominal frequency, then the output frequency will start at a higher value than desired as the PLL locks. If the system cannot tolerate this, the clkena signal can disable the output clocks until the PLL locks.

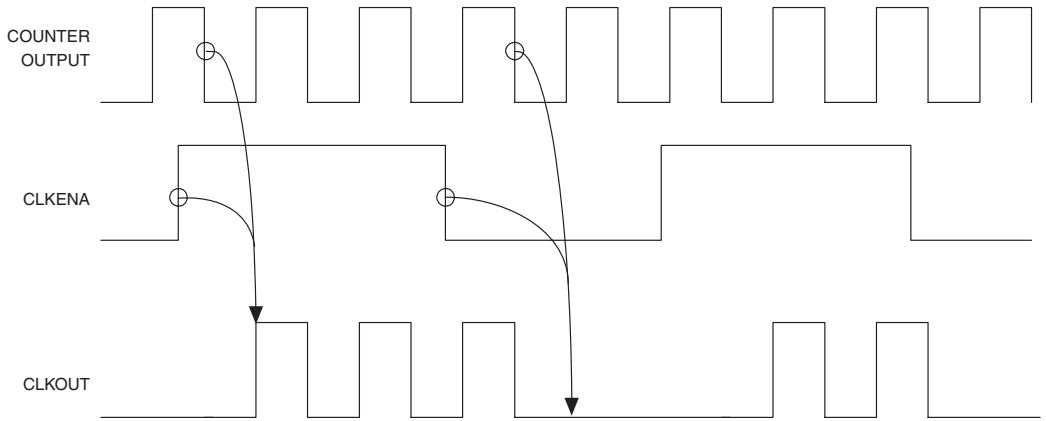
The pfdena signals control the phase frequency detector (PFD) output with a programmable gate. If you disable the PFD, the VCO operates at its last set value of control voltage and frequency with some long-term drift to a lower frequency. The system continues running when the PLL goes out of lock or the input clock is disabled. By maintaining the last locked frequency, the system has time to store its current settings before shutting down. You can either use your own control signal or a clkloss status signal to trigger pfdena.

The clkena signals control the enhanced PLL regional and global outputs. Each regional and global output port has its own clkena signal. The clkena signals synchronously disable or enable the clock at the PLL output port by gating the outputs of the *g* and *l* counters. The clkena signals are registered on the falling edge of the counter output clock to enable or disable the clock without glitches. Figure 2–57 shows the waveform example for a PLL clock port enable. The PLL can remain locked independent of the clkena signals since the loop-related counters are not affected. This feature is useful for applications that require a low power or sleep mode. Upon re-enabling, the PLL does not need a

resynchronization or relock period. The clkena signal can also disable clock outputs if the system is not tolerant to frequency overshoot during resynchronization.

The extclkena signals work in the same way as the clkena signals, but they control the external clock output counters (*e*0, *e*1, *e*2, and *e*3). Upon re-enabling, the PLL does not need a resynchronization or relock period unless the PLL is using external feedback mode. In order to lock in external feedback mode, the external output must drive the board trace back to the FBIN pin.

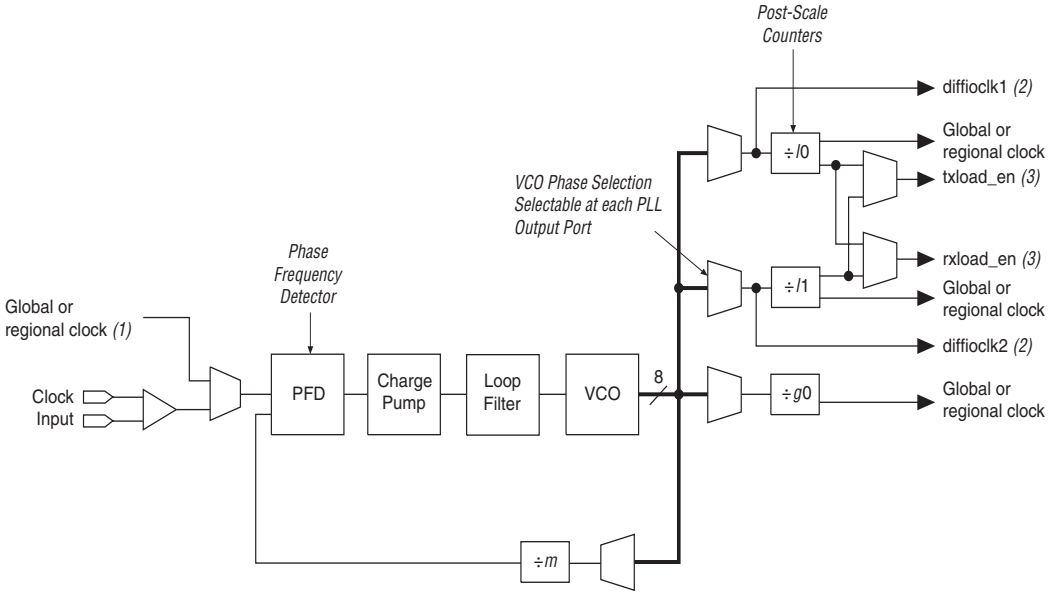
Figure 2–57. extclkena Signals



Fast PLLs

Stratix devices contain up to eight fast PLLs with high-speed serial interfacing ability, along with general-purpose features. Figure 2–58 shows a diagram of the fast PLL.

Figure 2–58. Stratix Device Fast PLL



Notes to Figure 2–58:

(1) The global or regional clock input can be driven by an output from another PLL or any dedicated CLK or FCLK pin. It cannot be driven by internally-generated global signals.

(2) In high-speed differential I/O support mode, this high-speed PLL clock feeds the SERDES. Stratix devices only support one rate of data transfer per fast PLL in high-speed differential I/O support mode.

(3) This signal is a high-speed differential I/O support SERDES control signal.

Clock Multiplication & Division

Stratix device fast PLLs provide clock synthesis for PLL output ports using *m*/(post scaler) scaling factors. The input clock is multiplied by the *m* feedback factor. Each output port has a unique post scale counter to divide down the high-frequency VCO. There is one multiply divider, *m*, per fast PLL with a range of 1 to 32. There are two post scale L dividers for regional and/or LVDS interface clocks, and *g*0 counter for global clock output port; all range from 1 to 32.

In the case of a high-speed differential interface, set the output counter to 1 to allow the high-speed VCO frequency to drive the SERDES. When used for clocking the SERDES, the *m* counter can range from 1 to 30. The VCO frequency is equal to $f_{IN} \times m$, where VCO frequency must be between 300 and 1000 MHz.

External Clock Inputs

Each fast PLL supports single-ended or differential inputs for source synchronous transmitters or for general-purpose use. Source-synchronous receivers support differential clock inputs. The fast PLL inputs are fed by CLK[0..3], CLK[8..11], and FPLL[7..10]CLK pins, as shown in Figure 2–50 on page 2–85.

Table 2–22 shows the I/O standards supported by fast PLL input pins.

Table 2–22. Fast PLL Port I/O Standards (Part 1 of 2)

I/O Standard	Input	
	INCLK	PLLENABLE
LVTTL	✓	✓
LVCMOS	✓	✓
2.5 V	✓	
1.8 V	✓	
1.5 V	✓	
3.3-V PCI		
3.3-V PCI-X 1.0		
LVPECL	✓	
3.3-V PCML	✓	
LVDS	✓	
HyperTransport technology	✓	
Differential HSTL	✓	
Differential SSTL		
3.3-V GTL		
3.3-V GTL+	✓	
1.5-V HSTL Class I	✓	
1.5-V HSTL Class II		
1.8-V HSTL Class I	✓	
1.8-V HSTL Class II		
SSTL-18 Class I	✓	
SSTL-18 Class II		
SSTL-2 Class I	✓	

Table 2–22. Fast PLL Port I/O Standards (Part 2 of 2)		
I/O Standard	Input	
	INCLK	PLLENABLE
SSTL-2 Class II	✓	
SSTL-3 Class I	✓	
SSTL-3 Class II	✓	
AGP (1× and 2×)		
CTT	✓	

Table 2–23 shows the performance on each of the fast PLL clock inputs when using LVDS, LVPECL, 3.3-V PCML, or HyperTransport technology.

Table 2–23. LVDS Performance on Fast PLL Input	
Fast PLL Clock Input	Maximum Input Frequency (MHz)
CLK0, CLK2, CLK9, CLK11, FPLL7CLK, FPLL8CLK, FPLL9CLK, FPLL10CLK	717 *(1)*
CLK1, CLK3, CLK8, CLK10	645

Note to Table 2–23:

(1) See the chapter *DC & Switching Characteristics* of the *Stratix Device Handbook, Volume 1* for more information.

External Clock Outputs

Each fast PLL supports differential or single-ended outputs for source-synchronous transmitters or for general-purpose external clocks. There are no dedicated external clock output pins. Any I/O pin can be driven by the fast PLL global or regional outputs as an external output pin. The I/O standards supported by any particular bank determines what standards are possible for an external clock output driven by the fast PLL in that bank.

Phase Shifting

Stratix device fast PLLs have advanced clock shift capability that enables programmable phase shifts. You can enter a phase shift (in degrees or time units) for each PLL clock output port or for all outputs together in one shift. You can perform phase shifting in time units with a resolution range of 125 to 416.66 ps. This resolution is a function of the VCO period, with the finest step being equal to an eighth (×0.125) of the VCO period.

Control Signals

The fast PLL has the same `lock` output, `pllenable` input, and `areset` input control signals as the enhanced PLL.

If the input clock stops and causes the PLL to lose lock, then the PLL must be reset for correct phase shift operation.

For more information on high-speed differential I/O support, see "High-Speed Differential I/O Support" on page 2–130.

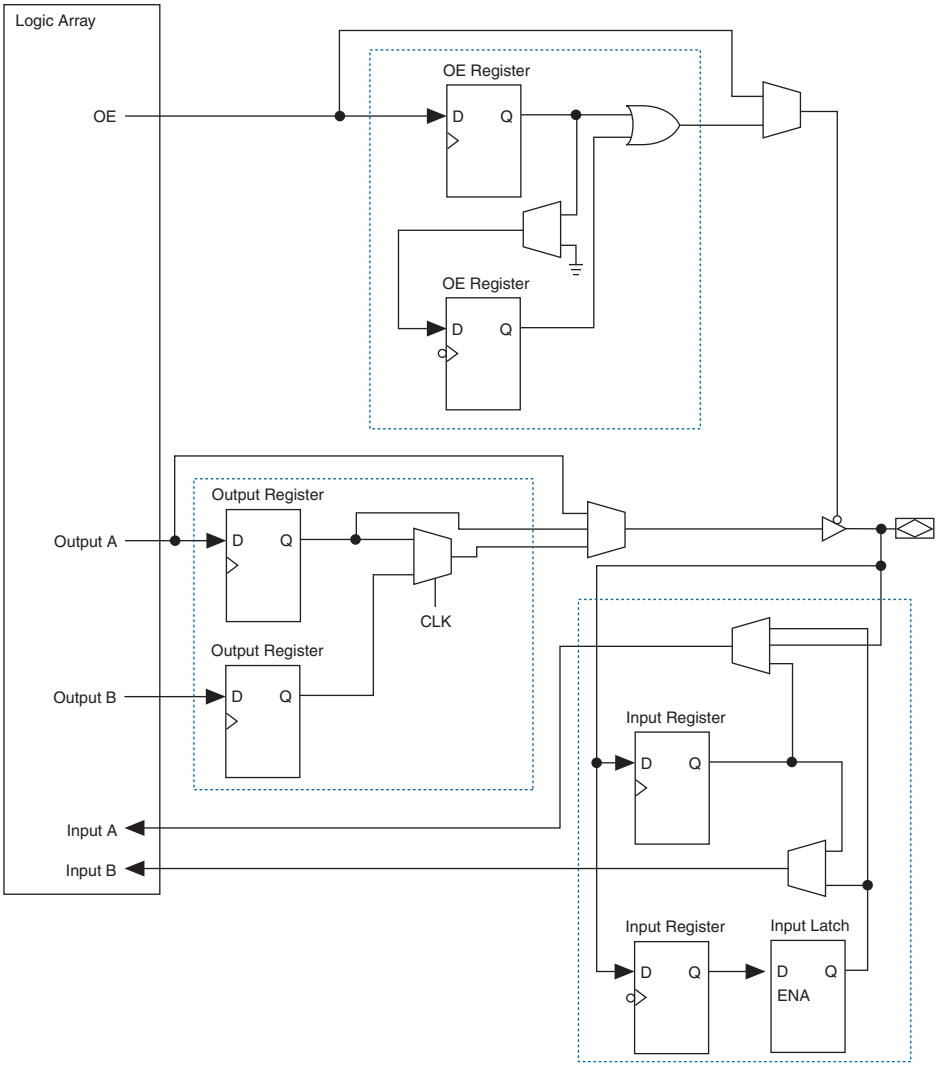
I/O Structure

IOEs provide many features, including:

- Dedicated differential and single-ended I/O buffers
- 3.3-V, 64-bit, 66-MHz PCI compliance
- 3.3-V, 64-bit, 133-MHz PCI-X 1.0 compliance
- Joint Test Action Group (JTAG) boundary-scan test (BST) support
- Differential on-chip termination for LVDS I/O standard
- Programmable pull-up during configuration
- Output drive strength control
- Slew-rate control
- Tri-state buffers
- Bus-hold circuitry
- Programmable pull-up resistors
- Programmable input and output delays
- Open-drain outputs
- DQ and DQS I/O pins
- Double-data rate (DDR) Registers

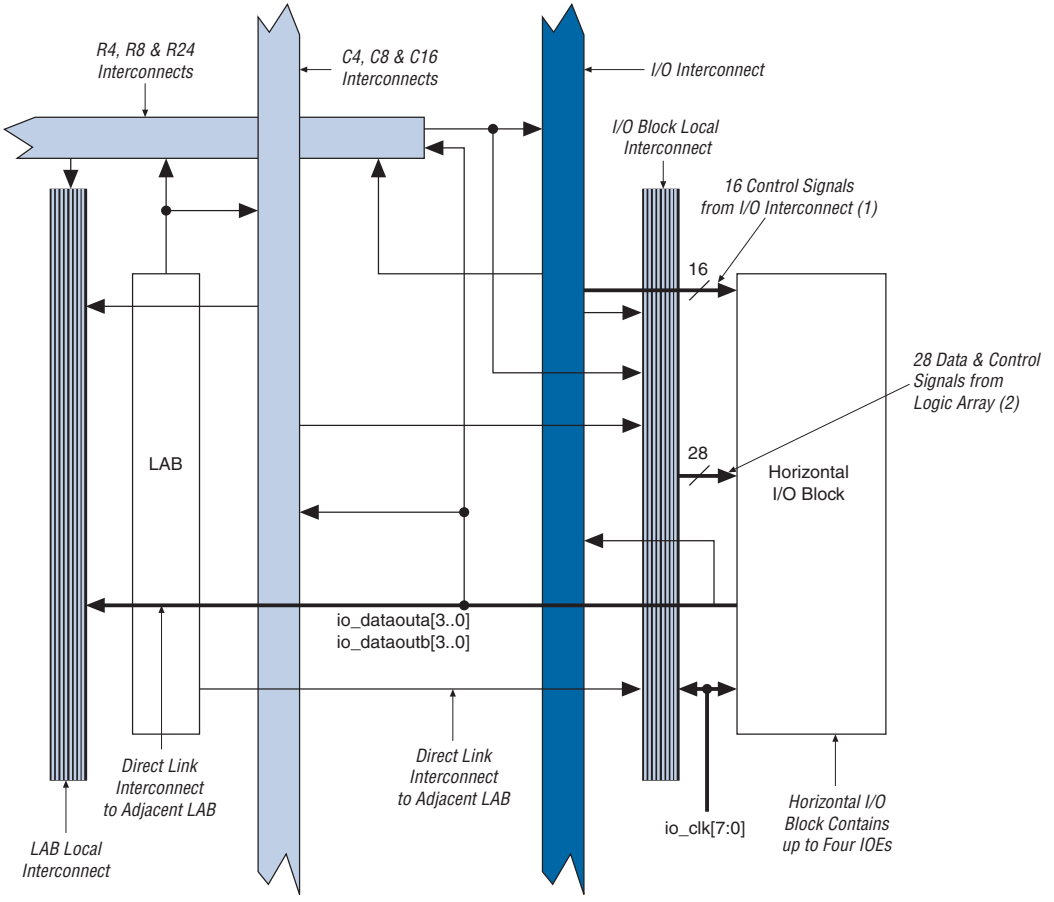
The IOE in Stratix devices contains a bidirectional I/O buffer, six registers, and a latch for a complete embedded bidirectional single data rate or DDR transfer. Figure 2–59 shows the Stratix IOE structure. The IOE contains two input registers (plus a latch), two output registers, and two output enable registers. The design can use both input registers and the latch to capture DDR input and both output registers to drive DDR outputs. Additionally, the design can use the output enable (OE) register for fast clock-to-output enable timing. The negative edge-clocked OE register is used for DDR SDRAM interfacing. The Quartus II software automatically duplicates a single OE register that controls multiple output or bidirectional pins.

Figure 2–59. Stratix IOE Structure



The IOEs are located in I/O blocks around the periphery of the Stratix device. There are up to four IOEs per row I/O block and six IOEs per column I/O block. The row I/O blocks drive row, column, or direct link interconnects. The column I/O blocks drive column interconnects. Figure 2–60 shows how a row I/O block connects to the logic array. Figure 2–61 shows how a column I/O block connects to the logic array.

Figure 2–60. Row I/O Block Connection to the Interconnect

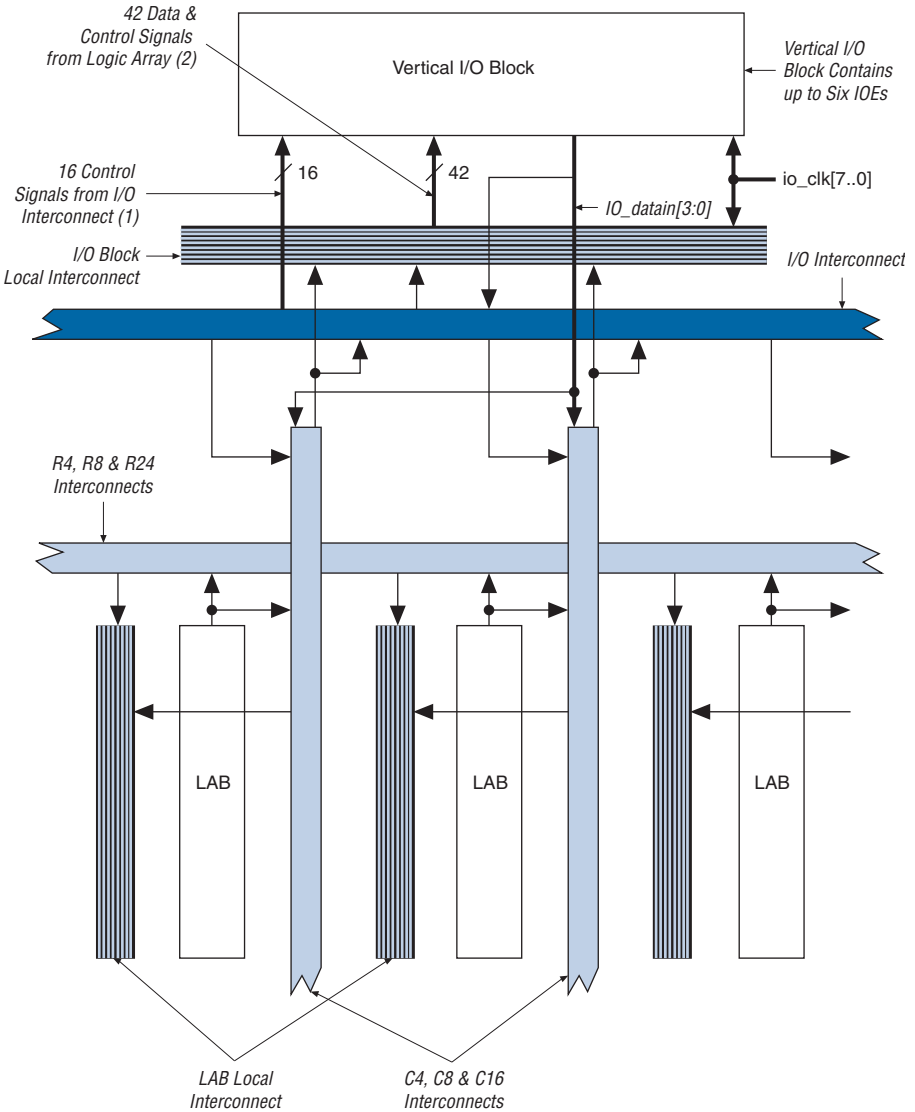


Notes to Figure 2–60:

(1) The 16 control signals are composed of four output enables `io_boe[3..0]`, four clock enables `io_bce[3..0]`, four clocks `io_clk[3..0]`, and four clear signals `io_bclr[3..0]`.

(2) The 28 data and control signals consist of eight data out lines: four lines each for DDR applications `io_dataouta[3..0]` and `io_dataoutb[3..0]`, four output enables `io_coe[3..0]`, four input clock enables `io_cce_in[3..0]`, four output clock enables `io_cce_out[3..0]`, four clocks `io_cclk[3..0]`, and four clear signals `io_cclr[3..0]`.

Figure 2–61. Column I/O Block Connection to the Interconnect



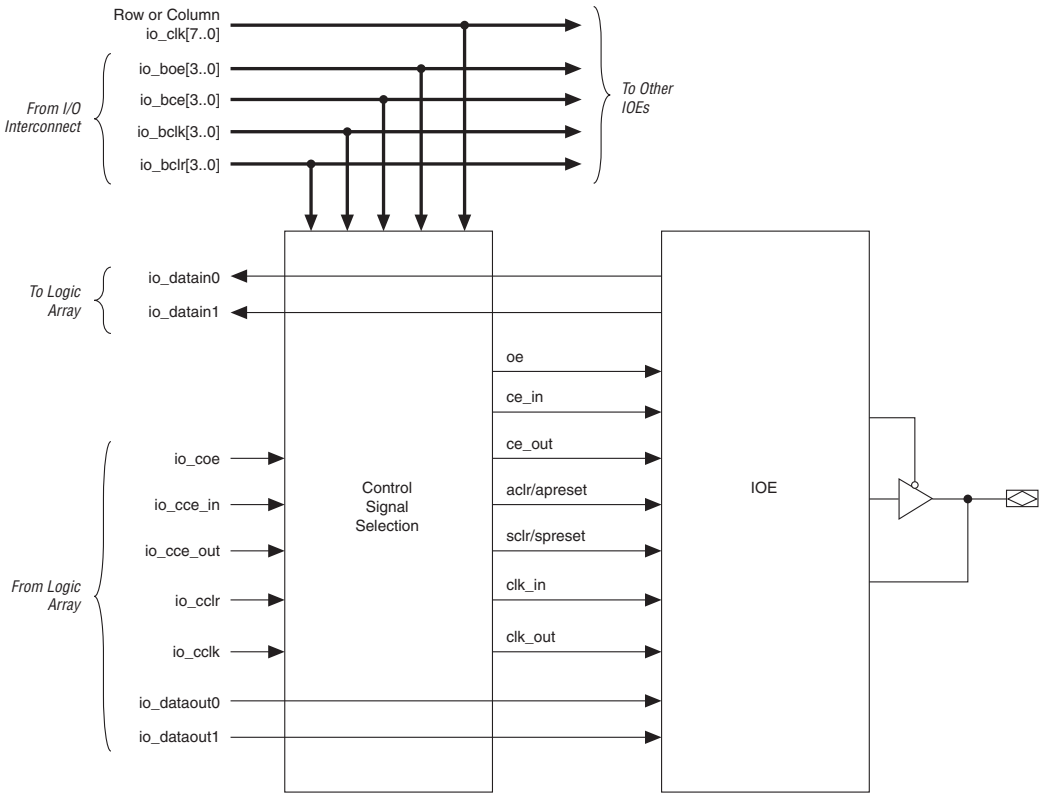
Notes to Figure 2–61:

(1) The 16 control signals are composed of four output enables `io_boe[3..0]`, four clock enables `io_bce[3..0]`, four clocks `io_bclk[3..0]`, and four clear signals `io_bclr[3..0]`.

(2) The 42 data and control signals consist of 12 data out lines; six lines each for DDR applications `io_dataouta[5..0]` and `io_dataoutb[5..0]`, six output enables `io_coe[5..0]`, six input clock enables `io_cce_in[5..0]`, six output clock enables `io_cce_out[5..0]`, six clocks `io_cclk[5..0]`, and six clear signals `io_cclr[5..0]`.

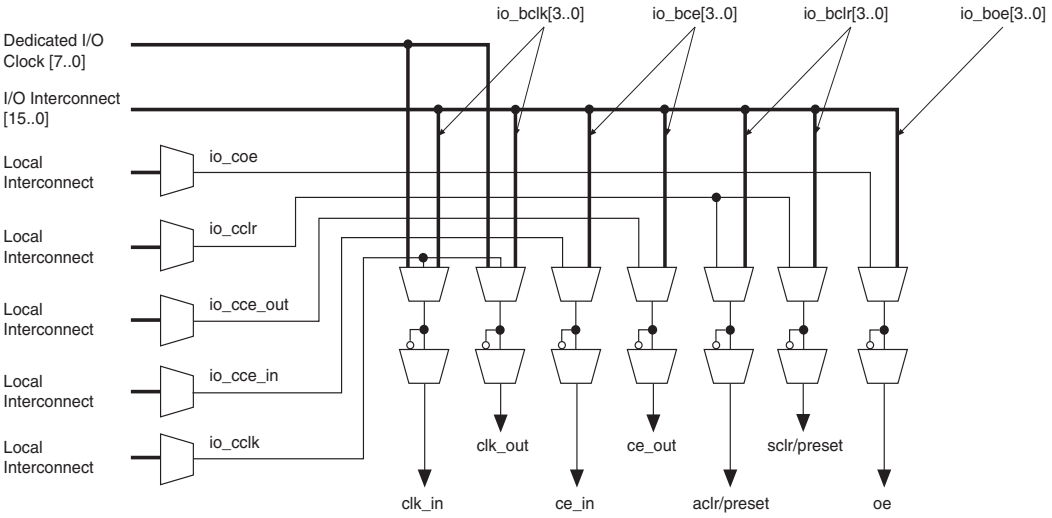
Stratix devices have an I/O interconnect similar to the R4 and C4 interconnect to drive high-fanout signals to and from the I/O blocks. There are 16 signals that drive into the I/O blocks composed of four output enables `io_boe[3..0]`, four clock enables `io_bce[3..0]`, four clocks `io_bclk[3..0]`, and four clear signals `io_bclr[3..0]`. The pin's `datain` signals can drive the IO interconnect, which in turn drives the logic array or other I/O blocks. In addition, the control and data signals can be driven from the logic array, providing a slower but more flexible routing resource. The row or column IOE clocks, `io_clk[7..0]`, provide a dedicated routing resource for low-skew, high-speed clocks. I/O clocks are generated from regional, global, or fast regional clocks (see "PLLs & Clock Networks" on page 2–73). Figure 2–62 illustrates the signal paths through the I/O block.

Figure 2–62. Signal Path through the I/O Block



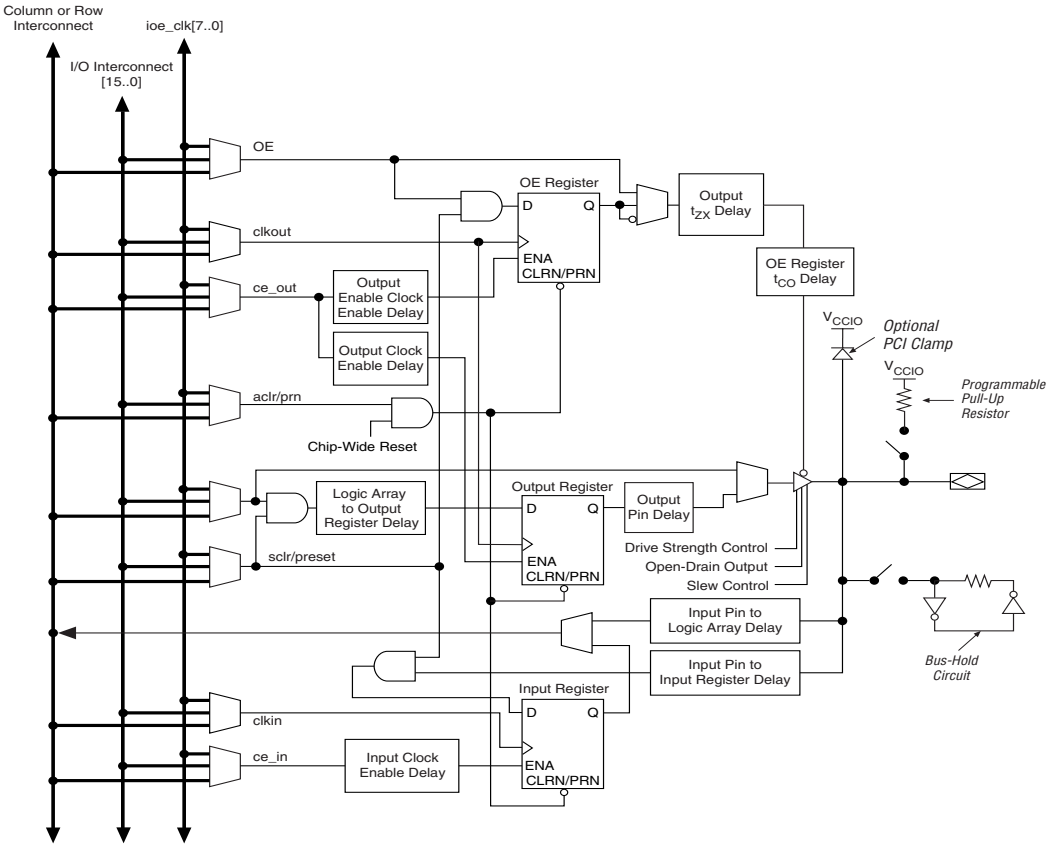
Each IOE contains its own control signal selection for the following control signals: oe, ce_in, ce_out, aclr/preset, sclr/preset, clk_in, and clk_out. Figure 2–63 illustrates the control signal selection.

Figure 2–63. Control Signal Selection per IOE



In normal bidirectional operation, the input register can be used for input data requiring fast setup times. The input register can have its own clock input and clock enable separate from the OE and output registers. The output register can be used for data requiring fast clock-to-output performance. The OE register can be used for fast clock-to-output enable timing. The OE and output register share the same clock source and the same clock enable source from local interconnect in the associated LAB, dedicated I/O clocks, and the column and row interconnects. Figure 2–64 shows the IOE in bidirectional configuration.

Figure 2–64. Stratix IOE in Bidirectional I/O Configuration Note (1)



Note to Figure 2–64:
(1) All input signals to the IOE can be inverted at the IOE.

The Stratix device IOE includes programmable delays that can be activated to ensure zero hold times, input IOE register-to-logic array register transfers, or logic array-to-output IOE register transfers.

A path in which a pin directly drives a register may require the delay to ensure zero hold time, whereas a path in which a pin drives a register through combinatorial logic may not require the delay. Programmable delays exist for decreasing input-pin-to-logic-array and IOE input register delays. The Quartus II Compiler can program these delays to automatically minimize setup time while providing a zero hold time. Programmable delays can increase the register-to-pin delays for output

and/or output enable registers. A programmable delay exists to increase the t_{ZX} delay to the output pin, which is required for ZBT interfaces. Table 2–24 shows the programmable delays for Stratix devices.

Table 2–24. Stratix Programmable Delay Chain

Programmable Delays	Quartus II Logic Option
Input pin to logic array delay	Decrease input delay to internal cells
Input pin to input register delay	Decrease input delay to input register
Output pin delay	Increase delay to output pin
Output enable register t_{CO} delay	Increase delay to output enable pin
Output t_{ZX} delay	Increase t_{ZX} delay to output pin
Output clock enable delay	Increase output clock enable delay
Input clock enable delay	Increase input clock enable delay
Logic array to output register delay	Decrease input delay to output register
Output enable clock enable delay	Increase output enable clock enable delay

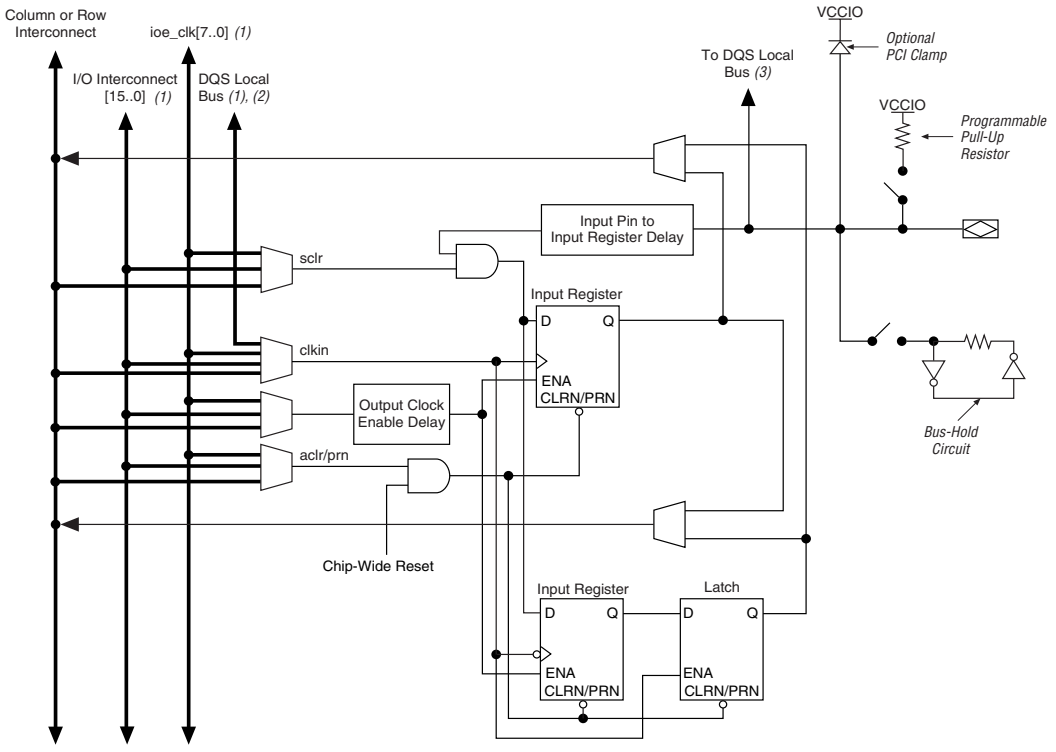
The IOE registers in Stratix devices share the same source for clear or preset. You can program preset or clear for each individual IOE. You can also program the registers to power up high or low after configuration is complete. If programmed to power up low, an asynchronous clear can control the registers. If programmed to power up high, an asynchronous preset can control the registers. This feature prevents the inadvertent activation of another device's active-low input upon power-up. If one register in an IOE uses a preset or clear signal then all registers in the IOE must use that same signal if they require preset or clear. Additionally a synchronous reset signal is available for the IOE registers.

Double-Data Rate I/O Pins

Stratix devices have six registers in the IOE, which support DDR interfacing by clocking data on both positive and negative clock edges. The IOEs in Stratix devices support DDR inputs, DDR outputs, and bidirectional DDR modes.

When using the IOE for DDR inputs, the two input registers clock double rate input data on alternating edges. An input latch is also used within the IOE for DDR input acquisition. The latch holds the data that is present during the clock high times. This allows both bits of data to be synchronous with the same clock edge (either rising or falling). Figure 2–65 shows an IOE configured for DDR input. Figure 2–66 shows the DDR input timing diagram.

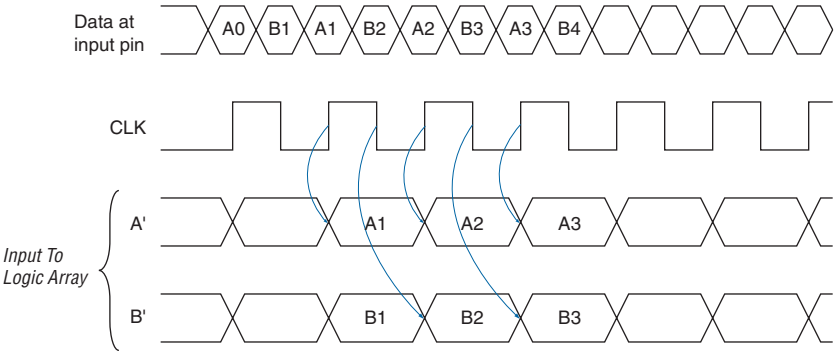
Figure 2–65. Stratix IOE in DDR Input I/O Configuration Note (1)



Notes to Figure 2–65:

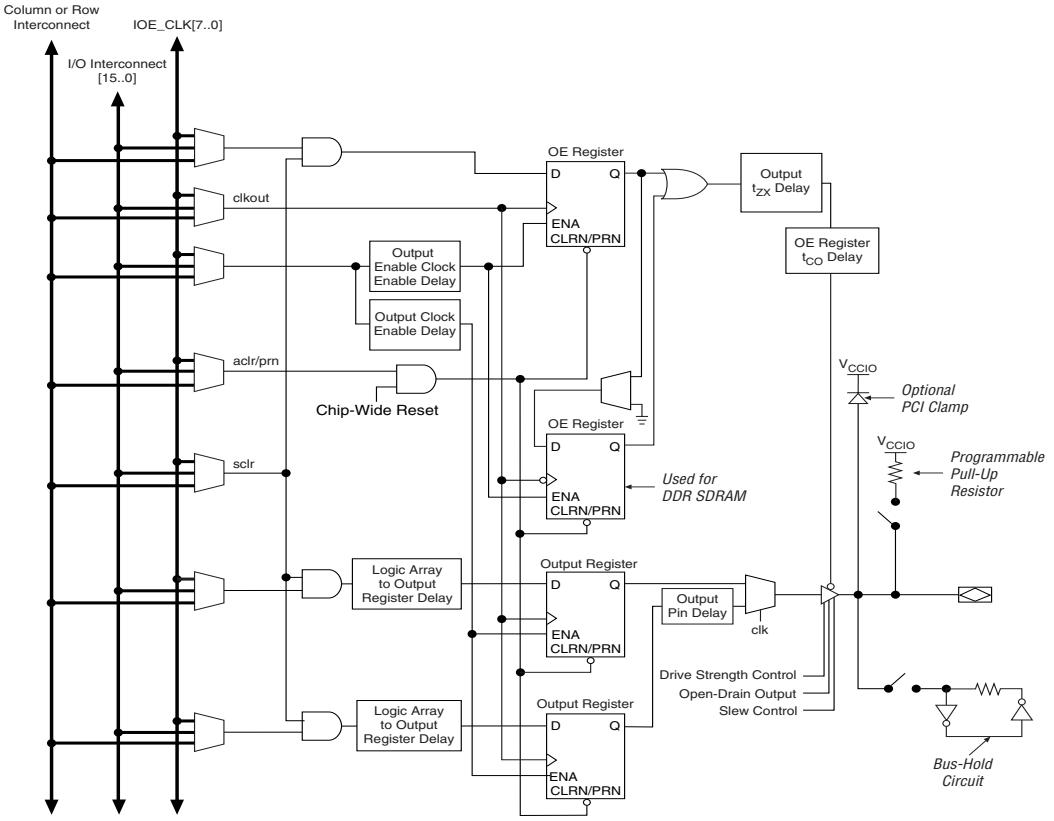
(1) All input signals to the IOE can be inverted at the IOE.
(2) This signal connection is only allowed on dedicated DQ function pins.
(3) This signal is for dedicated DQS function pins only.

Figure 2–66. Input Timing Diagram in DDR Mode



When using the IOE for DDR outputs, the two output registers are configured to clock two data paths from LEs on rising clock edges. These output registers are multiplexed by the clock to drive the output pin at a ×2 rate. One output register clocks the first bit out on the clock high time, while the other output register clocks the second bit out on the clock low time. Figure 2–67 shows the IOE configured for DDR output. Figure 2–68 shows the DDR output timing diagram.

Figure 2–67. Stratix IOE in DDR Output I/O Configuration Notes (1), (2)

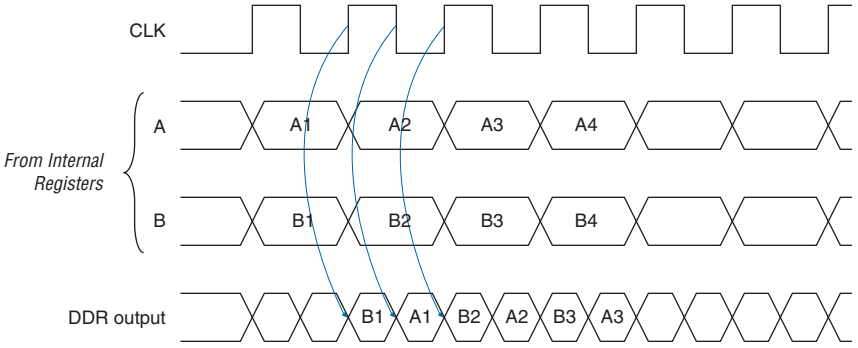


Notes to Figure 2–67:

(1) All input signals to the IOE can be inverted at the IOE.

(2) The tristate is by default active high. It can, however, be designed to be active low.

Figure 2–68. Output Timing Diagram in DDR Mode



The Stratix IOE operates in bidirectional DDR mode by combining the DDR input and DDR output configurations. Stratix device I/O pins transfer data on a DDR bidirectional bus to support DDR SDRAM. The negative-edge-clocked OE register holds the OE signal inactive until the falling edge of the clock. This is done to meet DDR SDRAM timing requirements.

External RAM Interfacing

Stratix devices support DDR SDRAM at up to 200 MHz (400-Mbps data rate) through dedicated phase-shift circuitry, QDR and QDRII SRAM interfaces up to 167 MHz, and ZBT SRAM interfaces up to 200 MHz. Stratix devices also provide preliminary support for reduced latency DRAM II (RLDRAM II) at rates up to 200 MHz through the dedicated phase-shift circuitry.

In addition to the required signals for external memory interfacing, Stratix devices offer the optional clock enable signal. By default the Quartus II software sets the clock enable signal high, which tells the output register to update with new values. The output registers hold their own values if the design sets the clock enable signal low. See Figure 2–64.

To find out more about the DDR SDRAM specification, see the JEDEC web site (**www.jedec.org**). For information on memory controller megafunctions for Stratix devices, see the Altera web site (**www.altera.com**). See *AN 342: Interfacing DDR SDRAM with Stratix & Stratix GX Devices* for more information on DDR SDRAM interface in Stratix. Also see *AN 349: QDR SRAM Controller Reference Design for Stratix & Stratix GX Devices* and *AN 329: ZBT SRAM Controller Reference Design for Stratix & Stratix GX Devices*.

Tables 2–25 and 2–26 show the performance specification for DDR SDRAM, RLDRAM II, QDR SRAM, QDRII SRAM, and ZBT SRAM interfaces in EP1S10 through EP1S40 devices and in EP1S60 and EP1S80 devices. The DDR SDRAM and QDR SRAM numbers in Table 2–25 have been verified with hardware characterization with third-party DDR SDRAM and QDR SRAM devices over temperature and voltage extremes.

Table 2–25. External RAM Support in EP1S10 through EP1S40 Devices

DDR Memory Type	I/O Standard	Maximum Clock Rate (MHz)						
		-5 Speed Grade	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade	
		Flip-Chip	Flip-Chip	Wire-Bond	Flip-Chip	Wire-Bond	Flip-Chip	Wire-Bond
DDR SDRAM *(1)*, *(2)*	SSTL-2	200	167	133	133	100	100	100
DDR SDRAM - side banks *(2)*, *(3)*, *(4)*	SSTL-2	150	133	110	133	100	100	100
RLDRAM II *(4)*	1.8-V HSTL	200	*(5)*	*(5)*	*(5)*	*(5)*	*(5)*	*(5)*
QDR SRAM *(6)*	1.5-V HSTL	167	167	133	133	100	100	100
QDRII SRAM *(6)*	1.5-V HSTL	200	167	133	133	100	100	100
ZBT SRAM *(7)*	LVTTL	200	200	200	167	167	133	133

Notes to Table 2–25:

(1) These maximum clock rates apply if the Stratix device uses DQS phase-shift circuitry to interface with DDR SDRAM. DQS phase-shift circuitry is only available in the top and bottom I/O banks (I/O banks 3, 4, 7, and 8).

(2) For more information on DDR SDRAM, see *AN 342: Interfacing DDR SDRAM with Stratix & Stratix GX Devices.*

(3) DDR SDRAM is supported on the Stratix device side I/O banks (I/O banks 1, 2, 5, and 6) without dedicated DQS phase-shift circuitry. The read DQS signal is ignored in this mode.

(4) These performance specifications are preliminary.

(5) This device does not support RLDRAM II.

(6) For more information on QDR or QDRII SRAM, see *AN 349: QDR SRAM Controller Reference Design for Stratix & Stratix GX Devices*.

(7) For more information on ZBT SRAM, see *AN 329: ZBT SRAM Controller Reference Design for Stratix & Stratix GX Devices*.

Table 2–26. External RAM Support in EP1S60 & EP1S80 Devices

DDR Memory Type	I/O Standard	Maximum Clock Rate (MHz)		
		-5 Speed Grade	-6 Speed Grade	-7 Speed Grade
DDR SDRAM *(1)*, *(2)*	SSTL-2	167	167	133
DDR SDRAM - side banks *(2)*, *(3)*	SSTL-2	150	133	133
QDR SRAM *(4)*	1.5-V HSTL	133	133	133
QDRII SRAM *(4)*	1.5-V HSTL	167	167	133
ZBT SRAM *(5)*	LVTTL	200	200	167

Notes to Table 2–26:

(1) These maximum clock rates apply if the Stratix device uses DQS phase-shift circuitry to interface with DDR SDRAM. DQS phase-shift circuitry is only available in the top and bottom I/O banks (I/O banks 3, 4, 7, and 8).

(2) For more information on DDR SDRAM, see *AN 342: Interfacing DDR SDRAM with Stratix & Stratix GX Devices*.

(3) DDR SDRAM is supported on the Stratix device side I/O banks (I/O banks 1, 2, 5, and 6) without dedicated DQS phase-shift circuitry. The read DQS signal is ignored in this mode. Numbers are preliminary.

(4) For more information on QDR or QDRII SRAM, see *AN 349: QDR SRAM Controller Reference Design for Stratix & Stratix GX Devices*.

(5) For more information on ZBT SRAM, see *AN 329: ZBT SRAM Controller Reference Design for Stratix & Stratix GX Devices*.

In addition to six I/O registers and one input latch in the IOE for interfacing to these high-speed memory interfaces, Stratix devices also have dedicated circuitry for interfacing with DDR SDRAM. In every Stratix device, the I/O banks at the top (I/O banks 3 and 4) and bottom (I/O banks 7 and 8) of the device support DDR SDRAM up to 200 MHz. These pins support DQS signals with DQ bus modes of ×8, ×16, or ×32.

Table 2–27 shows the number of DQ and DQS buses that are supported per device.

Table 2–27. DQS & DQ Bus Mode Support (Part 1 of 2) Note (1)

Device	Package	Number of ×8 Groups	Number of ×16 Groups	Number of ×32 Groups
EP1S10	672-pin BGA 672-pin FineLine BGA	12 *(2)*	0	0
	484-pin FineLine BGA 780-pin FineLine BGA	16 *(3)*	0	4
EP1S20	484-pin FineLine BGA	18*(4)*	7 *(5)*	4
	672-pin BGA 672-pin FineLine BGA	16*(3)*	7 *(5)*	4
	780-pin FineLine BGA	20	7 *(5)*	4

Table 2–27. DQS & DQ Bus Mode Support (Part 2 of 2) Note (1)

Device	Package	Number of ×8 Groups	Number of ×16 Groups	Number of ×32 Groups
EP1S25	672-pin BGA 672-pin FineLine BGA	16 (3)	8	4
	780-pin FineLine BGA 1,020-pin FineLine BGA	20	8	4
EP1S30	956-pin BGA 780-pin FineLine BGA 1,020-pin FineLine BGA	20	8	4
EP1S40	956-pin BGA 1,020-pin FineLine BGA 1,508-pin FineLine BGA	20	8	4
EP1S60	956-pin BGA 1,020-pin FineLine BGA 1,508-pin FineLine BGA	20	8	4
EP1S80	956-pin BGA 1,508-pin FineLine BGA 1,923-pin FineLine BGA	20	8	4

Notes to Table 2–27:

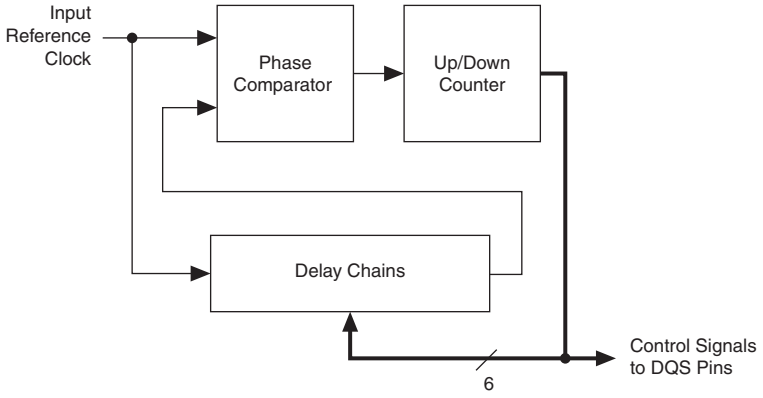
(1) See the *Selectable I/O Standards in Stratix & Stratix GX Devices* chapter in the *Stratix Device Handbook, Volume 2* for V_{REF} guidelines.
(2) These packages have six groups in I/O banks 3 and 4 and six groups in I/O banks 7 and 8.
(3) These packages have eight groups in I/O banks 3 and 4 and eight groups in I/O banks 7 and 8.
(4) This package has nine groups in I/O banks 3 and 4 and nine groups in I/O banks 7 and 8.
(5) These packages have three groups in I/O banks 3 and 4 and four groups in I/O banks 7 and 8.

A compensated delay element on each DQS pin automatically aligns input DQS synchronization signals with the data window of their corresponding DQ data signals. The DQS signals drive a local DQS bus in the top and bottom I/O banks. This DQS bus is an additional resource to the I/O clocks and is used to clock DQ input registers with the DQS signal.

Two separate single phase-shifting reference circuits are located on the top and bottom of the Stratix device. Each circuit is driven by a system reference clock through the `CLK` pins that is the same frequency as the DQS signal. Clock pins `CLK[15..12]p` feed the phase-shift circuitry on the top of the device and clock pins `CLK[7..4]p` feed the phase-shift circuitry on the bottom of the device. The phase-shifting reference circuit on the top of the device controls the compensated delay elements for all 10 DQS pins located at the top of the device. The phase-shifting reference circuit on the bottom of the device controls the compensated delay elements for all 10 DQS pins located on the bottom of the device. All 10 delay elements (DQS signals) on either the top or bottom of the device

shift by the same degree amount. For example, all 10 DQS pins on the top of the device can be shifted by 90° and all 10 DQS pins on the bottom of the device can be shifted by 72°. The reference circuits require a maximum of 256 system reference clock cycles to set the correct phase on the DQS delay elements. Figure 2–69 illustrates the phase-shift reference circuit control of each DQS delay shift on the top of the device. This same circuit is duplicated on the bottom of the device.

Figure 2–69. Simplified Diagram of the DQS Phase-Shift Circuitry



See the *External Memory Interfaces* chapter in the *Stratix Device Handbook, Volume 2* for more information on external memory interfaces.

Programmable Drive Strength

The output buffer for each Stratix device I/O pin has a programmable drive strength control for certain I/O standards. The LVTTL and LVCMOS standard has several levels of drive strength that the user can control. SSTL-3 Class I and II, SSTL-2 Class I and II, HSTL Class I and II, and 3.3-V GTL+ support a minimum setting, the lowest drive strength that guarantees the I_{OH}/I_{OL} of the standard. Using minimum settings provides signal slew rate control to reduce system noise and signal overshoot.

Table 2–28 shows the possible settings for the I/O standards with drive strength control.

Table 2–28. Programmable Drive Strength

I/O Standard	I_{OH} / I_{OL} Current Strength Setting (mA)
3.3-V LVTTL	24 *(1)*, 16, 12, 8, 4
3.3-V LVCMOS	24 *(2)*, 12 *(1)*, 8, 4, 2
2.5-V LVTTL/LVCMOS	16 *(1)*, 12, 8, 2
1.8-V LVTTL/LVCMOS	12 *(1)*, 8, 2
1.5-V LVCMOS	8 *(1)*, 4, 2
GTL/GTL+ 1.5-V HSTL Class I and II 1.8-V HSTL Class I and II SSTL-3 Class I and II SSTL-2 Class I and II SSTL-18 Class I and II	Support max and min strength

Notes to Table 2–28:
(1) This is the Quartus II software default current setting.
(2) I/O banks 1, 2, 5, and 6 do not support this setting.

Quartus II software version 4.2 and later will report current strength as "PCI Compliant" for 3.3-V PCI, 3.3-V PCI-X 1.0, and Compact PCI I/O standards.

Stratix devices support series on-chip termination (OCT) using programmable drive strength. For more information, contact your Altera Support Representative.

Open-Drain Output

Stratix devices provide an optional open-drain (equivalent to an open-collector) output for each I/O pin. This open-drain output enables the device to provide system-level control signals (e.g., interrupt and write-enable signals) that can be asserted by any of several devices.

Slew-Rate Control

The output buffer for each Stratix device I/O pin has a programmable output slew-rate control that can be configured for low-noise or high-speed performance. A faster slew rate provides high-speed transitions for high-performance systems. However, these fast transitions may introduce noise transients into the system. A slow slew rate reduces system noise, but adds a nominal delay to rising and falling edges. Each

I/O pin has an individual slew-rate control, allowing you to specify the slew rate on a pin-by-pin basis. The slew-rate control affects both the rising and falling edges.

Bus Hold

Each Stratix device I/O pin provides an optional bus-hold feature. The bus-hold circuitry can weakly hold the signal on an I/O pin at its last-driven state. Since the bus-hold feature holds the last-driven state of the pin until the next input signal is present, an external pull-up or pull-down resistor is not needed to hold a signal level when the bus is tri-stated.

Table 2–29 shows bus hold support for different pin types.

Table 2–29. Bus Hold Support

Pin Type	Bus Hold
I/O pins	✓
CLK[15..0]	
CLK[0,1,2,3,8,9,10,11]	
FCLK	✓
FPLL[7..10]CLK	

The bus-hold circuitry also pulls undriven pins away from the input threshold voltage where noise can cause unintended high-frequency switching. You can select this feature individually for each I/O pin. The bus-hold output drives no higher than V_{CCIO} to prevent overdriving signals. If the bus-hold feature is enabled, the programmable pull-up option cannot be used. Disable the bus-hold feature when using open-drain outputs with the GTL+ I/O standard or when the I/O pin has been configured for differential signals.

The bus-hold circuitry uses a resistor with a nominal resistance (R_{BH}) of approximately 7 kΩ to weakly pull the signal level to the last-driven state. See the *DC & Switching Characteristics* chapter of the *Stratix Device Handbook, Volume 1* for the specific sustaining current driven through this resistor and overdrive current used to identify the next-driven input level. This information is provided for each V_{CCIO} voltage level.

The bus-hold circuitry is active only after configuration. When going into user mode, the bus-hold circuit captures the value on the pin present at the end of configuration.

Programmable Pull-Up Resistor

Each Stratix device I/O pin provides an optional programmable pull-up resistor during user mode. If this feature is enabled for an I/O pin, the pull-up resistor (typically 25 kΩ) weakly holds the output to the V_{CCIO} level of the output pin's bank. Table 2–30 shows which pin types support the weak pull-up resistor feature.

Table 2–30. Programmable Weak Pull-Up Resistor Support

Pin Type	Programmable Weak Pull-Up Resistor
I/O pins	✓
CLK[15..0]	
FCLK	✓
FPLL[7..10]CLK	
Configuration pins	
JTAG pins	✓ *(1)*

Note to Table 2–30:
(1) TDO pins do not support programmable weak pull-up resistors.

Advanced I/O Standard Support

Stratix device IOEs support the following I/O standards:

- LVTTL
- LVCMOS
- 1.5 V
- 1.8 V
- 2.5 V
- 3.3-V PCI
- 3.3-V PCI-X 1.0
- 3.3-V AGP (1× and 2×)
- LVDS
- LVPECL
- 3.3-V PCML
- HyperTransport
- Differential HSTL (on input/output clocks only)
- Differential SSTL (on output column clock pins only)
- GTL/GTL+
- 1.5-V HSTL Class I and II

- 1.8-V HSTL Class I and II
- SSTL-3 Class I and II
- SSTL-2 Class I and II
- SSTL-18 Class I and II
- CTT

Table 2–31 describes the I/O standards supported by Stratix devices.

Table 2–31. Stratix Supported I/O Standards

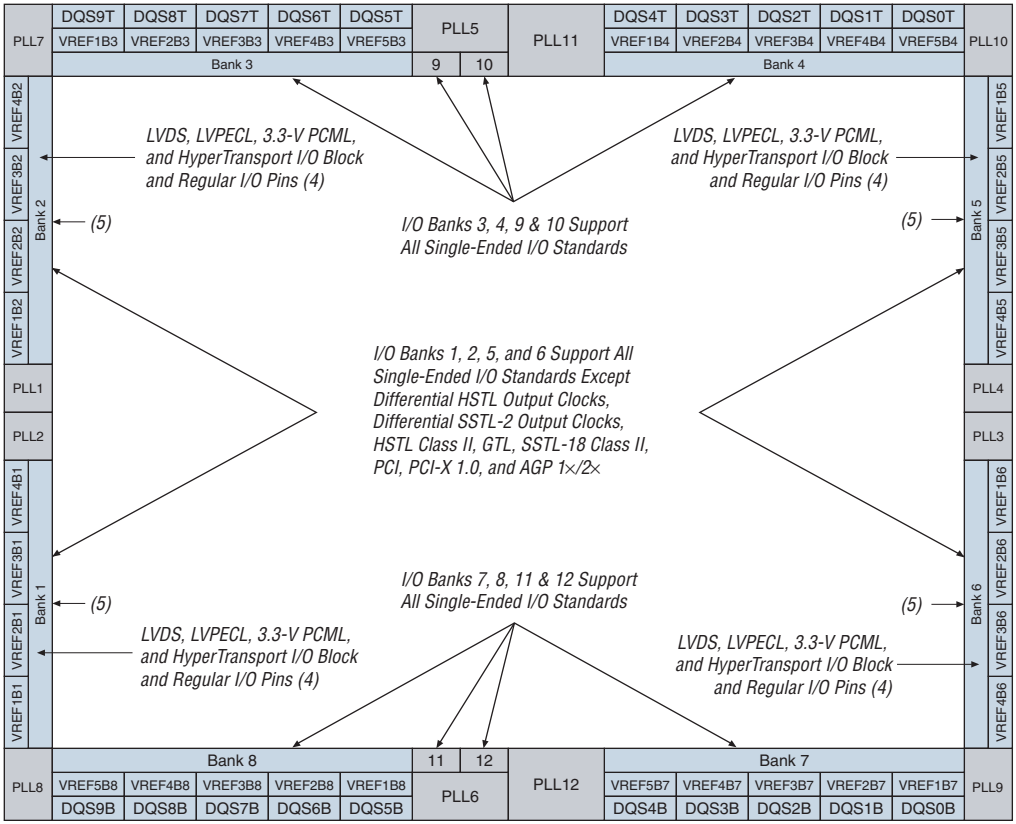
I/O Standard	Type	Input Reference Voltage (V_{REF}) (V)	Output Supply Voltage (V_{CCIO}) (V)	Board Termination Voltage (V_{TT}) (V)
LVTTL	Single-ended	N/A	3.3	N/A
LVCMOS	Single-ended	N/A	3.3	N/A
2.5 V	Single-ended	N/A	2.5	N/A
1.8 V	Single-ended	N/A	1.8	N/A
1.5 V	Single-ended	N/A	1.5	N/A
3.3-V PCI	Single-ended	N/A	3.3	N/A
3.3-V PCI-X 1.0	Single-ended	N/A	3.3	N/A
LVDS	Differential	N/A	3.3	N/A
LVPECL	Differential	N/A	3.3	N/A
3.3-V PCML	Differential	N/A	3.3	N/A
HyperTransport	Differential	N/A	2.5	N/A
Differential HSTL *(1)*	Differential	0.75	1.5	0.75
Differential SSTL *(2)*	Differential	1.25	2.5	1.25
GTL	Voltage-referenced	0.8	N/A	1.20
GTL+	Voltage-referenced	1.0	N/A	1.5
1.5-V HSTL Class I and II	Voltage-referenced	0.75	1.5	0.75
1.8-V HSTL Class I and II	Voltage-referenced	0.9	1.8	0.9
SSTL-18 Class I and II	Voltage-referenced	0.90	1.8	0.90
SSTL-2 Class I and II	Voltage-referenced	1.25	2.5	1.25
SSTL-3 Class I and II	Voltage-referenced	1.5	3.3	1.5
AGP (1× and 2°)	Voltage-referenced	1.32	3.3	N/A
CTT	Voltage-referenced	1.5	3.3	1.5

Notes to Table 2–31:
(1) This I/O standard is only available on input and output clock pins.
(2) This I/O standard is only available on output column clock pins.

For more information on I/O standards supported by Stratix devices, see the *Selectable I/O Standards in Stratix & Stratix GX Devices* chapter of the *Stratix Device Handbook, Volume 2*.

Stratix devices contain eight I/O banks in addition to the four enhanced PLL external clock out banks, as shown in Figure 2–70. The four I/O banks on the right and left of the device contain circuitry to support high-speed differential I/O for LVDS, LVPECL, 3.3-V PCML, and HyperTransport inputs and outputs. These banks support all I/O standards listed in Table 2–31 except PCI I/O pins or PCI-X 1.0, GTL, SSTL-18 Class II, and HSTL Class II outputs. The top and bottom I/O banks support all single-ended I/O standards. Additionally, Stratix devices support four enhanced PLL external clock output banks, allowing clock output capabilities such as differential support for SSTL and HSTL. Table 2–32 shows I/O standard support for each I/O bank.

Figure 2–70. Stratix I/O Banks *Notes (1), (2), (3)*



Notes to Figure 2–70:

(1) Figure 2–70 is a top view of the silicon die. This will correspond to a top-down view for non-flip-chip packages, but will be a reverse view for flip-chip packages.

(2) Figure 2–70 is a graphic representation only. See the device pin-outs on the web (**www.altera.com**) and the Quartus II software for exact locations.

(3) Banks 9 through 12 are enhanced PLL external clock output banks.

(4) If the high-speed differential I/O pins are not used for high-speed differential signaling, they can support all of the I/O standards except HSTL Class I and II, GTL, SSTL-18 Class II, PCI, PCI-X 1.0, and AGP 1× /2×.

(5) For guidelines for placing single-ended I/O pads next to differential I/O pads, see the *Selectable I/O Standards in Stratix and Stratix GX Devices* chapter in the *Stratix Device Handbook, Volume 2*.

Table 2–32 shows I/O standard support for each I/O bank.

Table 2–32. I/O Support by Bank (Part 1 of 2)

I/O Standard	Top & Bottom Banks (3, 4, 7 & 8)	Left & Right Banks (1, 2, 5 & 6)	Enhanced PLL External Clock Output Banks (9, 10, 11 & 12)
LVTTL	✓	✓	✓
LVCMOS	✓	✓	✓
2.5 V	✓	✓	✓
1.8 V	✓	✓	✓
1.5 V	✓	✓	✓
3.3-V PCI	✓		✓
3.3-V PCI-X 1.0	✓		✓
LVPECL		✓	✓
3.3-V PCML		✓	✓
LVDS		✓	✓
HyperTransport technology		✓	✓
Differential HSTL (clock inputs)	✓	✓	
Differential HSTL (clock outputs)			✓
Differential SSTL (clock outputs)			✓
3.3-V GTL	✓		✓
3.3-V GTL+	✓	✓	✓
1.5-V HSTL Class I	✓	✓	✓
1.5-V HSTL Class II	✓		✓
1.8-V HSTL Class I	✓	✓	✓
1.8-V HSTL Class II	✓		✓
SSTL-18 Class I	✓	✓	✓
SSTL-18 Class II	✓		✓
SSTL-2 Class I	✓	✓	✓
SSTL-2 Class II	✓	✓	✓
SSTL-3 Class I	✓	✓	✓

Table 2–32. I/O Support by Bank (Part 2 of 2)

I/O Standard	Top & Bottom Banks (3, 4, 7 & 8)	Left & Right Banks (1, 2, 5 & 6)	Enhanced PLL External Clock Output Banks (9, 10, 11 & 12)
SSTL-3 Class II	✓	✓	✓
AGP (1× and 2×)	✓		✓
CTT	✓	✓	✓

Each I/O bank has its own VCCIO pins. A single device can support 1.5-, 1.8-, 2.5-, and 3.3-V interfaces; each bank can support a different standard independently. Each bank also has dedicated VREF pins to support any one of the voltage-referenced standards (such as SSTL-3) independently.

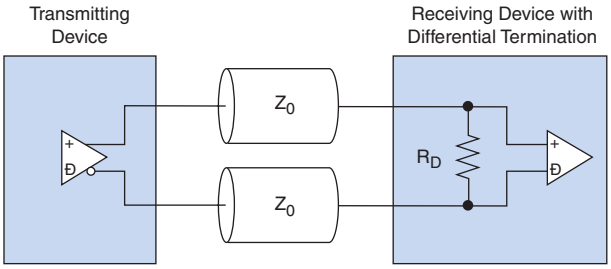
Each I/O bank can support multiple standards with the same V_{CCIO} for input and output pins. Each bank can support one voltage-referenced I/O standard. For example, when V_{CCIO} is 3.3 V, a bank can support LVTTL, LVCMOS, 3.3-V PCI, and SSTL-3 for inputs and outputs.

Differential On-Chip Termination

Stratix devices provide differential on-chip termination (LVDS I/O standard) to reduce reflections and maintain signal integrity. Differential on-chip termination simplifies board design by minimizing the number of external termination resistors required. Termination can be placed inside the package, eliminating small stubs that can still lead to reflections. The internal termination is designed using transistors in the linear region of operation.

Stratix devices support internal differential termination with a nominal resistance value of 137.5 Ω for LVDS input receiver buffers. LVPECL signals require an external termination resistor. Figure 2–71 shows the device with differential termination.

Figure 2–71. LVDS Input Differential On-Chip Termination



I/O banks on the left and right side of the device support LVDS receiver (far-end) differential termination.

Table 2–33 shows the Stratix device differential termination support.

Table 2–33. Differential Termination Supported by I/O Banks

Differential Termination Support	I/O Standard Support	Top & Bottom Banks (3, 4, 7 & 8)	Left & Right Banks (1, 2, 5 & 6)
Differential termination *(1)*, *(2)*	LVDS		✓

Notes to Table 2–33:
(1) Clock pin CLK0, CLK2, CLK9, CLK11, and pins FPLL[7..10]CLK do not support differential termination.
(2) Differential termination is only supported for LVDS because of a 3.3-V V_{CCIO}.

Table 2–34 shows the termination support for different pin types.

Table 2–34. Differential Termination Support Across Pin Types

Pin Type	R_D
Top and bottom I/O banks (3, 4, 7, and 8)	
DIFFIO_RX[]	✓
CLK[0,2,9,11],CLK[4-7],CLK[12-15]	
CLK[1,3,8,10]	✓
FCLK	
FPLL[7..10]CLK	

The differential on-chip resistance at the receiver input buffer is 118 Ω ±20 %.

However, there is additional resistance present between the device ball and the input of the receiver buffer, as shown in Figure 2–72. This resistance is because of package trace resistance (which can be calculated as the resistance from the package ball to the pad) and the parasitic layout metal routing resistance (which is shown between the pad and the intersection of the on-chip termination and input buffer).

Figure 2–72. Differential Resistance of LVDS Differential Pin Pair (R_D)

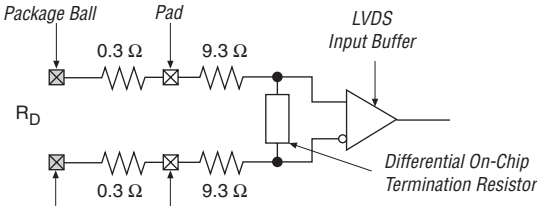


Table 2–35 defines the specification for internal termination resistance for commercial devices.

Table 2–35. Differential On-Chip Termination

Symbol	Description	Conditions	Resistance			Unit
			Min	Typ	Max	
R_D *(2)*	Internal differential termination for LVDS	Commercial *(1)*, *(3)*	110	135	165	W
		Industrial *(2)*, *(3)*	100	135	170	W

Notes to Table 2–35:

(1) Data measured over minimum conditions (T_j = 0 C, V_{CCIO} +5%) and maximum conditions (T_j = 85 C, V_{CCIO} = –5%).

(2) Data measured over minimum conditions (T_j = –40 C, V_{CCIO} +5%) and maximum conditions (T_j = 100 C, V_{CCIO} = –5%).

(3) LVDS data rate is supported for 840 Mbps using internal differential termination.

MultiVolt I/O Interface

The Stratix architecture supports the MultiVolt I/O interface feature, which allows Stratix devices in all packages to interface with systems of different supply voltages.

The Stratix VCCINT pins must always be connected to a 1.5-V power supply. With a 1.5-V V_{CCINT} level, input pins are 1.5-V, 1.8-V, 2.5-V, and 3.3-V tolerant. The VCCIO pins can be connected to either a 1.5-V, 1.8-V, 2.5-V, or 3.3-V power supply, depending on the output requirements.

The output levels are compatible with systems of the same voltage as the power supply (i.e., when VCCIO pins are connected to a 1.5-V power supply, the output levels are compatible with 1.5-V systems). When VCCIO pins are connected to a 3.3-V power supply, the output high is 3.3 V and is compatible with 3.3-V or 5.0-V systems.

Table 2–36 summarizes Stratix MultiVolt I/O support.

Table 2–36. Stratix MultiVolt I/O Support Note (1)

V_{CCIO} (V)	Input Signal (5)					Output Signal (6)				
	1.5 V	1.8 V	2.5 V	3.3 V	5.0 V	1.5 V	1.8 V	2.5 V	3.3 V	5.0 V
1.5	✓	✓	✓ (2)	✓ (2)		✓				
1.8	✓ (2)	✓	✓ (2)	✓ (2)		✓ (3)	✓			
2.5			✓	✓		✓ (3)	✓ (3)	✓		
3.3			✓ (2)	✓	✓ (4)	✓ (3)	✓ (3)	✓ (3)	✓	✓

Notes to Table 2–36:

(1) To drive inputs higher than V_{CCIO} but less than 4.1 V, disable the PCI clamping diode. However, to drive 5.0-V inputs to the device, enable the PCI clamping diode to prevent V_I from rising above 4.0 V.
(2) The input pin current may be slightly higher than the typical value.
(3) Although V_{CCIO} specifies the voltage necessary for the Stratix device to drive out, a receiving device powered at a different level can still interface with the Stratix device if it has inputs that tolerate the V_{CCIO} value.
(4) Stratix devices can be 5.0-V tolerant with the use of an external resistor and the internal PCI clamp diode.
(5) This is the external signal that is driving the Stratix device.
(6) This represents the system voltage that Stratix supports when a VCCIO pin is connected to a specific voltage level. For example, when VCCIO is 3.3 V and if the I/O standard is LVTTL/LVCMOS, the output high of the signal coming out from Stratix is 3.3 V and is compatible with 3.3-V or 5.0-V systems.

High-Speed Differential I/O Support

Stratix devices contain dedicated circuitry for supporting differential standards at speeds up to 840 Mbps. The following differential I/O standards are supported in the Stratix device: LVDS, LVPECL, HyperTransport, and 3.3-V PCML.

There are four dedicated high-speed PLLs in the EP1S10 to EP1S25 devices and eight dedicated high-speed PLLs in the EP1S30 to EP1S80 devices to multiply reference clocks and drive high-speed differential SERDES channels.

See the Stratix device pin-outs at **www.altera.com** for additional high speed DIFFIO pin information for Stratix devices.

Table 2–37 shows the number of channels that each fast PLL can clock in EP1S10, EP1S20, and EP1S25 devices. Tables 2–38 through Table 2–41 show this information for EP1S30, EP1S40, EP1S60, and EP1S80 devices.

Table 2–37. EP1S10, EP1S20 & EP1S25 Device Differential Channels (Part 1 of 2) Note (1)

Device	Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs			
					PLL 1	PLL 2	PLL 3	PLL 4
EP1S10	484-pin FineLine BGA	Transmitter (2)	20	840 (4)	5	5	5	5
				840 (3)	10	10	10	10
		Receiver	20	840 (4)	5	5	5	5
				840 (3)	10	10	10	10
	672-pin FineLine BGA 672-pin BGA	Transmitter (2)	36	624 (4)	9	9	9	9
				624 (3)	18	18	18	18
		Receiver	36	624 (4)	9	9	9	9
				624 (3)	18	18	18	18
	780-pin FineLine BGA	Transmitter (2)	44	840 (4)	11	11	11	11
				840 (3)	22	22	22	22
		Receiver	44	840 (4)	11	11	11	11
				840 (3)	22	22	22	22
EP1S20	484-pin FineLine BGA	Transmitter (2)	24	840 (4)	6	6	6	6
				840 (3)	12	12	12	12
		Receiver	20	840 (4)	5	5	5	5
				840 (3)	10	10	10	10
	672-pin FineLine BGA 672-pin BGA	Transmitter (2)	48	624 (4)	12	12	12	12
				624 (3)	24	24	24	24
		Receiver	50	624 (4)	13	12	12	13
				624 (3)	25	25	25	25
	780-pin FineLine BGA	Transmitter (2)	66	840 (4)	17	16	16	17
				840 (3)	33	33	33	33
		Receiver	66	840 (4)	17	16	16	17
				840 (3)	33	33	33	33

Table 2–37. EP1S10, EP1S20 & EP1S25 Device Differential Channels (Part 2 of 2) Note (1)

Device	Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs			
					PLL 1	PLL 2	PLL 3	PLL 4
EP1S25	672-pin FineLine BGA 672-pin BGA	Transmitter (2)	56	624 (4)	14	14	14	14
				624 (3)	28	28	28	28
		Receiver	58	624 (4)	14	15	15	14
				624 (3)	29	29	29	29
	780-pin FineLine BGA	Transmitter (2)	70	840 (4)	18	17	17	18
				840 (3)	35	35	35	35
		Receiver	66	840 (4)	17	16	16	17
				840 (3)	33	33	33	33
	1,020-pin FineLine BGA	Transmitter (2)	78	840 (4)	19	20	20	19
				840 (3)	39	39	39	39
		Receiver	78	840 (4)	19	20	20	19
				840 (3)	39	39	39	39

Notes to Table 2–37:

(1) The first row for each transmitter or receiver reports the number of channels driven directly by the PLL. The second row below it shows the maximum channels a PLL can drive if cross bank channels are used from the adjacent center PLL. For example, in the 484-pin FineLine BGA EP1S10 device, PLL 1 can drive a maximum of five channels at 840 Mbps or a maximum of 10 channels at 840 Mbps. The Quartus II software may also merge receiver and transmitter PLLs when a receiver is driving a transmitter. In this case, one fast PLL can drive both the maximum numbers of receiver and transmitter channels.

(2) The number of channels listed includes the transmitter clock output (`tx_outclock`) channel. If the design requires a DDR clock, it can use an extra data channel.

(3) These channels span across two I/O banks per side of the device. When a center PLL clocks channels in the opposite bank on the same side of the device it is called cross-bank PLL support. Both center PLLs can clock cross-bank channels simultaneously if, for example, PLL_1 is clocking all receiver channels and PLL_2 is clocking all transmitter channels. You cannot have two adjacent PLLs simultaneously clocking cross-bank receiver channels or two adjacent PLLs simultaneously clocking transmitter channels. Cross-bank allows for all receiver channels on one side of the device to be clocked on one clock while all transmitter channels on the device are clocked on the other center PLL. Crossbank PLLs are supported at full-speed, 840 Mbps. For wire-bond devices, the full-speed is 624 Mbps.

(4) These values show the channels available for each PLL without crossing another bank.

When you span two I/O banks using cross-bank support, you can route only two load enable signals total between the PLLs. When you enable `rx_data_align`, you use both `rxloadena` and `txloadena` of a PLL. That leaves no `loadena` for the second PLL.

The only way you can use the `rx_data_align` is if one of the following is true:

- The receiver PLL is only clocking receive channels (no resources for the transmitter)
- If all channels can fit in one I/O bank

Table 2–38. EP1S30 Differential Channels Note (1)

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
780-pin FineLine BGA	Transmitter (4)	70	840	18	17	17	18	(6)	(6)	(6)	(6)
			840 (5)	35	35	35	35	(6)	(6)	(6)	(6)
	Receiver	66	840	17	16	16	17	(6)	(6)	(6)	(6)
			840 (5)	33	33	33	33	(6)	(6)	(6)	(6)
956-pin BGA	Transmitter (4)	80	840	19	20	20	19	20	20	20	20
			840 (5)	39	39	39	39	20	20	20	20
	Receiver	80	840	20	20	20	20	19	20	20	19
			840 (5)	40	40	40	40	19	20	20	19
1,020-pin FineLine BGA	Transmitter (4)	80 (2) (7)	840	19 (1)	20	20	19 (1)	20	20	20	20
			840 (5),(8)	39 (1)	39 (1)	39 (1)	39 (1)	20	20	20	20
	Receiver	80 (2) (7)	840	20	20	20	20	19 (1)	20	20	19 (1)
			840 (5),(8)	40	40	40	40	19 (1)	20	20	19 (1)

Table 2–39. EP1S40 Differential Channels (Part 1 of 2) Note (1)

Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
780-pin FineLine BGA	Transmitter (4)	68	840	18	16	16	18	(6)	(6)	(6)	(6)
			840 (5)	34	34	34	34	(6)	(6)	(6)	(6)
	Receiver	66	840	17	16	16	17	(6)	(6)	(6)	(6)
			840 (5)	33	33	33	33	(6)	(6)	(6)	(6)

Table 2–39. EP1S40 Differential Channels (Part 2 of 2) *Note (1)*

Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs *(2), (3)*			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
956-pin BGA	Transmitter *(4)*	80	840	18	17	17	18	20	20	20	20
			840 *(5)*	35	35	35	35	20	20	20	20
	Receiver	80	840	20	20	20	20	18	17	17	18
			840 *(5)*	40	40	40	40	18	17	17	18
1,020-pin FineLine BGA	Transmitter *(4)*	80 (10) *(7)*	840	18 (2)	17 (3)	17 (3)	18 (2)	20	20	20	20
			840 *(5)*, *(8)*	35 (5)	35 (5)	35 (5)	35 (5)	20	20	20	20
	Receiver	80 (10) *(7)*	840	20	20	20	20	18 (2)	17 (3)	17 (3)	18 (2)
			840 *(5)*, *(8)*	40	40	40	40	18 (2)	17 (3)	17 (3)	18 (2)
1,508-pin FineLine BGA	Transmitter *(4)*	80 (10) *(7)*	840	18 (2)	17 (3)	17 (3)	18 (2)	20	20	20	20
			840 *(5)*, *(8)*	35 (5)	35 (5)	35 (5)	35 (5)	20	20	20	20
	Receiver	80 (10) *(7)*	840	20	20	20	20	18 (2)	17 (3)	17 (3)	18 (2)
			840 *(5)*, *(8)*	40	40	40	40	18 (2)	17 (3)	17 (3)	18 (2)

Table 2–40. EP1S60 Differential Channels (Part 1 of 2) *Note (1)*

Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs *(2), (3)*			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
956-pin BGA	Transmitter *(4)*	80	840	12	10	10	12	20	20	20	20
			840 *(5)*, *(8)*	22	22	22	22	20	20	20	20
	Receiver	80	840	20	20	20	20	12	10	10	12
			840 *(5)*, *(8)*	40	40	40	40	12	10	10	12

Table 2–40. EP1S60 Differential Channels (Part 2 of 2) Note (1)

Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
1,020-pin FineLine BGA	Transmitter (4)	80 (12) (7)	840	12 (2)	10 (4)	10 (4)	12 (2)	20	20	20	20
			840 (5), (8)	22 (6)	22 (6)	22 (6)	22 (6)	20	20	20	20
	Receiver	80 (10) (7)	840	20	20	20	20	12 (8)	10 (10)	10 (10)	12 (8)
			840 (5), (8)	40	40	40	40	12 (8)	10 (10)	10 (10)	12 (8)
1,508-pin FineLine BGA	Transmitter (4)	80 (36) (7)	840	12 (8)	10 (10)	10 (10)	12 (8)	20	20	20	20
			840 (5),(8)	22 (18)	22 (18)	22 (18)	22 (18)	20	20	20	20
	Receiver	80 (36) (7)	840	20	20	20	20	12 (8)	10 (10)	10 (10)	12 (8)
			840 (5),(8)	40	40	40	40	12 (8)	10 (10)	10 (10)	12 (8)

Table 2–41. EP1S80 Differential Channels (Part 1 of 2) Note (1)

Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
956-pin BGA	Transmitter (4)	80 (40) (7)	840	10	10	10	10	20	20	20	20
			840 (5),(8)	20	20	20	20	20	20	20	20
	Receiver	80	840	20	20	20	20	10	10	10	10
			840 (5),(8)	40	40	40	40	10	10	10	10
1,020-pin FineLine BGA	Transmitter (4)	92 (12) (7)	840	10 (2)	10 (4)	10 (4)	10 (2)	20	20	20	20
			840 (5),(8)	20 (6)	20 (6)	20 (6)	20 (6)	20	20	20	20
	Receiver	90 (10) (7)	840	20	20	20	20	10 (2)	10 (3)	10 (3)	10 (2)
			840 (5),(8)	40	40	40	40	10 (2)	10 (3)	10 (3)	10 (2)

Table 2–41. EP1S80 Differential Channels (Part 2 of 2) Note (1)

Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
1,508-pin FineLine BGA	Transmitter (4)	80 (72) (7)	840	10 (10)	10 (10)	10 (10)	10 (10)	20 (8)	20 (8)	20 (8)	20 (8)
			840 (5),(8)	20 (20)	20 (20)	20 (20)	20 (20)	20 (8)	20 (8)	20 (8)	20 (8)
	Receiver	80 (56) (7)	840	20	20	20	20	10 (14)	10 (14)	10 (14)	10 (14)
			840 (5),(8)	40	40	40	40	10 (14)	10 (14)	10 (14)	10 (14)

Notes to Tables 2–38 through 2–41:

(1) The first row for each transmitter or receiver reports the number of channels driven directly by the PLL. The second row below it shows the maximum channels a PLL can drive if cross bank channels are used from the adjacent center PLL. For example, in the 780-pin FineLine BGA EP1S30 device, PLL 1 can drive a maximum of 18 transmitter channels at 840 Mbps or a maximum of 35 transmitter channels at 840 Mbps. The Quartus II software may also merge transmitter and receiver PLLs when a receiver is driving a transmitter. In this case, one fast PLL can drive both the maximum numbers of receiver and transmitter channels.

(2) Some of the channels accessible by the center fast PLL and the channels accessible by the corner fast PLL overlap. Therefore, the total number of channels is not the addition of the number of channels accessible by PLLs 1, 2, 3, and 4 with the number of channels accessible by PLLs 7, 8, 9, and 10. For more information on which channels overlap, see the Stratix device pin-outs at **www.altera.com**.

(3) The corner fast PLLs in this device support a data rate of 840 Mbps for channels labeled "high" speed in the device pin-outs at **www.altera.com**.

(4) The numbers of channels listed include the transmitter clock output (`tx_outclock`) channel. An extra data channel can be used if a DDR clock is needed.

(5) These channels span across two I/O banks per side of the device. When a center PLL clocks channels in the opposite bank on the same side of the device it is called cross-bank PLL support. Both center PLLs can clock cross-bank channels simultaneously if say PLL_1 is clocking all receiver channels and PLL_2 is clocking all transmitter channels. You cannot have two adjacent PLLs simultaneously clocking cross-bank receiver channels or two adjacent PLLs simultaneously clocking transmitter channels. Cross-bank allows for all receiver channels on one side of the device to be clocked on one clock while all transmitter channels on the device are clocked on the other center PLL. Crossbank PLLs are supported at full-speed, 840 Mbps. For wire-bond devices, the full-speed is 624 Mbps.

(6) PLLs 7, 8, 9, and 10 are not available in this device.

(7) The number in parentheses is the number of slow-speed channels, guaranteed to operate at up to 462 Mbps. These channels are independent of the high-speed differential channels. For the location of these channels, see the device pin-outs at **www.altera.com**.

(8) See the Stratix device pin-outs at **www.altera.com**. Channels marked "high" speed are 840 MBps and "low" speed channels are 462 MBps.

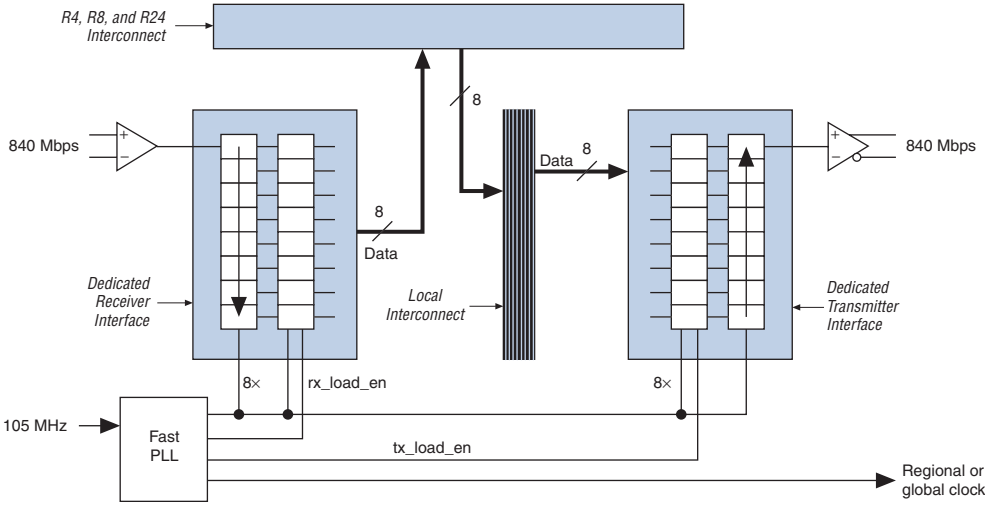
The high-speed differential I/O circuitry supports the following high speed I/O interconnect standards and applications:

- UTOPIA IV
- SPI-4 Phase 2 (POS-PHY Level 4)
- SFI-4
- 10G Ethernet XSBI
- RapidIO
- HyperTransport

Dedicated Circuitry

Stratix devices support source-synchronous interfacing with LVDS, LVPECL, 3.3-V PCML, or HyperTransport signaling at up to 840 Mbps. Stratix devices can transmit or receive serial channels along with a low-speed or high-speed clock. The receiving device PLL multiplies the clock by a integer factor W (W = 1 through 32). For example, a HyperTransport application where the data rate is 800 Mbps and the clock rate is 400 MHz would require that W be set to 2. The SERDES factor J determines the parallel data width to deserialize from receivers or to serialize for transmitters. The SERDES factor J can be set to 4, 7, 8, or 10 and does not have to equal the PLL clock-multiplication W value. For a J factor of 1, the Stratix device bypasses the SERDES block. For a J factor of 2, the Stratix device bypasses the SERDES block, and the DDR input and output registers are used in the IOE. See Figure 2–73.

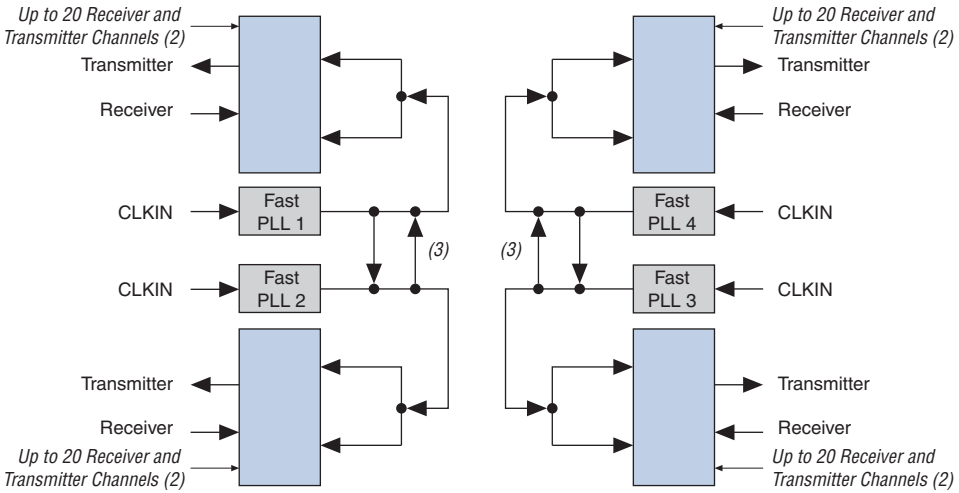
Figure 2–73. High-Speed Differential I/O Receiver / Transmitter Interface Example



An external pin or global or regional clock can drive the fast PLLs, which can output up to three clocks: two multiplied high-speed differential I/O clocks to drive the SERDES block and/or external pin, and a low-speed clock to drive the logic array.

The Quartus II MegaWizard® Plug-In Manager only allows the implementation of up to 20 receiver or 20 transmitter channels for each fast PLL. These channels operate at up to 840 Mbps. The receiver and transmitter channels are interleaved such that each I/O bank on the left and right side of the device has one receiver channel and one transmitter channel per LAB row. Figure 2–74 shows the fast PLL and channel layout in EP1S10, EP1S20, and EP1S25 devices. Figure 2–75 shows the fast PLL and channel layout in the EP1S30 to EP1S80 devices.

Figure 2–74. Fast PLL & Channel Layout in the EP1S10, EP1S20 or EP1S25 Devices *Note (1)*



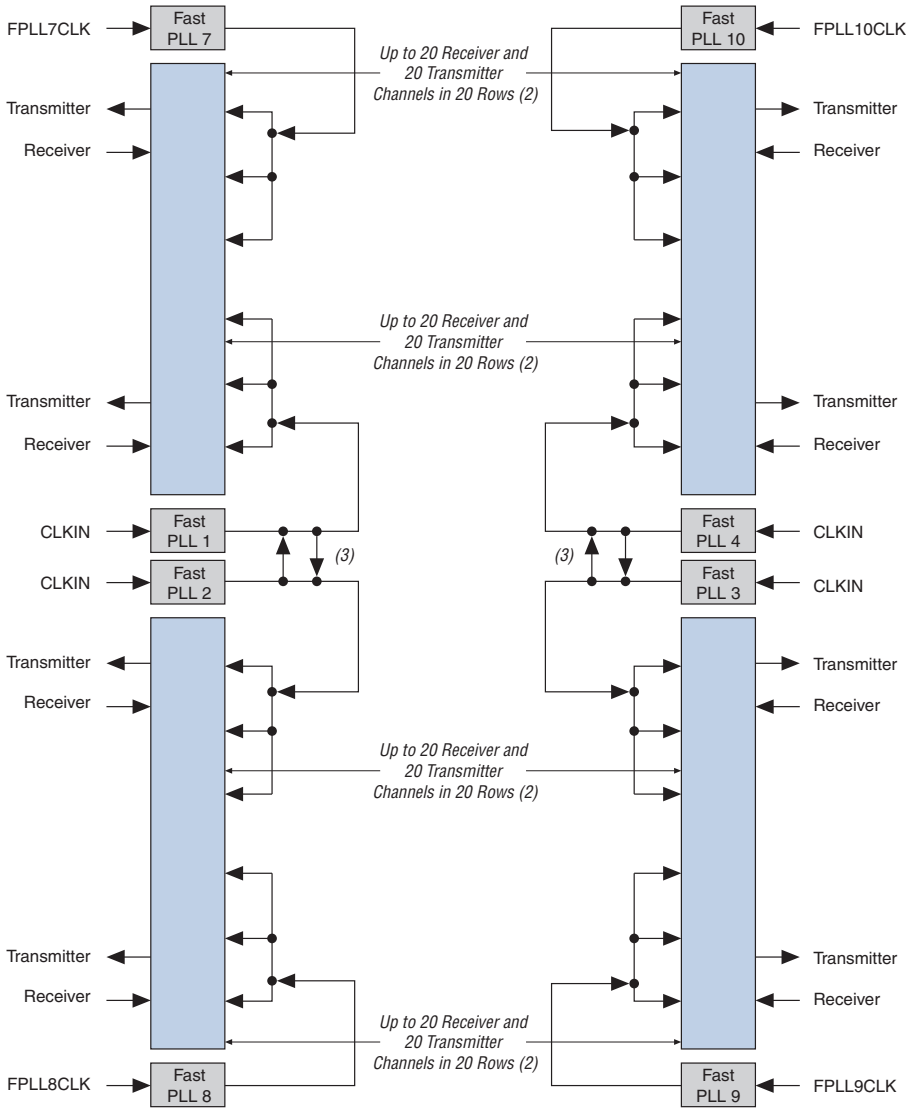
Notes to Figure 2–74:

(1) Wire-bond packages support up to 624 Mbps.

(2) See Table 2–41 for the number of channels each device supports.

(3) There is a multiplexer here to select the PLL clock source. If a PLL uses this multiplexer to clock channels outside of its bank quadrant, those clocked channels support up to 840 Mbps for "high" speed channels and 462 Mbps for "low" speed channels, as labeled in the device pin-outs at **www.altera.com**.

Figure 2–75. Fast PLL & Channel Layout in the EP1S30 to EP1S80 Devices Note (1)



Notes to Figure 2–75:

(1) Wire-bond packages support up to 624 Mbps.

(2) See Table 2–38 through 2–41 for the number of channels each device supports.

(3) There is a multiplexer here to select the PLL clock source. If a PLL uses this multiplexer to clock channels outside of its bank quadrant, those clocked channels support up to 840 Mbps for "high" speed channels and 462 Mbps for "low" speed channels as labeled in the device pin-outs at **www.altera.com**.

The transmitter external clock output is transmitted on a data channel. The txclk pin for each bank is located in between data transmitter pins. For ×1 clocks (e.g., 622 Mbps, 622 MHz), the high-speed PLL clock bypasses the SERDES to drive the output pins. For half-rate clocks (e.g., 622 Mbps, 311 MHz) or any other even-numbered factor such as 1/4, 1/7, 1/8, or 1/10, the SERDES automatically generates the clock in the Quartus II software.

For systems that require more than four or eight high-speed differential I/O clock domains, a SERDES bypass implementation is possible using IOEs.

Byte Alignment

For high-speed source synchronous interfaces such as POS-PHY 4, XSBI, RapidIO, and HyperTransport technology, the source synchronous clock rate is not a byte- or SERDES-rate multiple of the data rate. Byte alignment is necessary for these protocols since the source synchronous clock does not provide a byte or word boundary since the clock is one half the data rate, not one eighth. The Stratix device's high-speed differential I/O circuitry provides dedicated data realignment circuitry for user-controlled byte boundary shifting. This simplifies designs while saving LE resources. An input signal to each fast PLL can stall deserializer parallel data outputs by one bit period. You can use an LE-based state machine to signal the shift of receiver byte boundaries until a specified pattern is detected to indicate byte alignment.

Power Sequencing & Hot Socketing

Because Stratix devices can be used in a mixed-voltage environment, they have been designed specifically to tolerate any possible power-up sequence. Therefore, the VCCIO and VCCINT power supplies may be powered in any order.

Although you can power up or down the VCCIO and VCCINT power supplies in any sequence, you should not power down any I/O banks that contain configuration pins while leaving other I/O banks powered on. For power up and power down, all supplies (VCCINT and all VCCIO power planes) must be powered up and down within 100 ms of each other. This prevents I/O pins from driving out.

Signals can be driven into Stratix devices before and during power up without damaging the device. In addition, Stratix devices do not drive out during power up. Once operating conditions are reached and the device is configured, Stratix devices operate as specified by the user. For more information, see *Hot Socketing* in the *Selectable I/O Standards in Stratix & Stratix GX Devices* chapter in the *Stratix Device Handbook, Volume 2*.