Stratix

Programmable Logic Device Family

February 2002, ver. 1.0

Data Sheet

Introduction

Preliminary Information

The Stratix family of programmable logic devices (PLDs) is based on a 1.5-V, 0.13-µm, all-layer copper SRAM process, with densities up to 114,140 logic elements (LEs) and up to 10 Mbits of RAM. Stratix devices offer up to 28 digital signal processing (DSP) blocks with up to 224 (9-bit × 9-bit) embedded multipliers, optimized for DSP applications that enable efficient implementation of high-performance filters and multipliers. Stratix devices support various I/O standards and also offer a complete clock management solution with its hierarchical clock structure with up to 420-MHz performance and up to 12 phase-locked loops (PLLs).

Features...

- 10,570 to 114,140 LEs; see Table 1
- Up to 10,118,016 RAM bits (1,264,752 bytes) available without reducing logic resources
- TriMatrix™ memory consisting of three RAM block sizes to implement true dual-port memory and first-in first-out (FIFO) buffers up to 312 MHz
- High-speed DSP blocks provide dedicated implementation of multipliers (at up to 250 MHz), multiply- accumulate functions, and finite impulse response (FIR) filters
- Up to 16 global clocks with 22 clocking resources per device region
- Up to 12 enhanced PLLs per device provide spread spectrum, programmable bandwidth, clock switch-over, real-time PLL reconfiguration, and advanced multiplication and phase shifting
- Support for numerous single-ended and differential I/O standards
- High-speed differential I/O support on up to 116 channels with up to 80 channels optimized for 840 megabits per second (Mbps)
- Support for high-speed networking and communications bus standards including RapidIO, UTOPIA IV, CSIX, HyperTransport™ technology, 10G Ethernet XSBI, SPI-4 Phase 2 (POS-PHY Level 4), and SFI-4
- Terminator™ technology provides on-chip termination for differential and single-ended I/O pins with impedance matching
- Support for high-speed external memory, including zero bus turnaround (ZBT) SRAM, quad data rate (QDR and QDRII) SRAM, double data rate (DDR) SDRAM, DDR fast cycle RAM (FCRAM), and single data rate (SDR) SDRAM
- Support for multiple intellectual property megafunctions from Altera MegaCore® functions and Altera Megafunction Partners Program (AMPP℠) megafunctions
- Support for remote configuration updates

Table 1. Stratix Device Features (Part 1 of 2)

Feature	EP1S10	EP1S20	EP1S25	EP1S30
LEs	10,570	18,460	25,660	32,470
M512 RAM blocks (32 × 18 bits)	94	194	224	295
M4K RAM blocks (128 × 36 bits)	60	82	138	171
MegaRAM blocks (4K × 144 bits)	1	2	2	4
Total RAM bits	920,448	1,669,248	1,944,576	3,317,184
DSP blocks	6	10	10	12
Embedded multipliers *(1)*	48	80	80	96
PLLs	6	6	6	10
Maximum user I/O pins	422	582	702	726

Stratix Device Features (Part 2 of 2)

Feature	EP1S40	EP1S60	EP1S80	EP1S120
LEs	41,250	57,120	79,040	114,140
M512 RAM blocks (32 × 18 bits)	384	574	767	1,118
M4K RAM blocks (128 × 36 bits)	183	292	364	520
MegaRAM blocks (4K × 144 bits)	4	6	9	12
Total RAM bits	3,423,744	5,215,104	7,427,520	10,118,016
DSP blocks	14	18	22	28
Embedded multipliers *(1)*	112	144	176	224
PLLs	12	12	12	12
Maximum user I/O pins	818	1,018	1,234	1,310

Note to Table 1:

(1) This parameter lists the total number of 9 × 9-bit multipliers for each device. For the total number of 18 × 18-bit multipliers per device, divide the total number of 9 × 9-bit multipliers by 2. For the total number of 36 × 36-bit multipliers per device, decide the total number of 9 × 9-bit multipliers by 8.

Stratix devices are available in space-saving FineLine BGA™ and ball-grid array (BGA) packages (see Tables 2 through 4)

Table 2. Stratix Package Options & I/O Pin Counts

Device	672-Pin BGA	956-Pin BGA	672-Pin FineLine BGA	780-Pin FineLine BGA	1,020-Pin FineLine BGA	1,508-Pin FineLine BGA	1,923-Pin FineLine BGA
EP1S10	341		341	422			
EP1S20	422		422	582			
EP1S25	469		469	593	702		
EP1S30		679		593	726		
EP1S40		679			769	818	
EP1S60		679			769	1,018	
EP1S80		679				1,199	1,234
EP1S120							1,310

Table 3. Stratix BGA Package Sizes

Dimension	672 Pin	956 Pin
Pitch (mm)	1.27	1.27
Area (mm^2)	1,225	1,600
Length × width (mm × mm)	35 × 35	40 × 40

Table 4. Stratix FineLine BGA Package Sizes

Dimension	672 Pin	780 Pin	1,020 Pin	1,508 Pin	1,923 Pin
Pitch (mm)	1.00	1.00	1.00	1.00	1.00
Area (mm^2)	729	841	1,089	1,600	2,025
Length × width (mm × mm)	27 × 27	29 × 29	33 × 33	40 × 40	45 × 45

Table of Contents

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 provides 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 312 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 312 MHz. These blocks are grouped into columns across the device in between certain LABs.

MegaRAM 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 300 MHz. Several MegaRAM blocks are located individually or in pairs within the device's logic array.

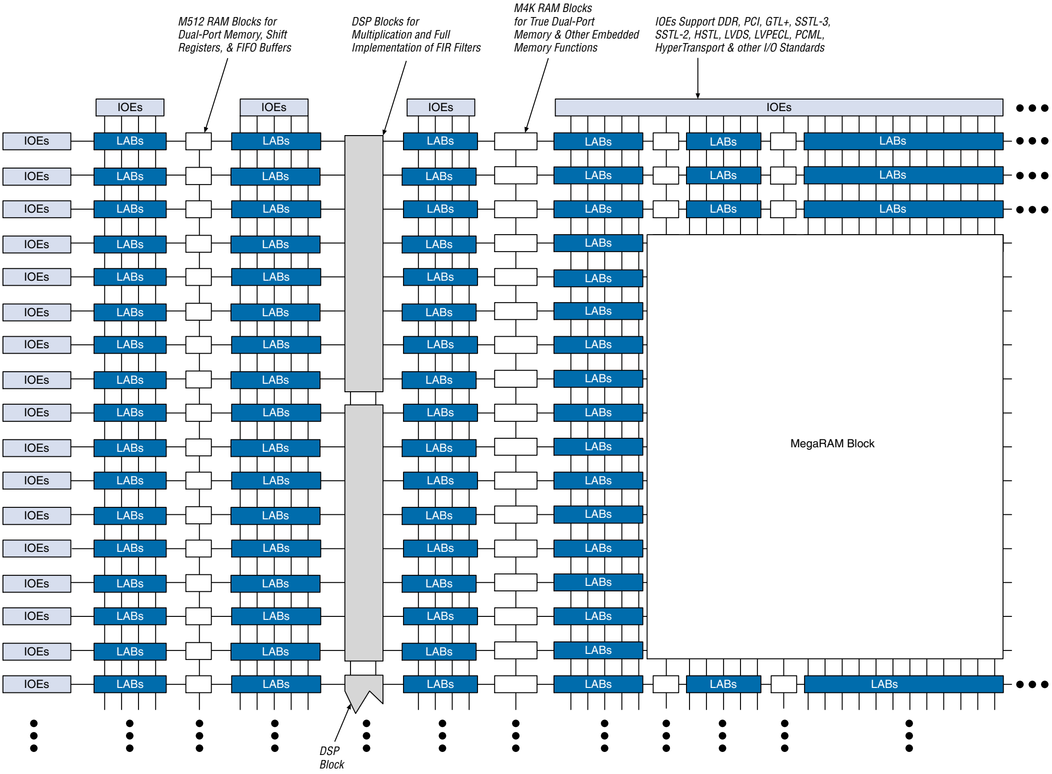
Digital signal processing (DSP) blocks can implement up to either eight full-precision 9 × 9-bit multipliers, four full-precision 18 × 18-bit multipliers, or one full-precision 36 × 36-bit multiplier with add or subtract features. These blocks also contain 18-bit input shift registers for digital signal 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, PCML, or HyperTransport technology I/O standards.

Figure 1 shows an overview of the Stratix device.

Figure 1. Stratix Block Diagram



The number of M512 RAM, M4K RAM, and DSP blocks varies by device along with row and column numbers and MegaRAM blocks. Table 5 lists the resources available in Stratix devices.

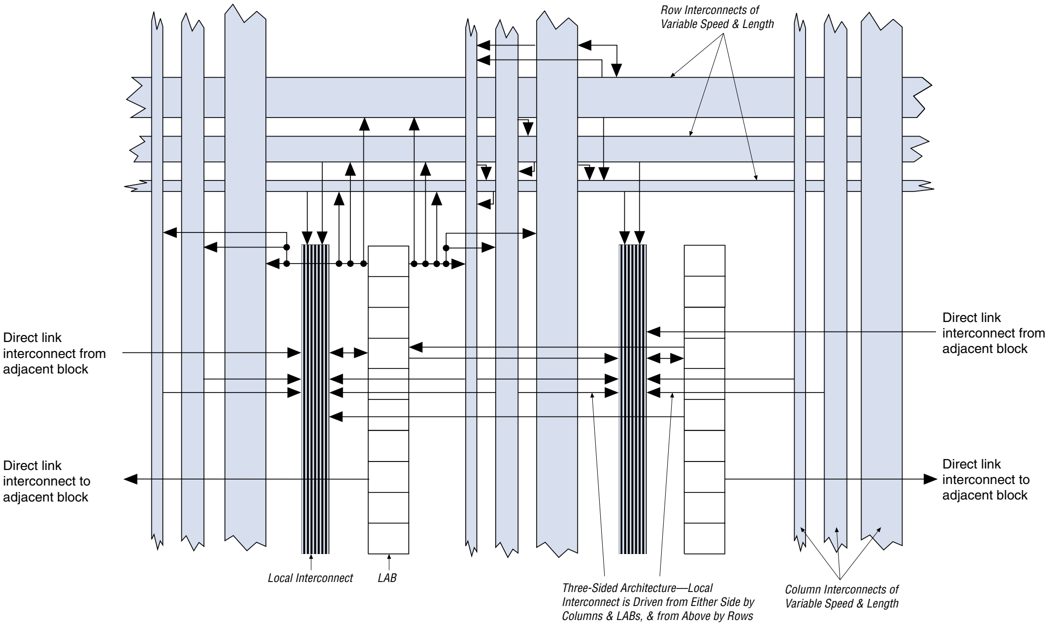
Table 5. Stratix Device Resources

Device	M512 RAM Columns/Blocks	M4K RAM Columns/Blocks	MegaRAM 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
EP1S120	11 / 1,118	4 / 520	12	2 / 28	101	130

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 shows the Stratix LAB.

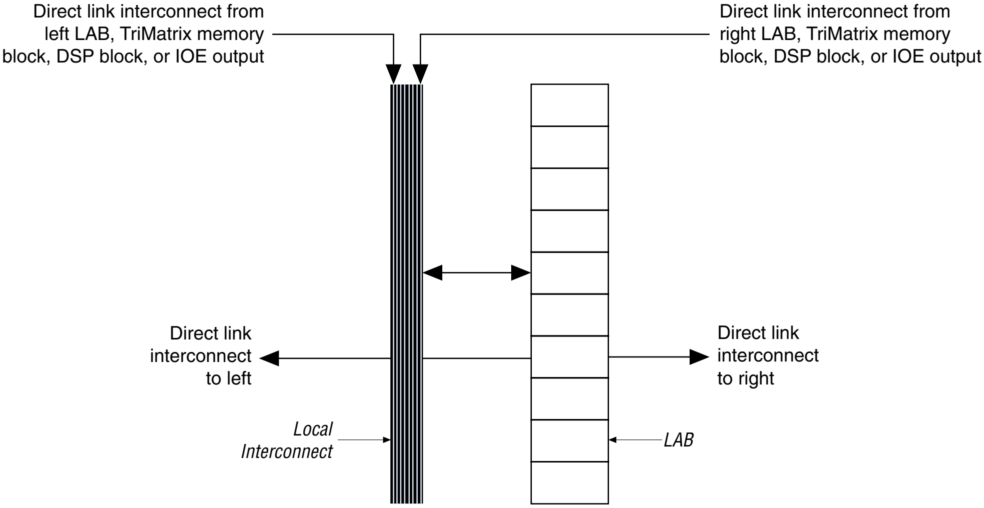
Figure 2. Stratix LAB Structure



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 3 shows the direct link connection.

Figure 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.

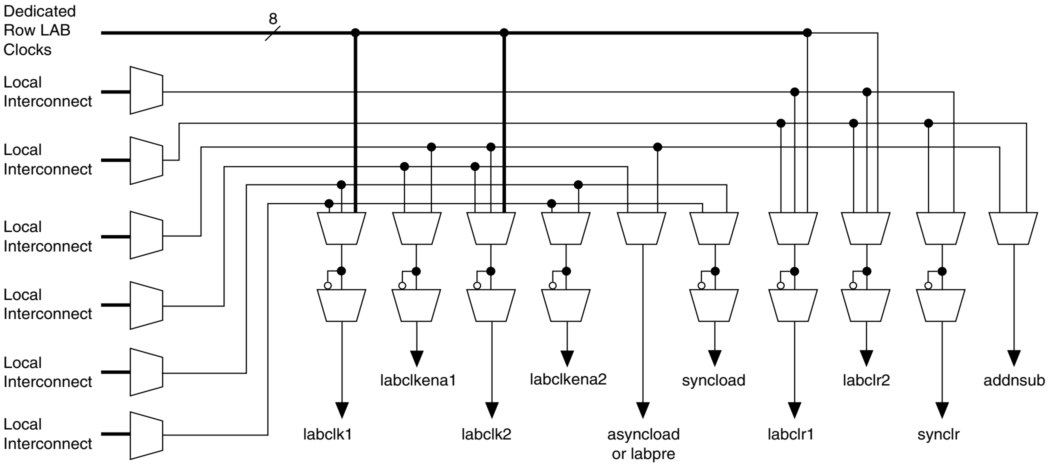
Each LAB can use two clocks and two clock enable signals. Each LAB's clock and clock enable signals are linked. For example, any LE in a particular LAB using the `labclk1` signal will also use `labclkena1`. If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock 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 additional 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 4 shows the LAB control signal generation circuit.

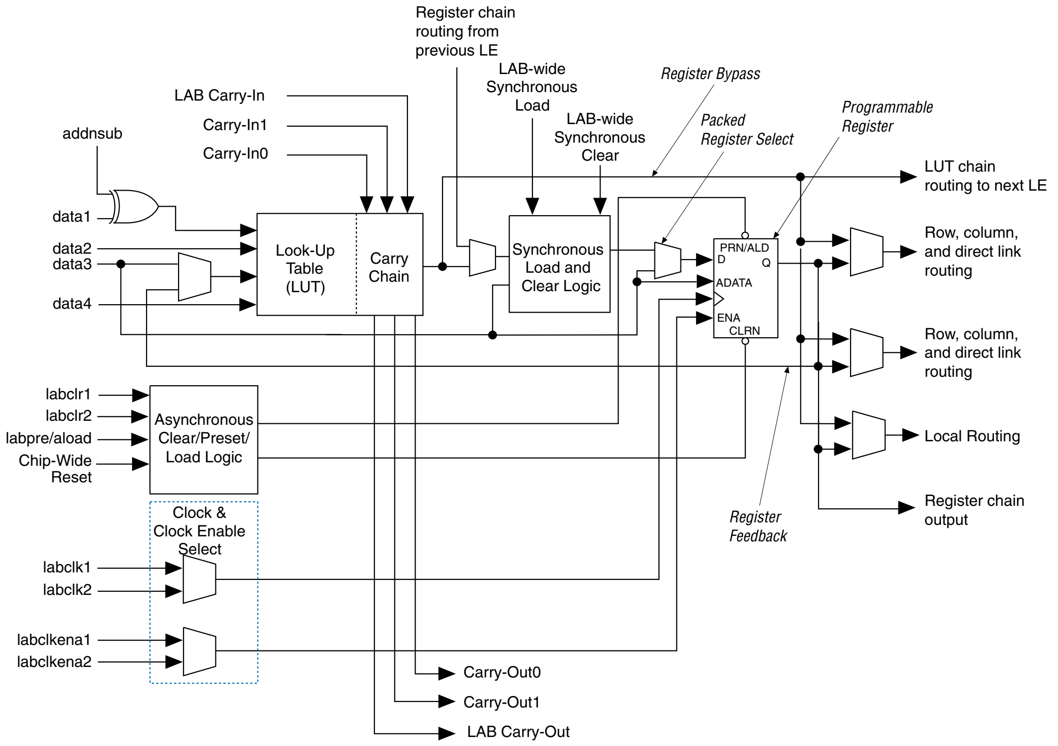
Figure 4. LAB-Wide Control Signals



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 5.

Figure 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 18 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 A 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

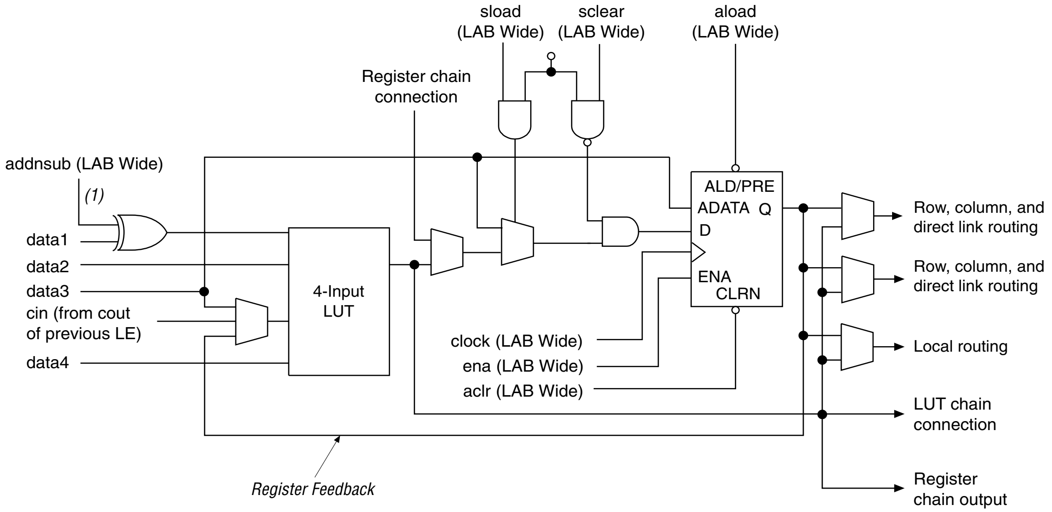
Each mode uses LE resources differently. In each mode, eight available inputs to the LE—the four data inputs from the LAB local interconnect; `carry-in0` and `carry-in1` from the previous LE; the LAB carry-in from the previous carry-chain LAB; and the register chain connection—are directed to different 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, the designer 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 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 6. LE in Normal Mode



Note to Figure 6:

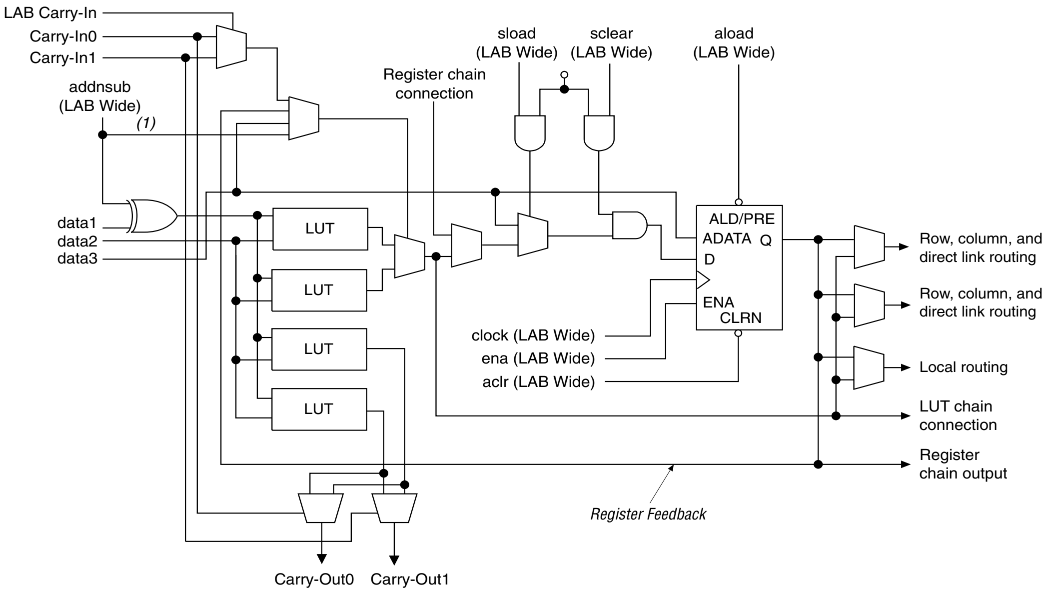
(1) This signal is only allowed in normal mode if the LE is at the end of an adder/subtractor chain.

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 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 7. LE in Dynamic Arithmetic Mode



Note to Figure 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.

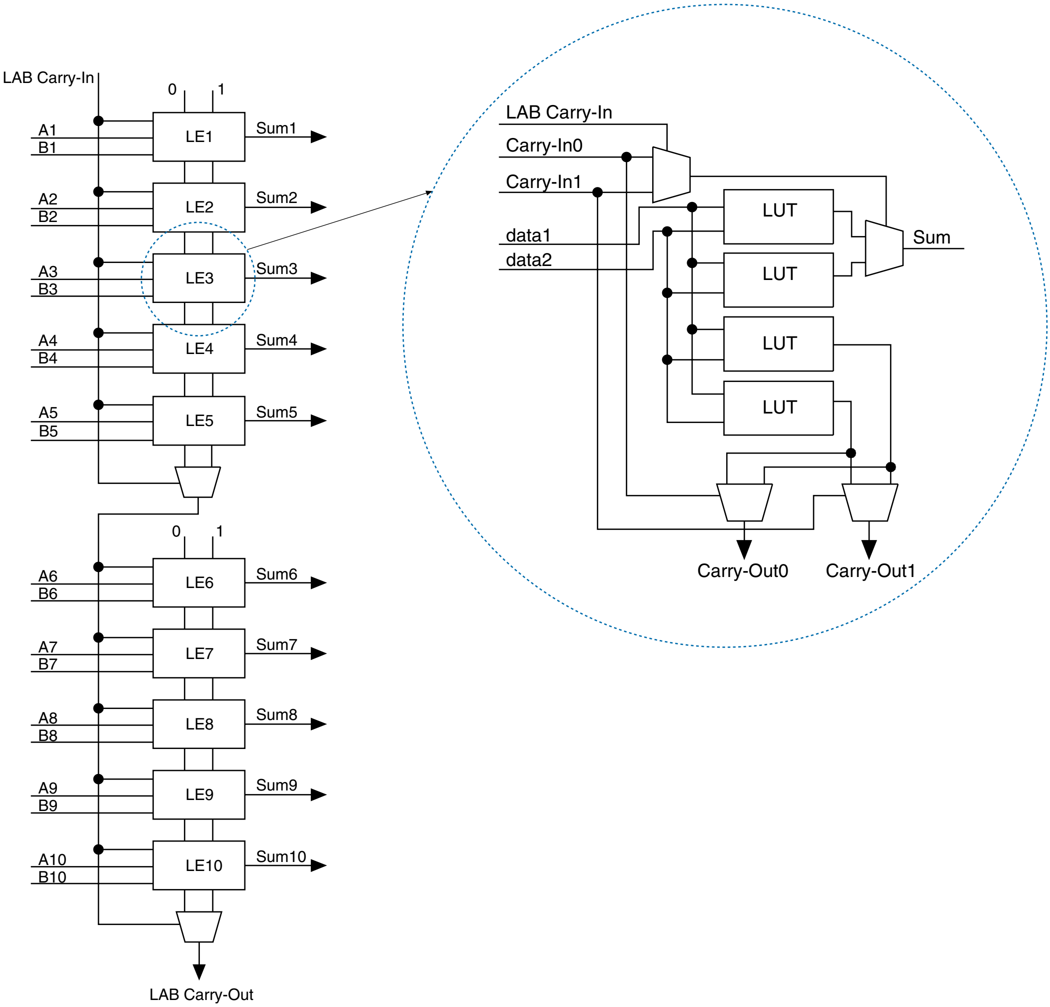
The speed advantage of the carry-select chain is in the parallel precomputation of carry chains. Since the LAB carry-in selects the precomputed carry chain, not every LE is in the critical path. Only the propagation delay between LAB carry-in generation (LE 5 and LE 10) are now part of the critical path.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

Figure 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 the designer 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 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.

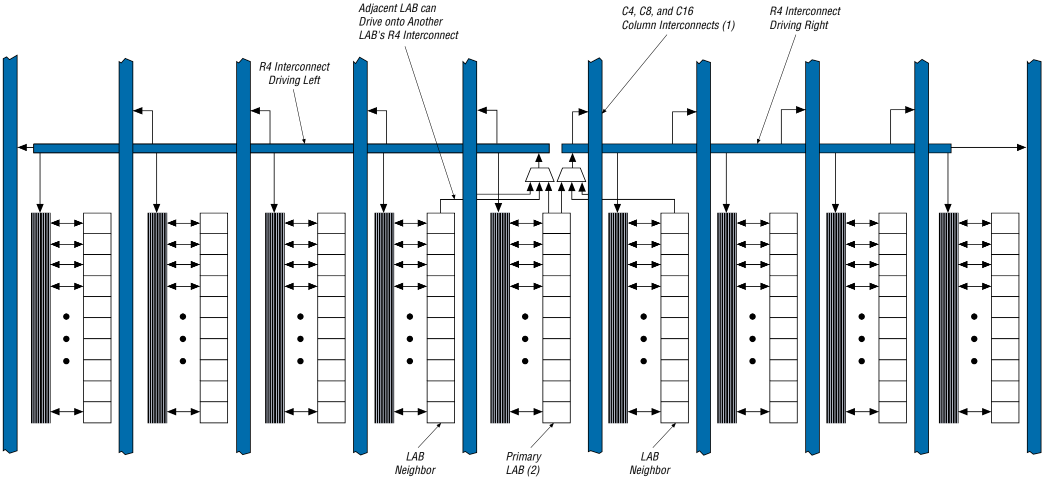
The MultiTrack interconnect consists of row and column interconnects that span fixed distances. A routing structure with fixed length resources for all devices allows predictable and repeatable performance when migrating through different device densities. Dedicated row interconnects route signals to 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 MegaRAM 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 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 9. R4 Interconnect Connections



Notes to Figure 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 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.

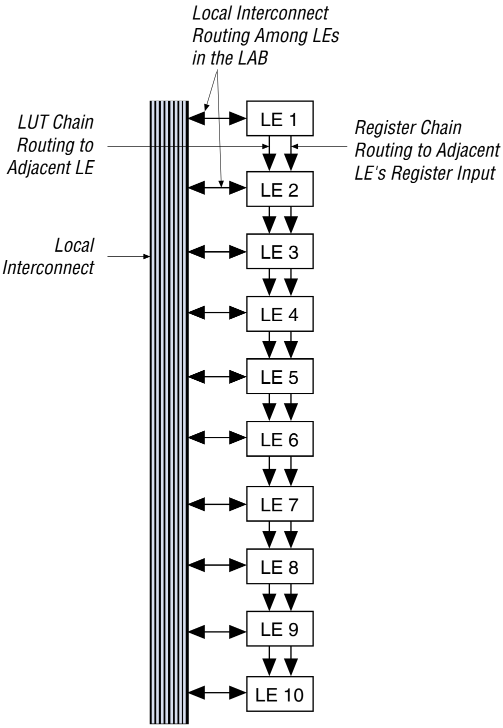
R24 row interconnects span 24 LABs and provide the fastest resource for long row connections between LABs, TriMatrix memory, DSP blocks, and IOEs. The R24 row interconnects can cross MegaRAM blocks. R24 row interconnects drive to other row or column interconnects at every fourth LAB and do not drive 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

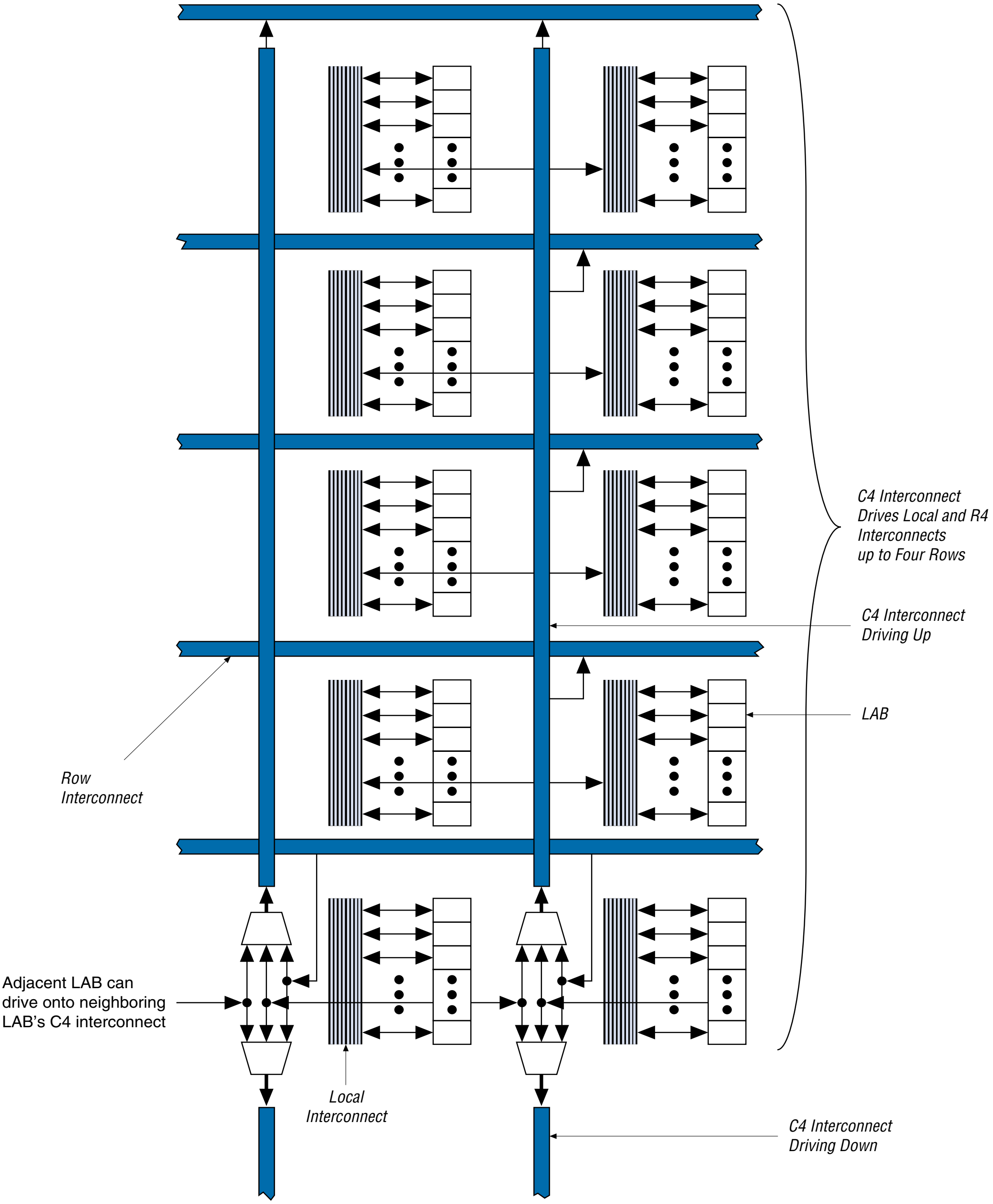
Stratix devices include an enhanced interconnect structure within LABs for routing LE output to LE input connections faster using LUT chain connections and register chain connections. The LUT chain connection allows the combinatorial output of an LE to directly drive the fast input of the LE right below 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 10 shows the LUT chain and register chain interconnects.

Figure 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 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 11. C4 Interconnect Connections *Note (1)*



Note to Figure 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 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 MegaRAM 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 6 shows the Stratix device’s routing scheme.

Table 6. 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	MegaRAM M512K 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			✓	✓	✓	✓		✓	✓				▨				
MegaRAM Block								✓	✓					▨			
DSP Blocks			✓	✓	✓	✓		✓	✓						▨		
Column IOE				✓				✓	✓	✓						▨	
Row IOE				✓		✓	✓	✓	✓	✓							▨

TriMatrix Memory

TriMatrix memory consists of three types of RAM blocks: M512, M4K, and MegaRAM 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 7 shows the size and features of the different RAM blocks.

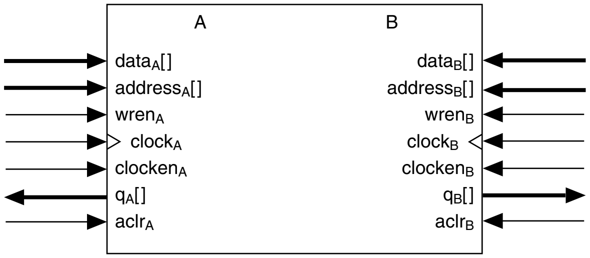
Table 7. TriMatrix Memory Features

Memory Feature	M512 RAM Block (32 × 18 Bits)	M4K RAM Block (128 × 36 Bits)	MegaRAM Block (4K × 144 Bits)
Maximum performance	312 MHz	312 MHz	300 MHz
True dual-port memory		✓	✓
Simple dual-port memory	✓	✓	✓
Single-port memory	✓	✓	✓
Byte enable		✓	✓
Parity bits	✓	✓	✓
Shift register	✓	✓	
Mixed clock mode	✓	✓	✓
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

Memory Modes

TriMatrix memory blocks include input registers that synchronize writes and output registers to pipeline designs and improve system performance. M4K and MegaRAM 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 12 shows true dual-port memory.

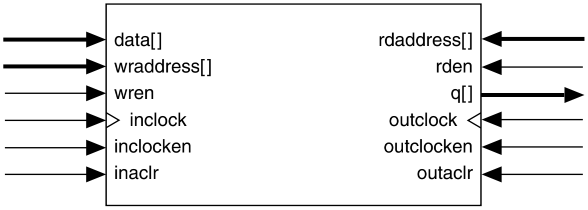
Figure 12. True Dual-Port Memory Configuration



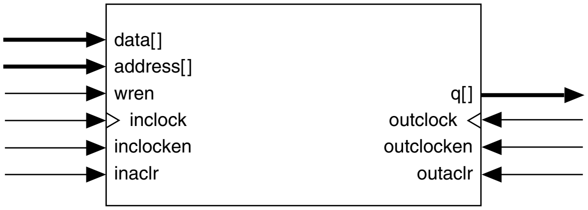
In addition to true dual-port memory, the memory blocks support simple dual-port and single-port RAM. Simple dual-port memory supports a simultaneous read and write. Single-port memory supports non-simultaneous reads and writes. Figure 13 shows these different RAM memory port configurations for TriMatrix memory.

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

Simple Dual-Port Memory



Single-Port Memory *(1)*



Note to Figure 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 fully synchronous 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. Pseudo-asynchronous 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 allowed. 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.

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. Designers can also use parity-size data words to store user-specified control bits. In the M4K and MegaRAM blocks, byte enables are also available for data input masking during write operations.

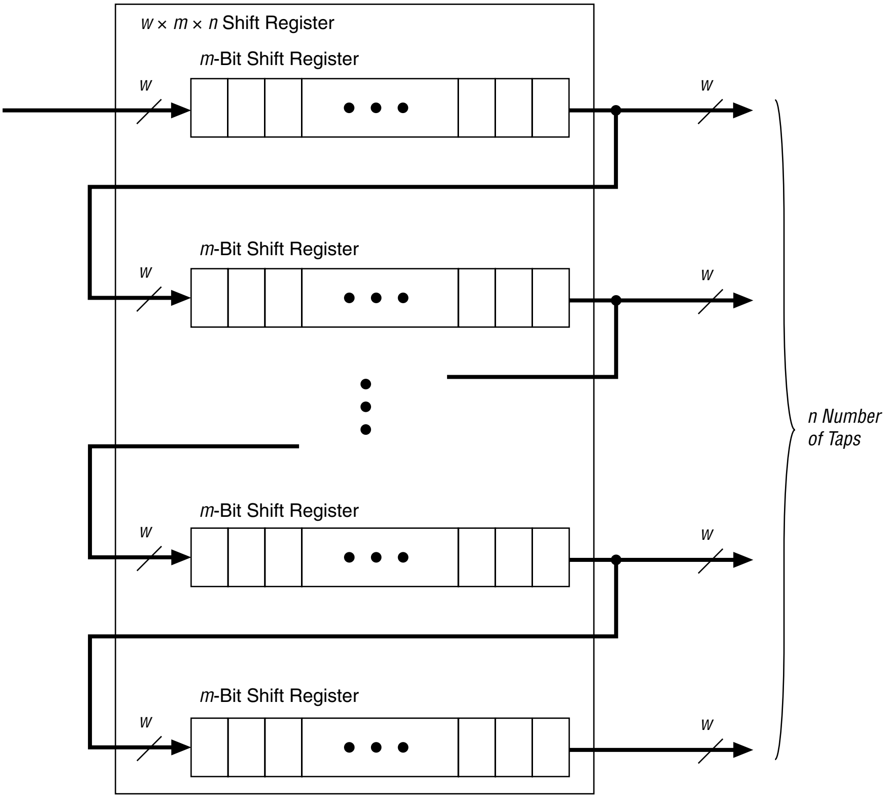
Shift Register Support

The designer 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 14 shows the TriMatrix memory block in the shift register mode.

Figure 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 MegaRAM 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. The designer 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, the designer 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 8 summarizes the possible M512 RAM block configurations.

Table 8. 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 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 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 16 shows the M512 RAM block to logic array interface.

Figure 15. M512 RAM Block Control Signals

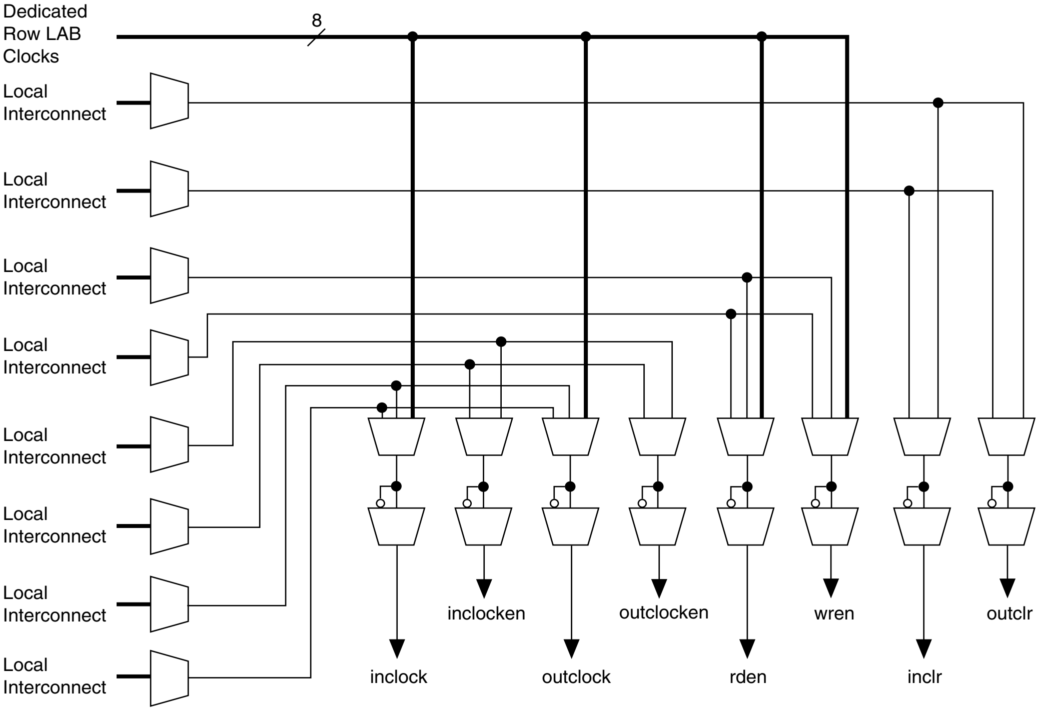
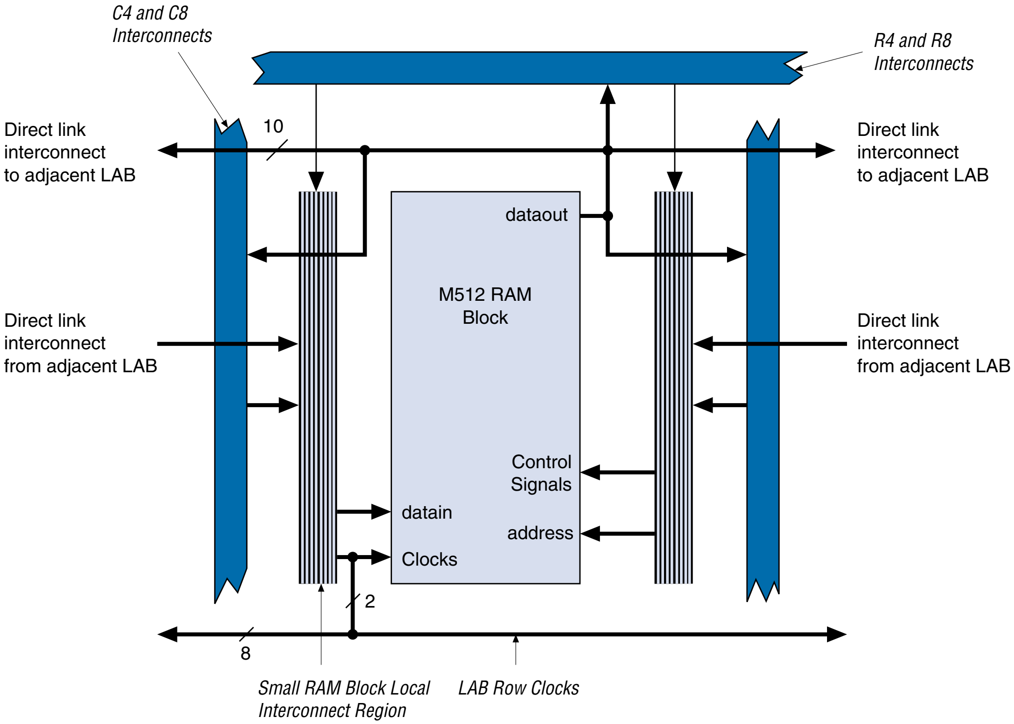


Figure 16. M512 RAM Block LAB Row Interface



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, the designer 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 9 and 10 summarize the possible M4K RAM block configurations.

Table 9. 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 10. 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, the designer 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 11 summarizes the byte selection.

Table 11. 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 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 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 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 18 shows the M4K RAM block to logic array interface.

Figure 17. M4K RAM Block Control Signals

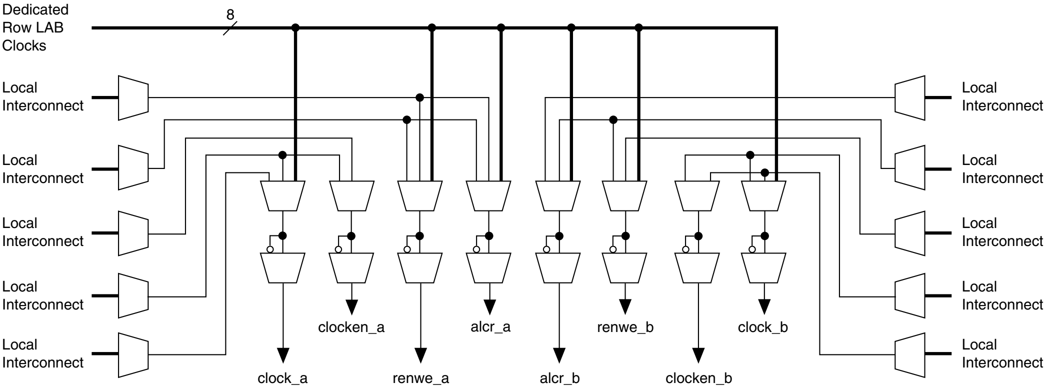
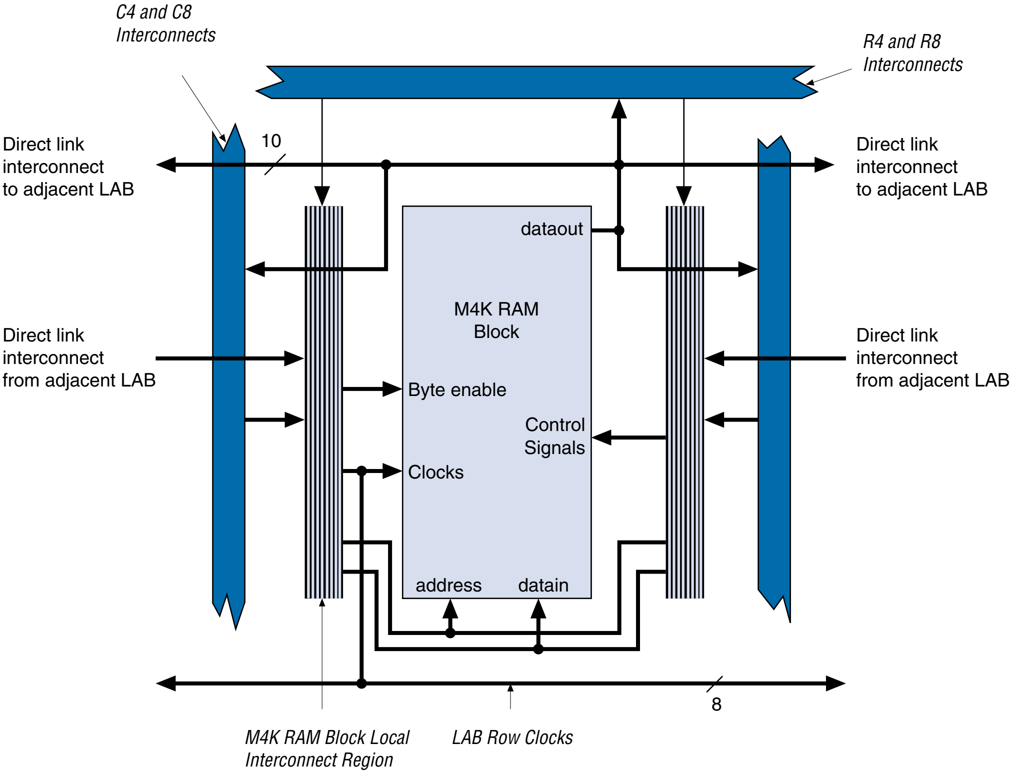


Figure 18. M4K RAM Block LAB Row Interface



MegaRAM Block

The largest TriMatrix memory block, the MegaRAM 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 MegaRAM block can be configured in the following modes:

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

The designer cannot use an initialization file to initialize the contents of a MegaRAM block. All MegaRAM block contents power up to an undefined value. Only synchronous operation is supported in the MegaRAM 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 12 and 13 summarizes the possible MegaRAM block configurations:

Table 12. MegaRAM 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 13. MegaRAM 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. Table 14 and Table 15 summarize the byte selection.

Table 14. Byte Enable for MegaRAM 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 15. MegaRAM 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]

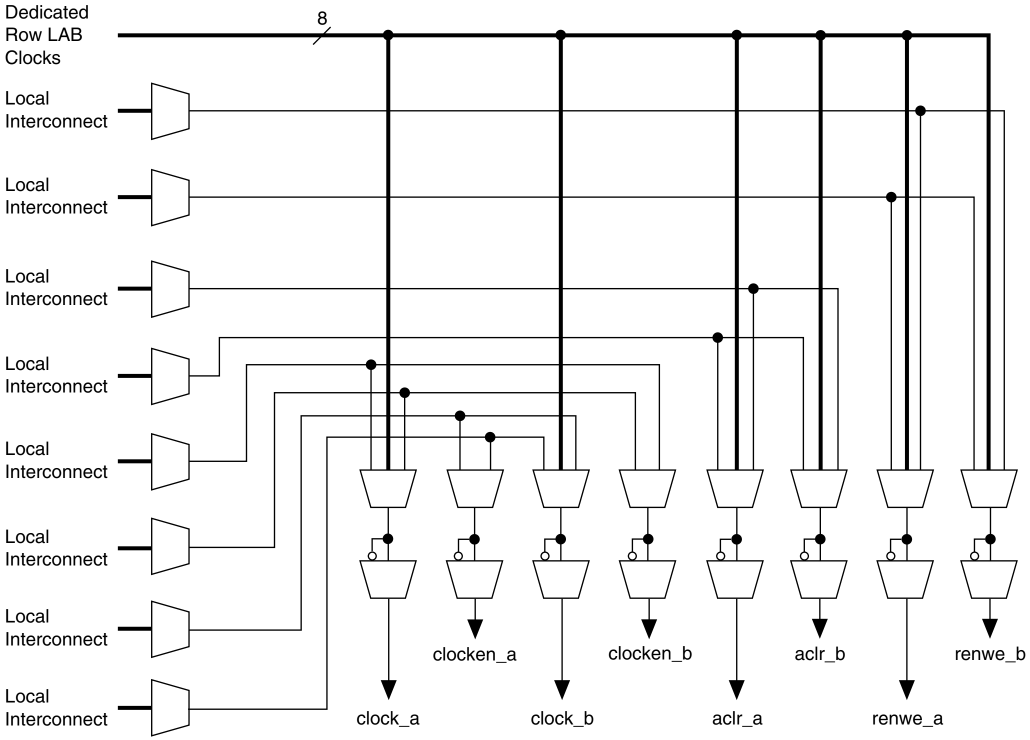
Notes to Tables 14 and 15:

(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.

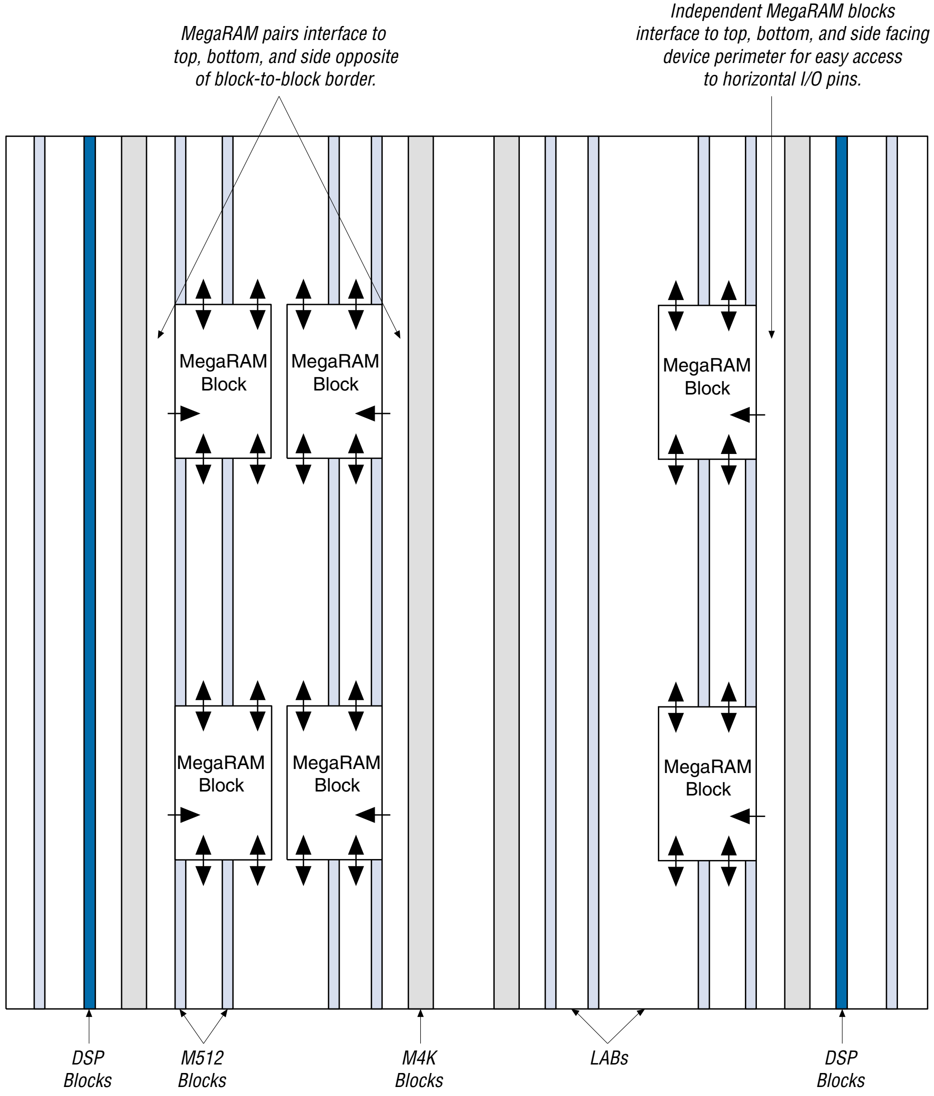
Similar to all RAM blocks, MegaRAM blocks can have different clocks on their inputs and outputs. All input registers—`renwe`, `datain`, address, and byte enable registers—are clocked together from either of the two clocks feeding the block. The output register can be bypassed. The eight `labclk` signals or local interconnect can drive the control signals for the A and B ports of the MegaRAM 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 19.

Figure 19. MegaRAM Block Control Signals



One of the MegaRAM 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 MegaRAM blocks are paired next to each other. In this case, the side of the MegaRAM block opposite the common side of the two blocks contains the input interface. The top and bottom sides of any MegaRAM 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 20 shows an example floorplan for the EP1S60 device and the location of the MegaRAM interfaces.

Figure 20. EP1S60 Device with MegaRAM Interface Locations *Note (1)*



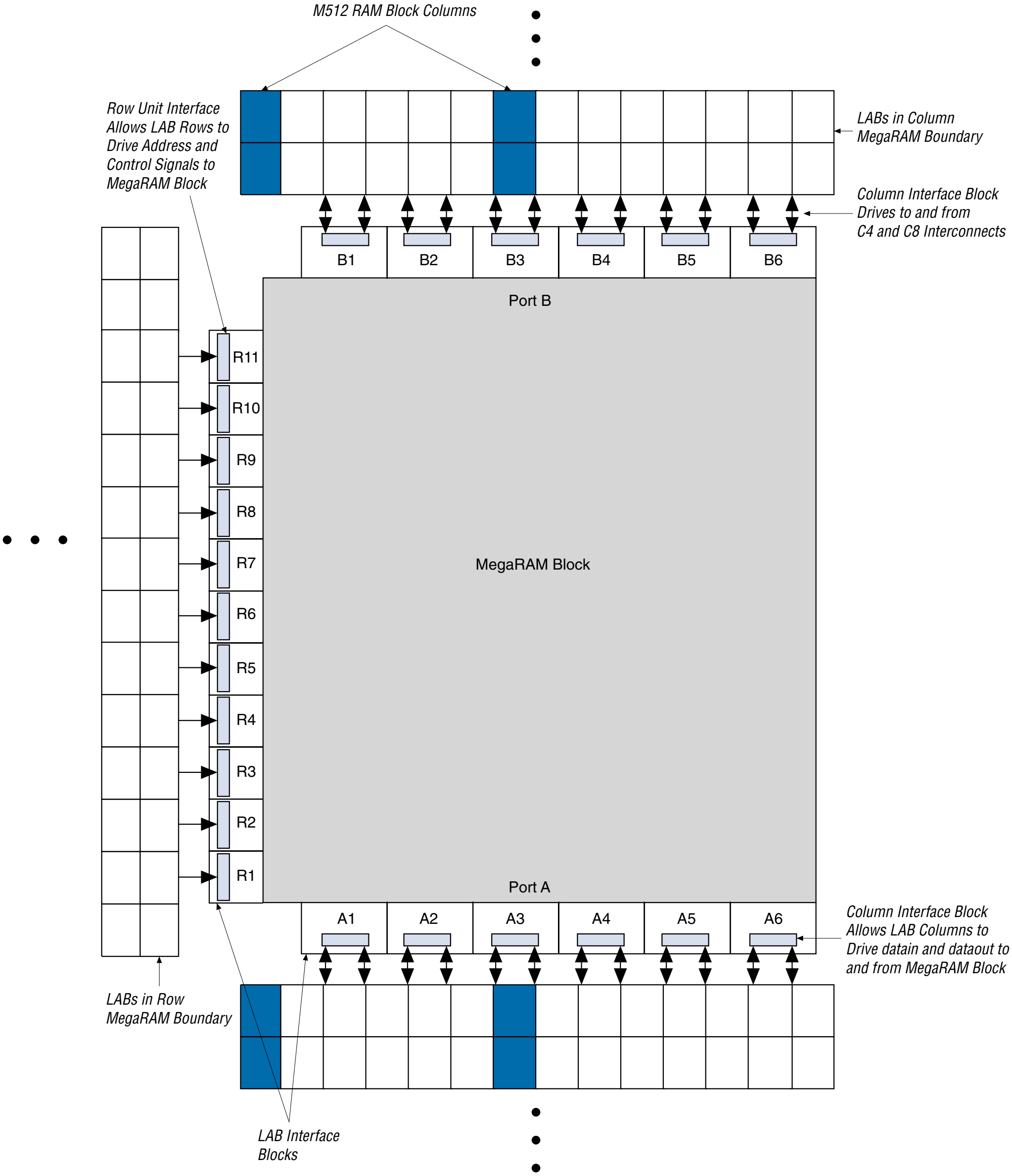
Note to Figure 20:

(1) Device shown is an EP1S60 device. The number and position of MegaRAM blocks varies in other devices.

The MegaRAM block local interconnect is driven by the R4, R8, C4, C8, and direct link interconnects from adjacent LABs. For independent MegaRAM blocks, up to 10 direct link address and control signal input connections to the MegaRAM Block are possible from the left adjacent LABs for MegaRAM blocks facing to the left, and another 10 possible from the right adjacent LABs for MegaRAM blocks facing to the right. For column interfacing, every MegaRAM column unit connects to the right and left column lines, allowing each MegaRAM column unit to communicate directly with three columns of LABs. Figures 21 through 23 show the interface between the MegaRAM block and the logic array.

Table 16 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).

Figure 21. Left-Facing MegaRAM to Interconnect Interface *Notes (1), (2)*



Notes to Figure 21:

(1) Only R24 and C16 interconnects cross the MegaRAM block boundaries.

(2) The right-facing MegaRAM 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 MegaRAM blocks.

Figure 22. MegaRAM Row Unit Interface to Interconnect

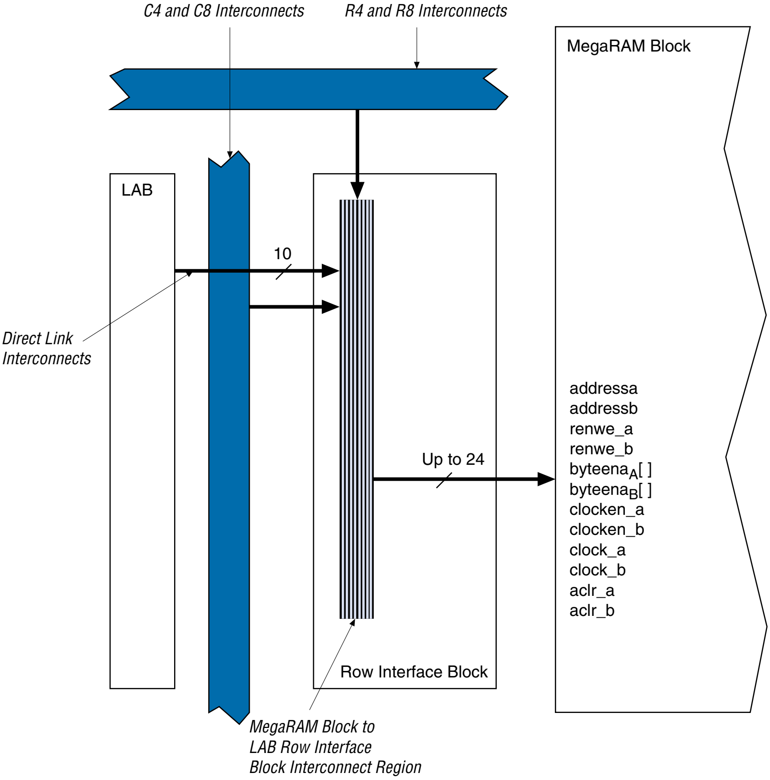


Figure 23. MegaRAM Column Unit Interface to Interconnect

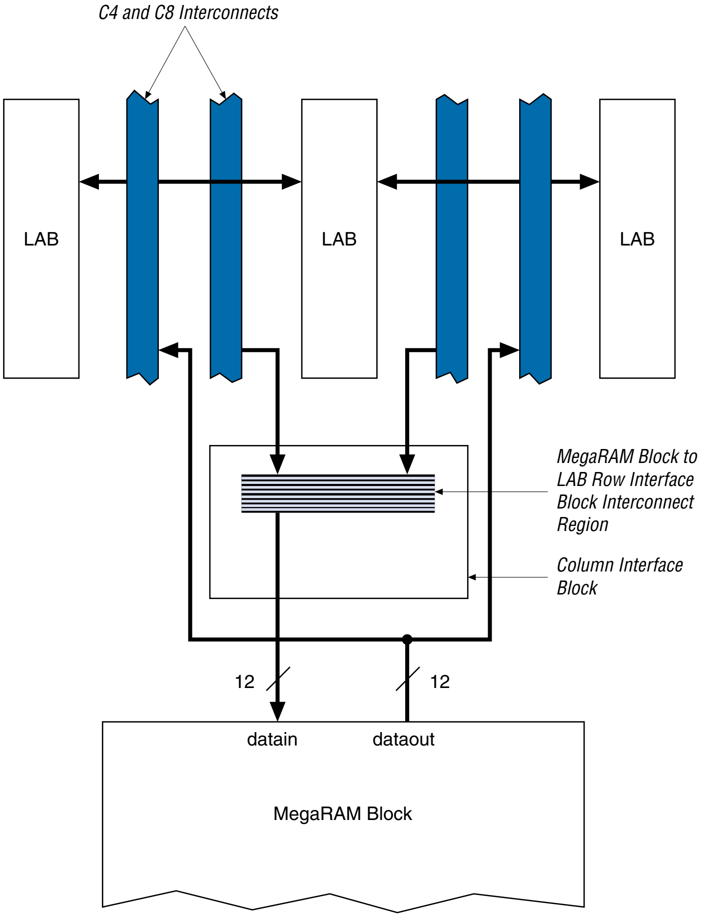


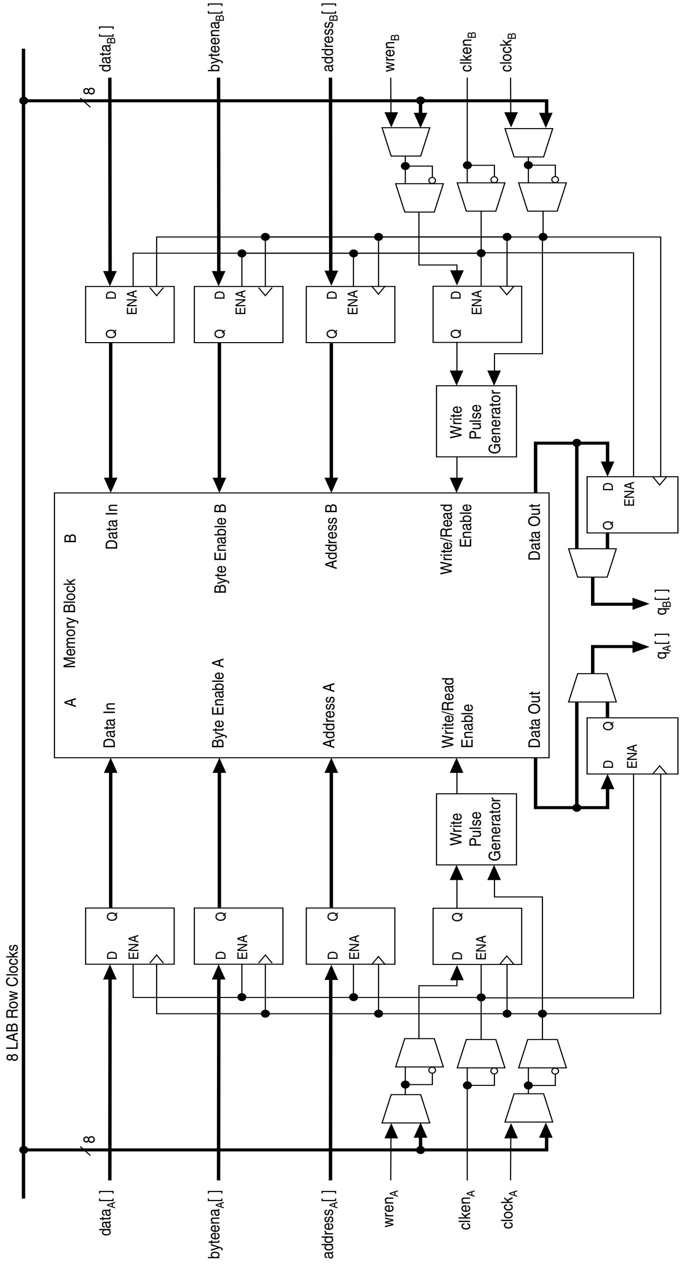
Table 16. MegaRAM Row & Column Interface Unit Signals

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 24 shows a TriMatrix memory block in independent clock mode.

Figure 24. Independent Clock Mode *Note (1)*



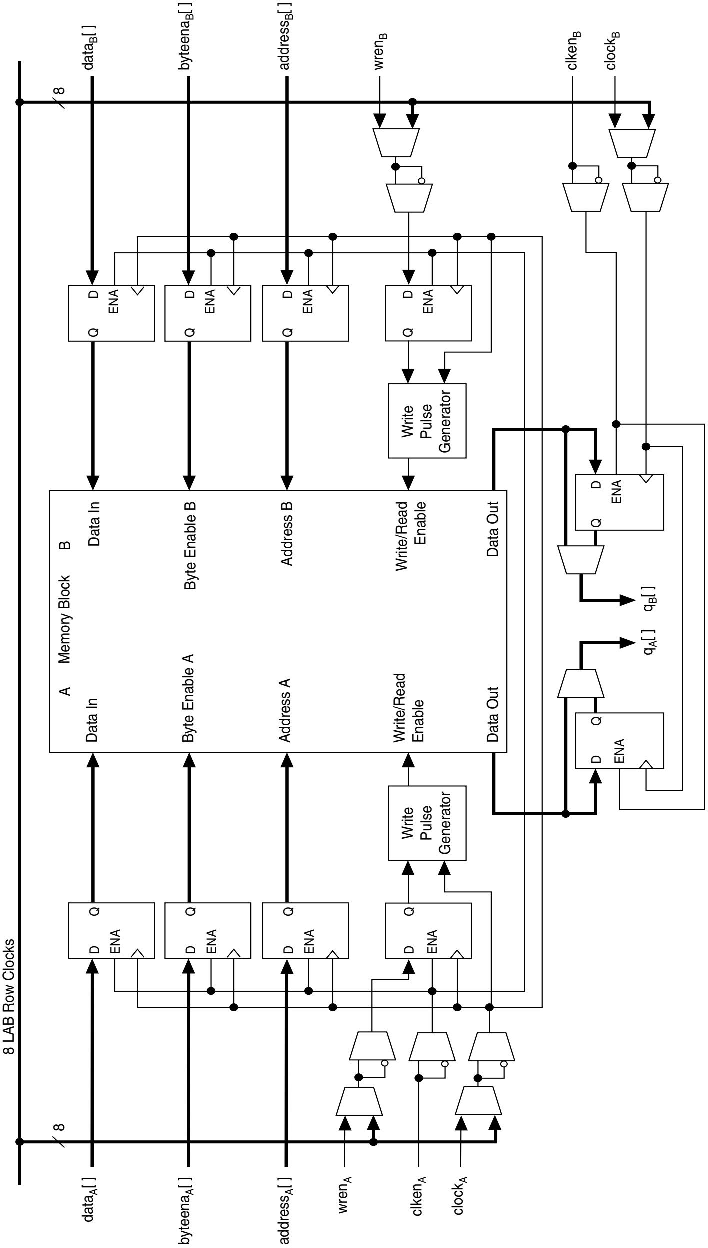
Note to Figure 24:

(1) All registers shown have asynchronous clear ports.

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 25 and 26 show the memory block in input/output clock mode.

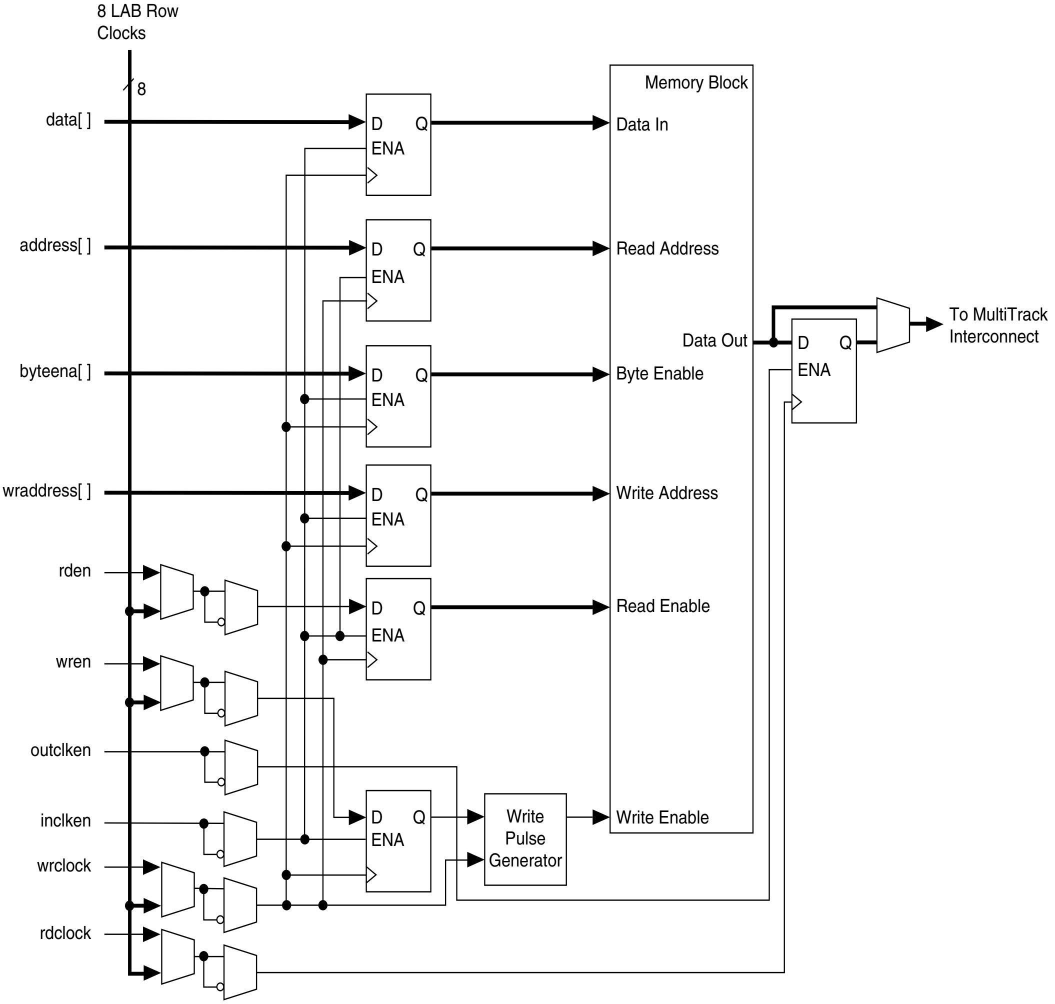
Figure 25. Input/Output Clock Mode in True Dual-Port Mode *Note (1)*



Note to Figure 25:

(1) All registers shown have asynchronous clear ports.

Figure 26. Input/Output Clock Mode in Simple Dual-Port Mode Note (1)



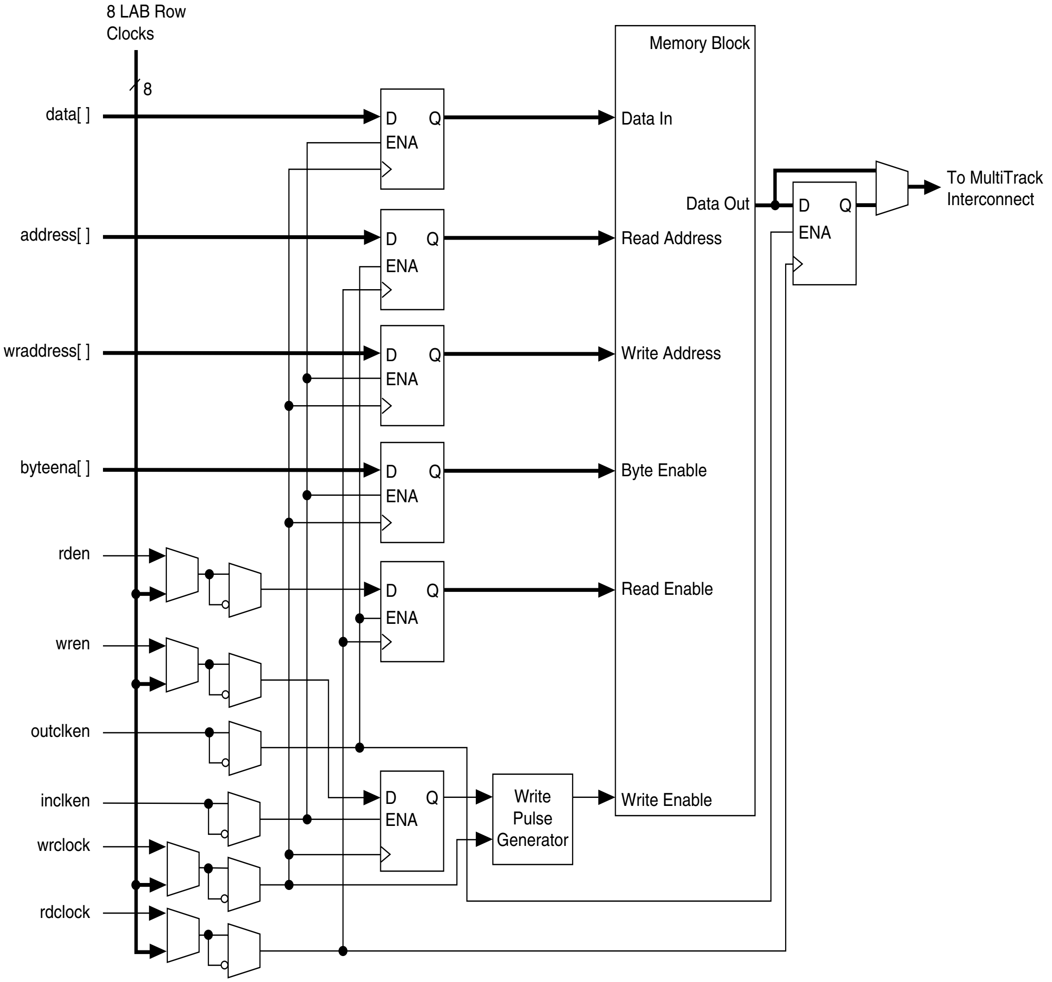
Note to Figure 26:

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

Read/Write Clock Mode

The memory blocks implement read/write clock mode for simple dual-port memory. The designer 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 27 shows a memory block in read/write clock mode.

Figure 27. Read/Write Clock Mode in Simple Dual-Port Mode Note (1)



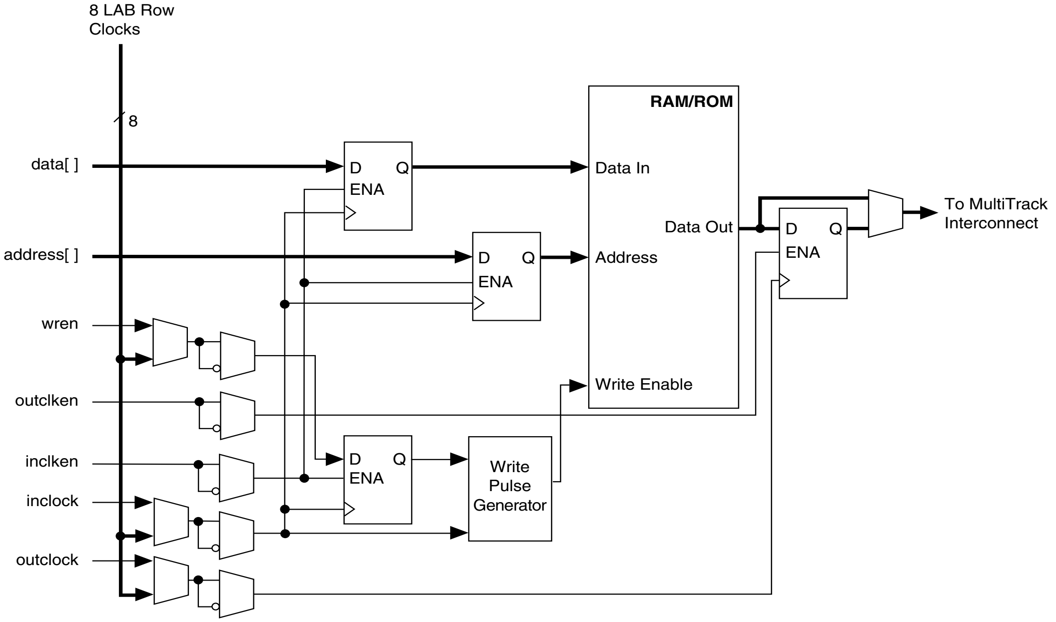
Note to Figure 27:

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

Single-Port Mode

The memory blocks also support single-port mode, used when simultaneous reads and writes are not required. See Figure 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 28. Single-Port Mode



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 17). Each DSP block can be configured to support:

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

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

Figure 29. DSP Blocks Arranged in Columns

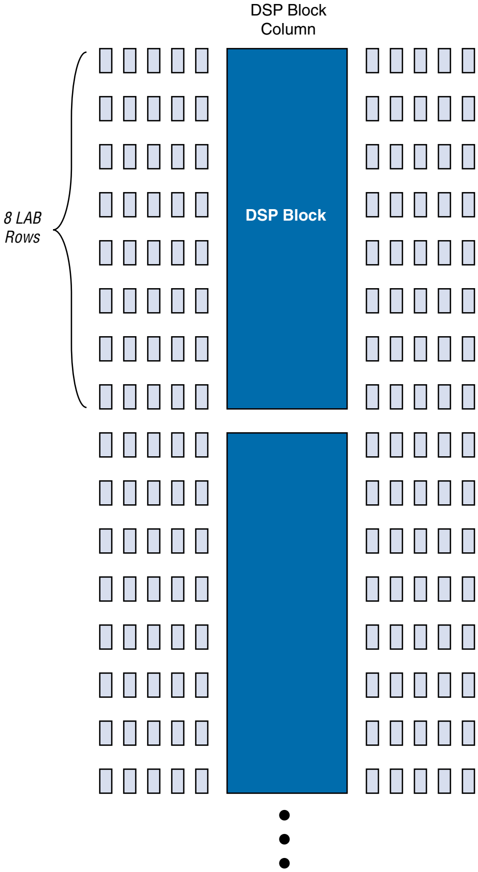


Table 17 shows the number of DSP blocks in each Stratix device.

Table 17. DSP Blocks in Stratix Devices Note (1)

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
EP1S120	28	224	112	28

Note to Table 17:

(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.

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

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

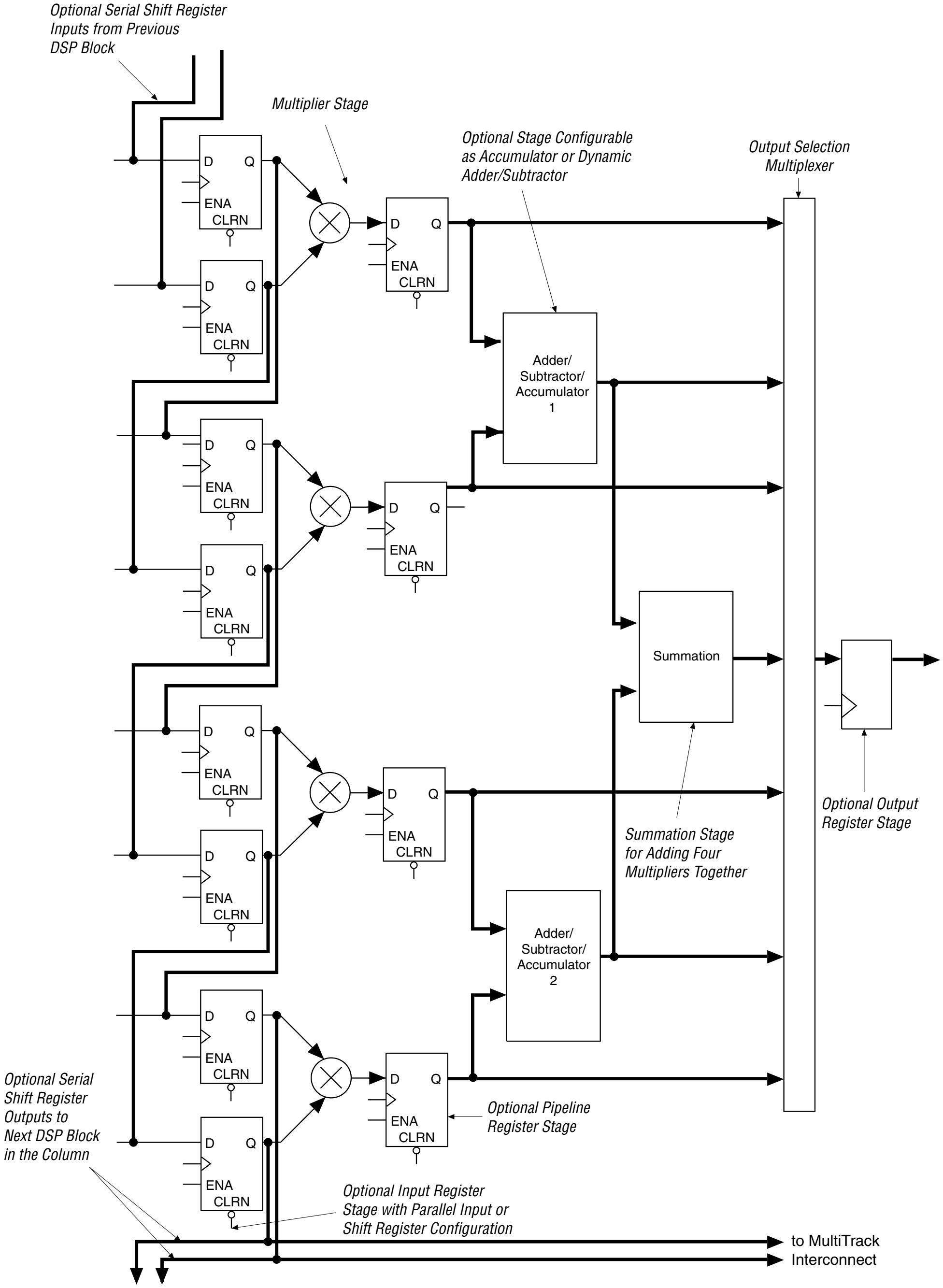
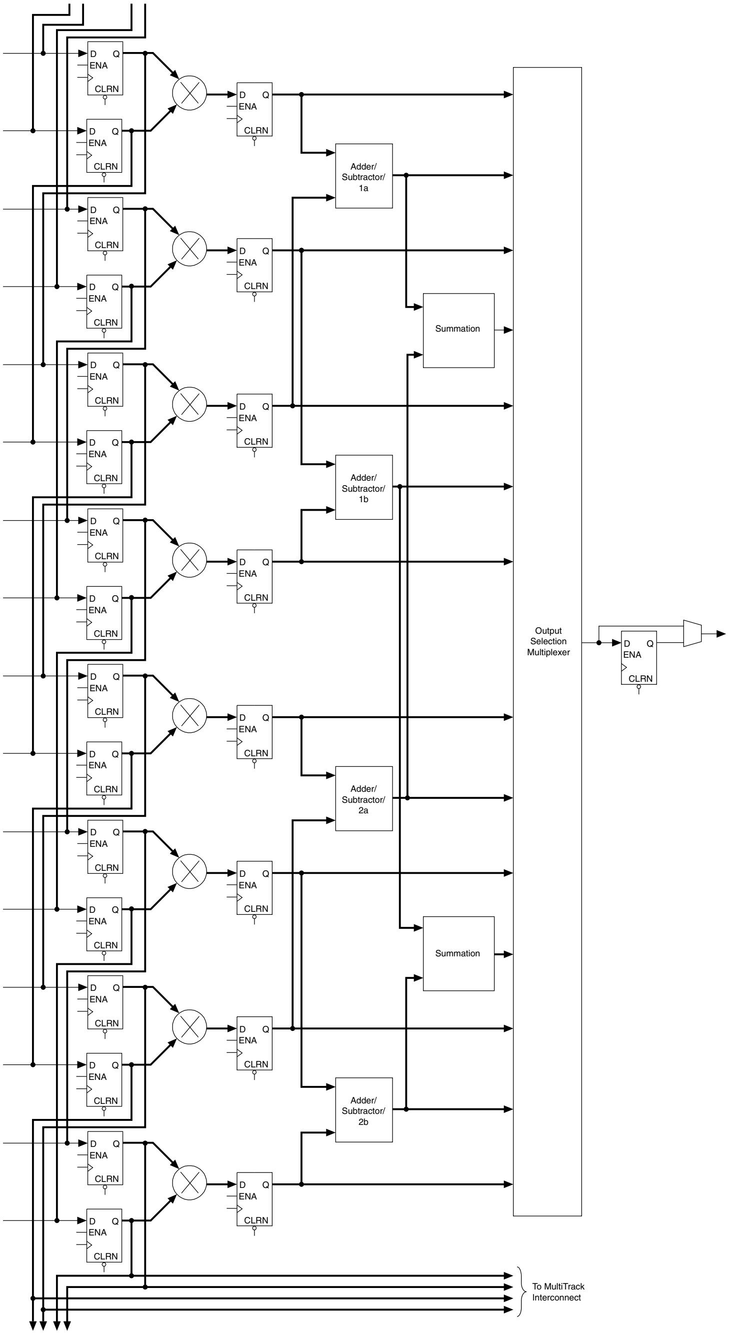


Figure 31. DSP Block Diagram for 9 × 9-Bit Configuration



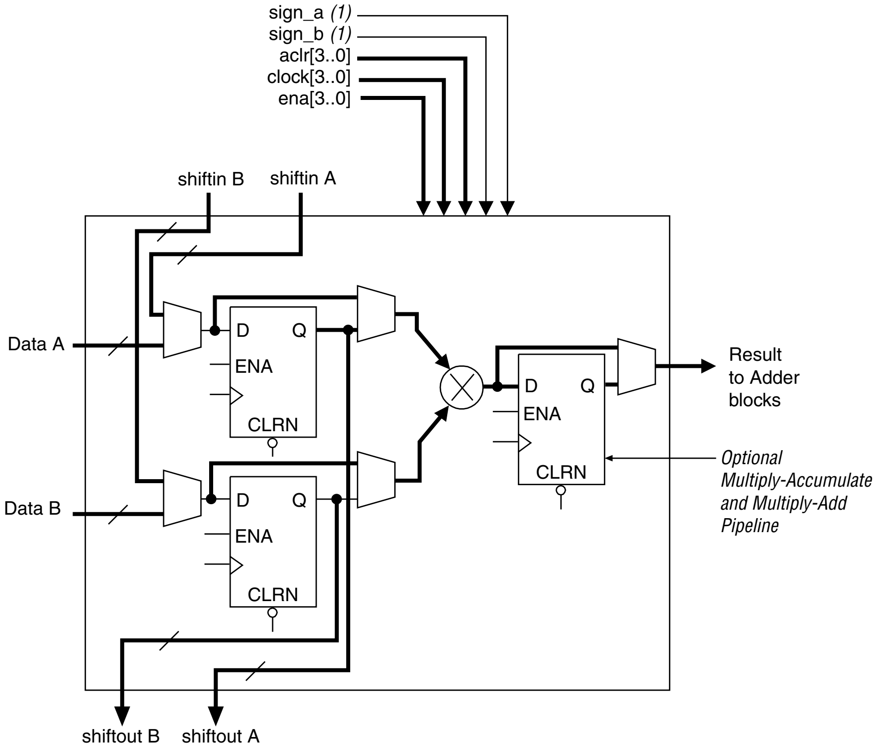
The DSP block consists of the following elements:

- 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 32.

Figure 32. Multiplier Sub-Block within Stratix DSP Block



Note to Figure 32:

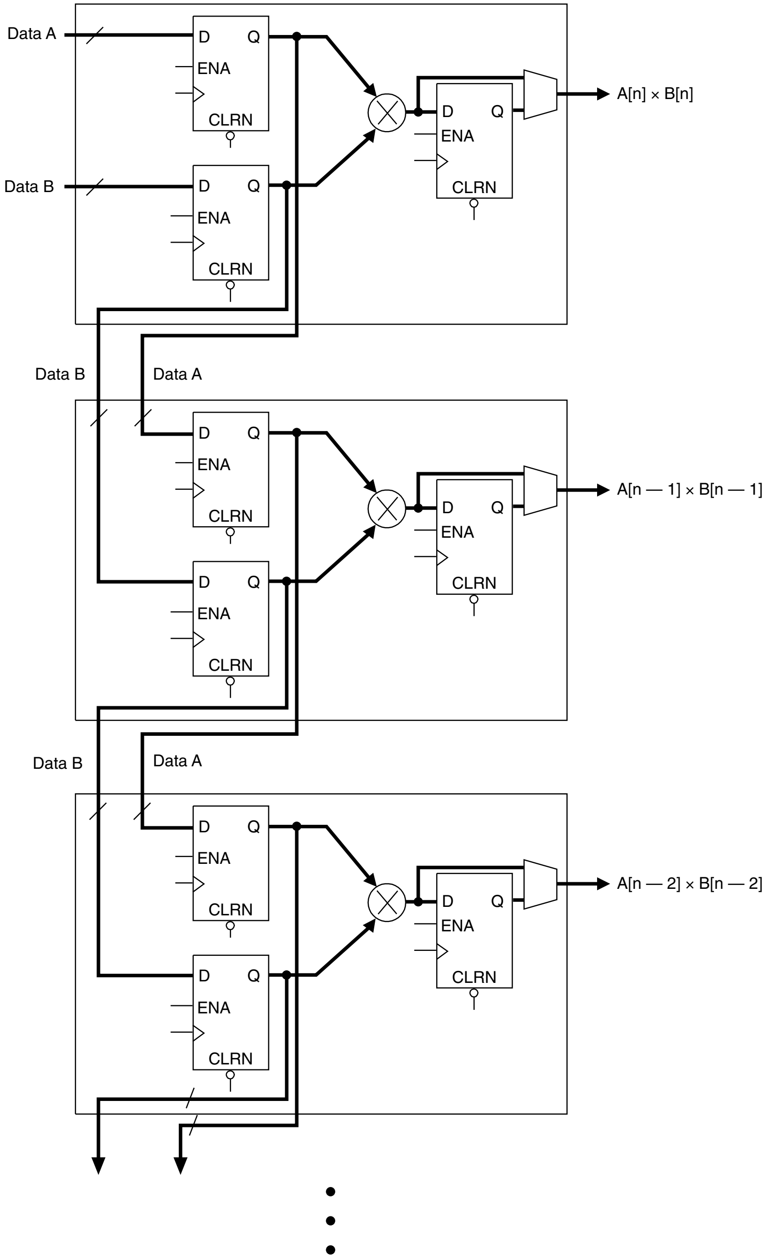
(1) These signals can be unregistered or registered once to match data path pipelines if required.

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. Designers 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. Designers 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.

Designers 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 33, to form a shift register chain. This chain can terminate in any block, i.e., designers can create any length of shift register chain up to 224 registers. The designer 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, the designer does not need to implement external shift registers in LAB LEs. The designer implements 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 33. Multiplier Sub-Blocks Using Input Shift Register Connections Note (1)



Note to Figure 33:

(1) Either Data A or Data B input can be set to parallel input for constant coefficient multiplication.

Table 18 shows the summary of input register modes for the DSP block.

Table 18. 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 19. 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 designers can register them to match the data path pipeline. The multipliers are full precision (i.e., 18 bits for the 18-bit multiply, 36-bits for the 36-bit multiply, etc.) regardless of whether `sign_a` or `sign_b` set the operands as signed or unsigned numbers.

Table 19. Multiplier Signed Representation

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

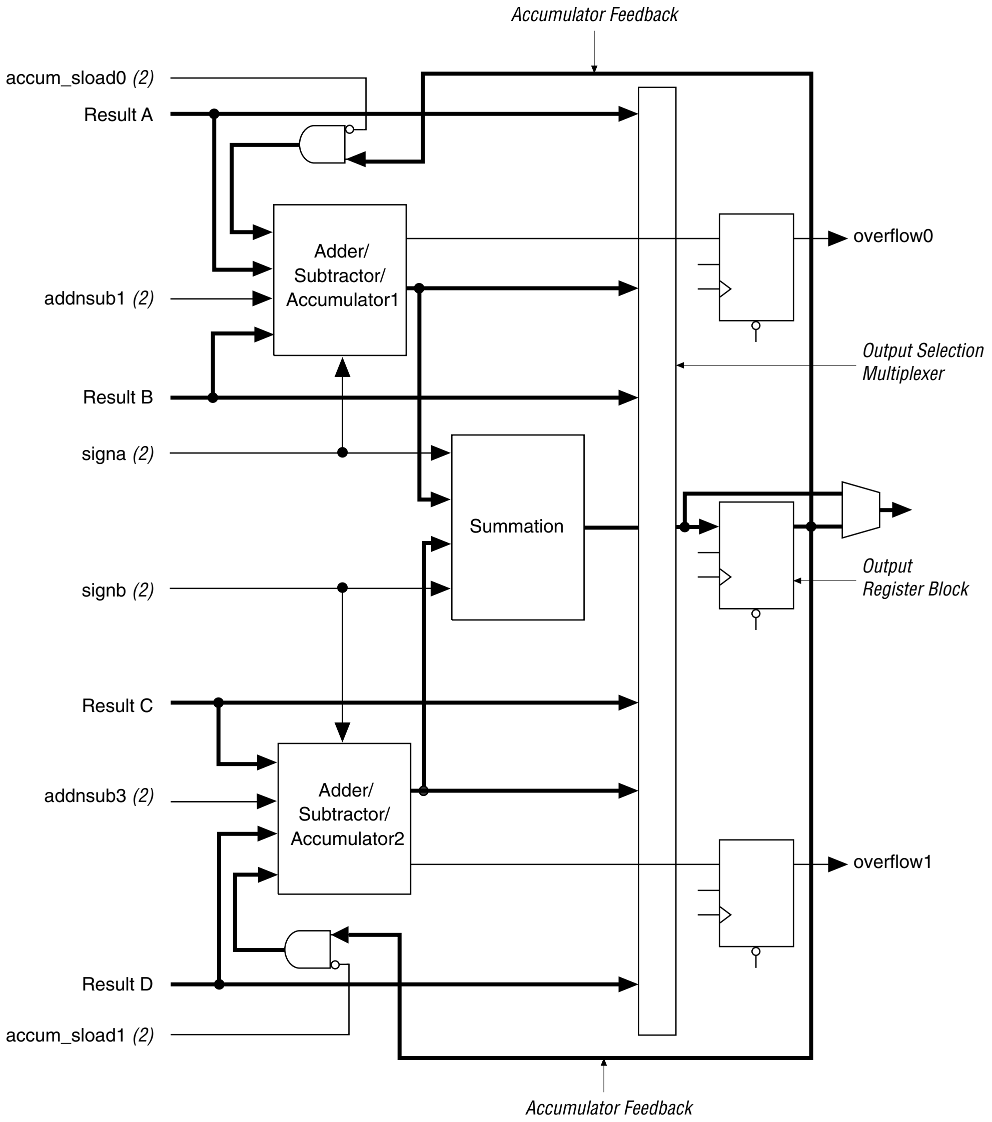
Pipeline/Post Multiply Register

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. The designer 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 34 shows the adder and output stages.

Figure 34. Adder/Output Blocks *Note (1)*



Notes to Figure 34:

(1) Adder/output block shown in Figure 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 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

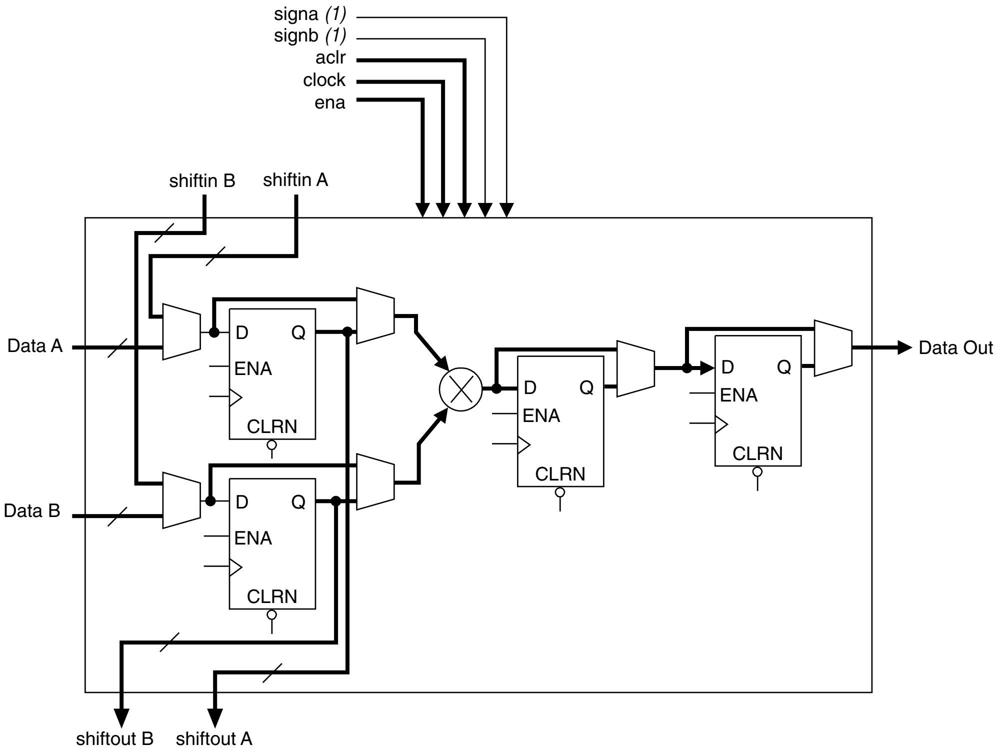
The adder, subtractor, and accumulate functions of a DSP block have four modes of operation:

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

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 35.

Figure 35. Simple Multiplier Mode

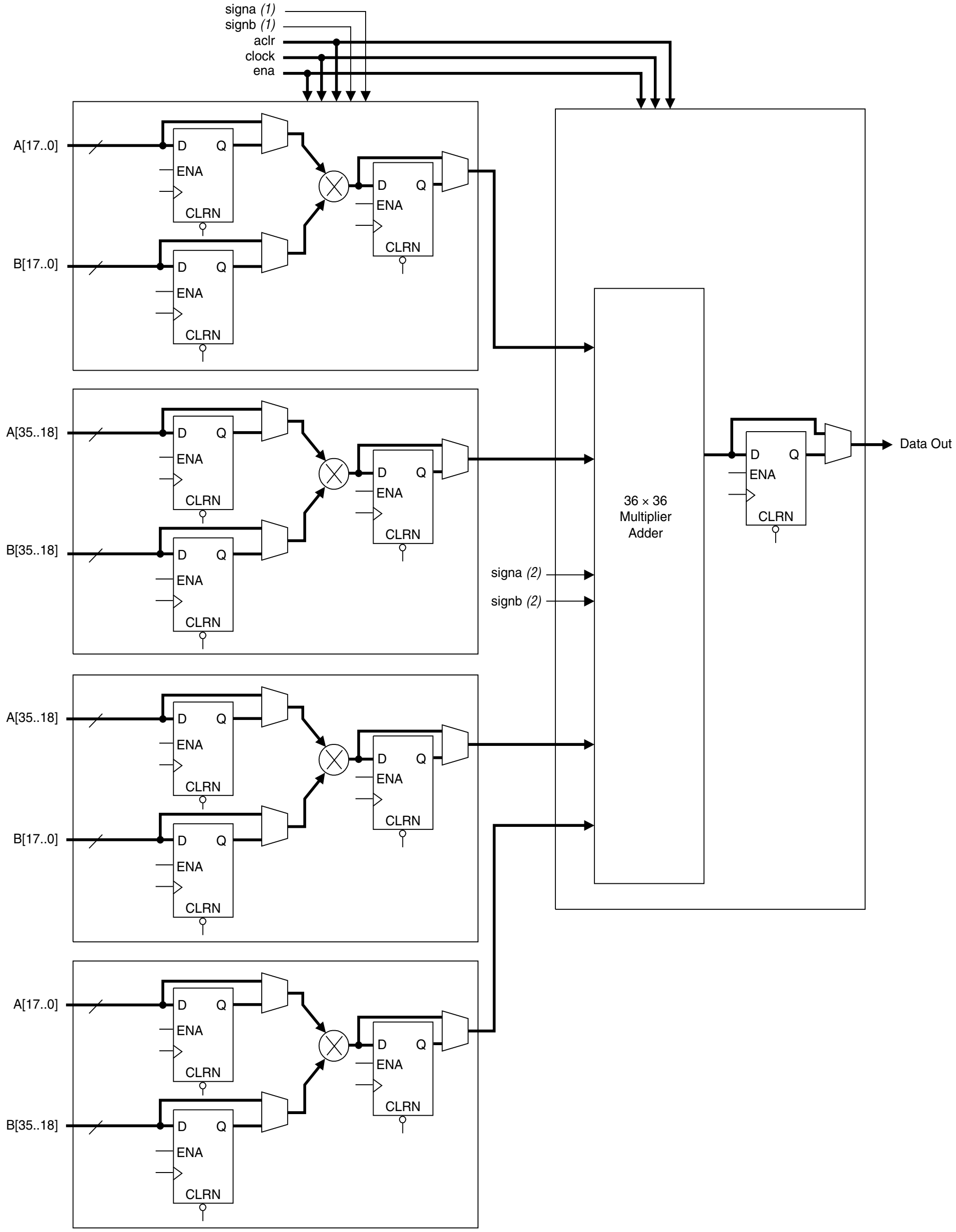


Note to Figure 35:

(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 36 shows the 36 × 36-bit multiply mode.

Figure 36. 36 × 36 Multiply Mode



Notes to Figure 36:

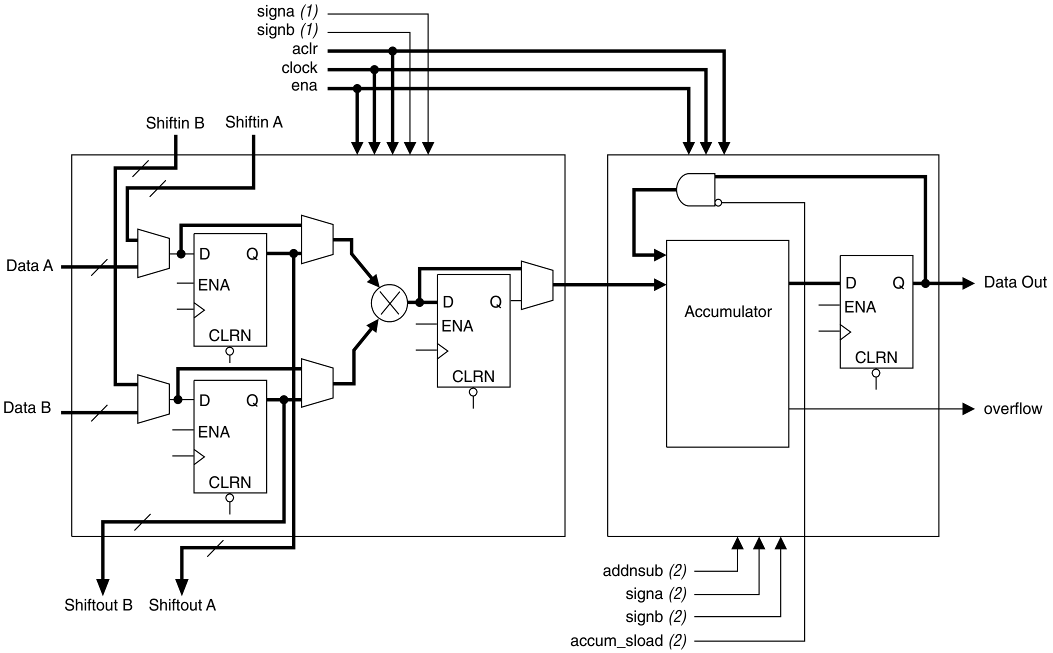
(1) These signals are not registered or registered once to match the pipeline.

(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 37), the DSP block drives multiplied results to the adder/subtractor/accumulator block configured as an accumulator. A designer 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, since 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 will indicate underflow condition.

Figure 37. Multiply-Accumulate Mode



Notes to Figure 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

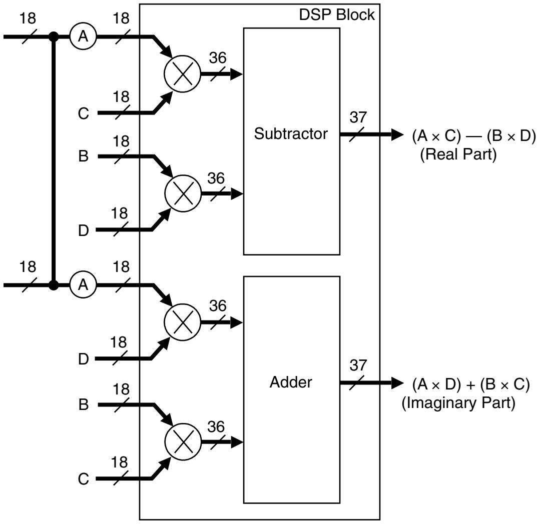
The two-multipliers adder mode uses the adder/subtractor/accumulator block to add or subtract the outputs of the multiplier block, which is useful for applications such as FFT functions and complex FIR filters. A single DSP block can implement two sums or differences from two 18 × 18-bit multipliers each or four sums or differences from two 9 × 9-bit multipliers each.

Designers 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 $[(a \times c) - (b \times d)]$ using one subtractor and the imaginary part $[(a \times d) + (b \times c)]$ 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 38 shows an 18-bit two-multipliers adder.

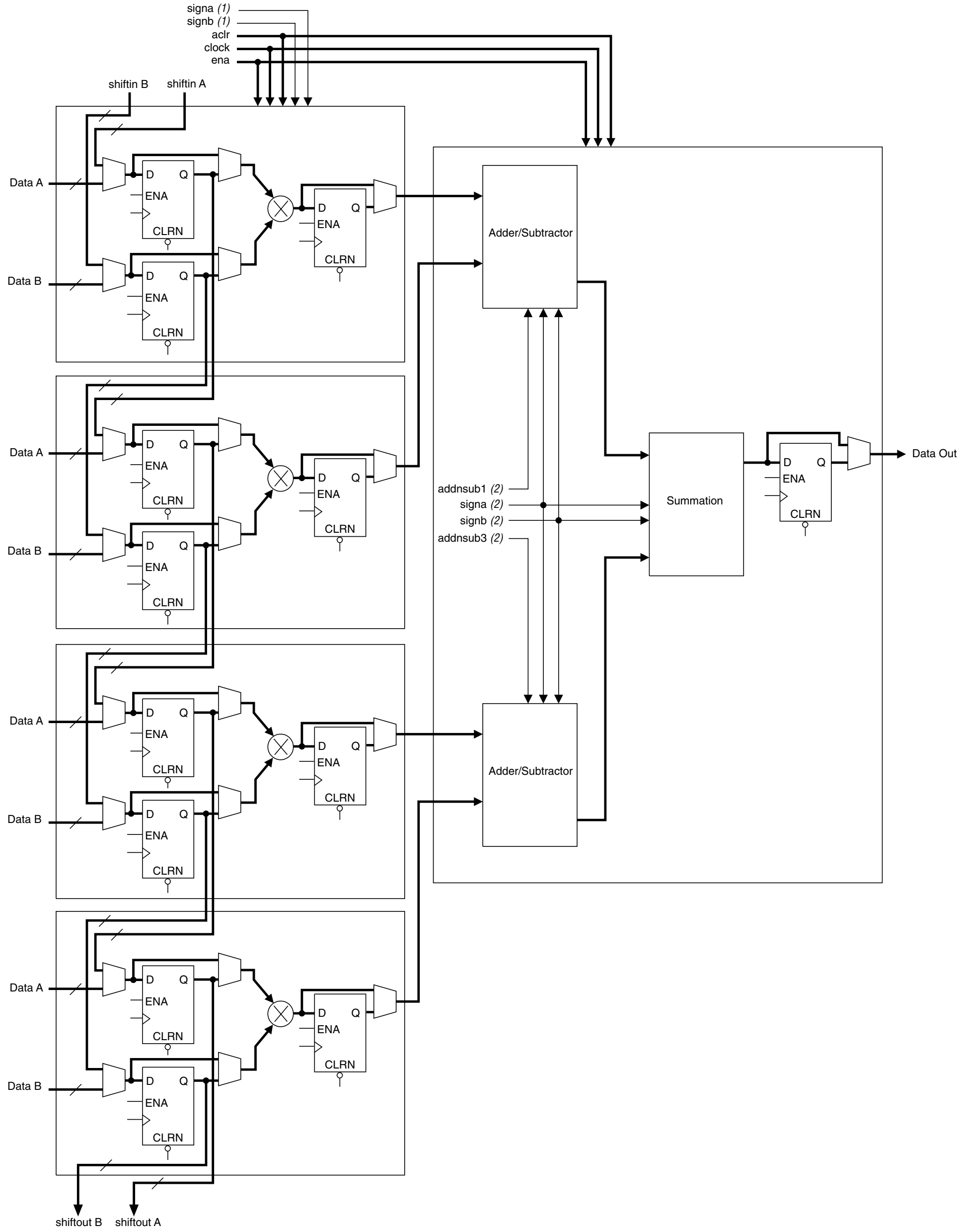
Figure 38. Two-Multipliers Adder Mode Implementing Complex Multiply



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 39 shows the four multipliers adder mode.

Figure 39. Four-Multipliers Adder Mode



Notes to Figure 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 20 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 20. Multiplier Size & Configurations per DSP block

DSP Block Mode	9 × 9	18 × 18	36 × 36
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	–

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. The designer 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 40 and 41 show the DSP block interfaces to LAB rows.

Figure 40. DSP Block Interconnect Interface

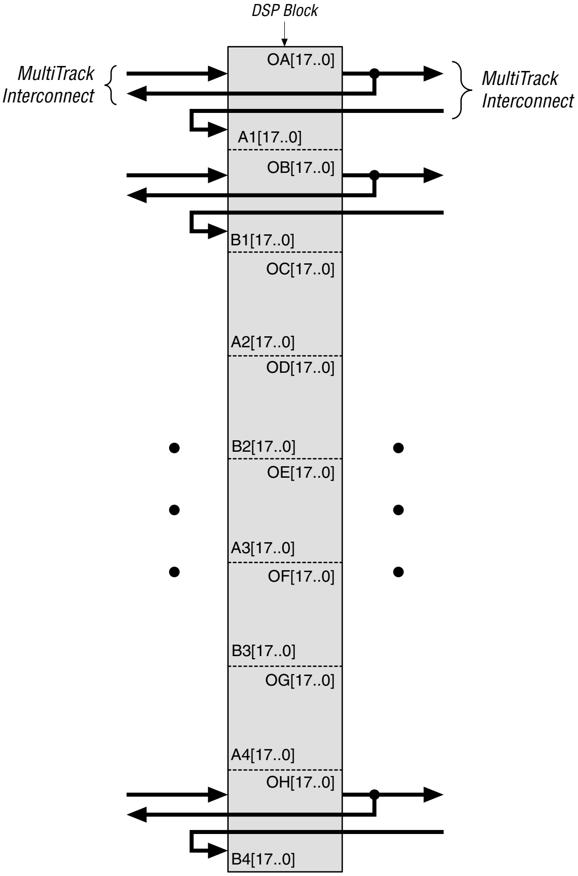
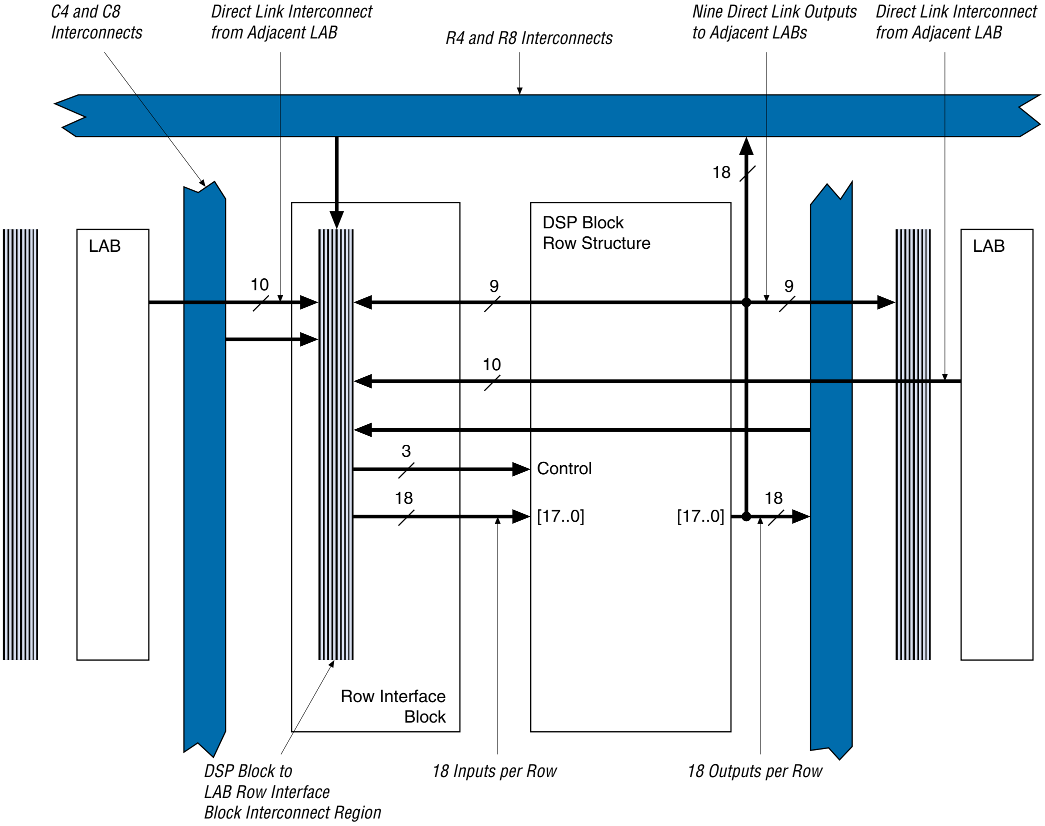


Figure 41. DSP Block Interface to Interconnect



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 21.

Table 21. 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

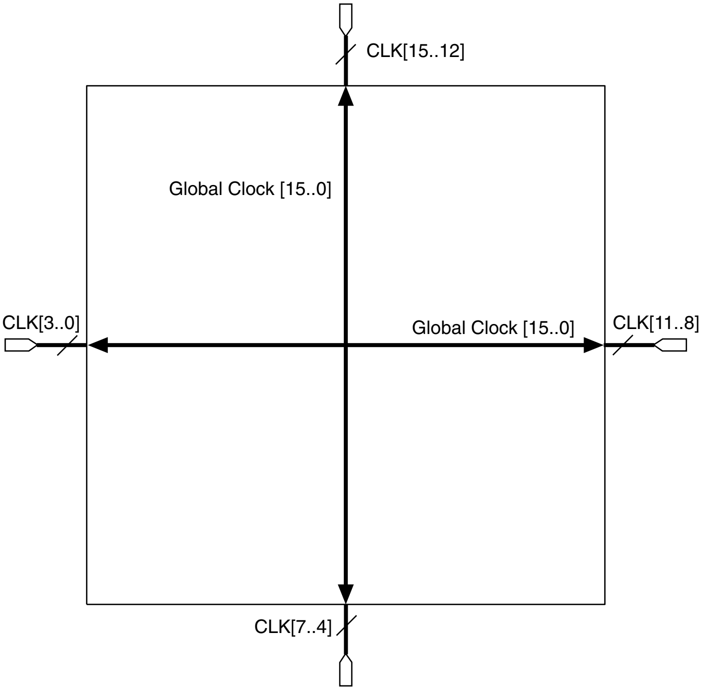
Stratix devices provide 16 dedicated global clock networks, 16 regional clock networks (four per device quadrant), and 8 dedicated fast regional clock networks. These clocks are organized into a hierarchical clock structure that allows for up to 22 clocks per device region with low skew and delay. This 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 Figures 42 and 43. 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 42 shows the 16 dedicated `CLK` pins driving global clock networks.

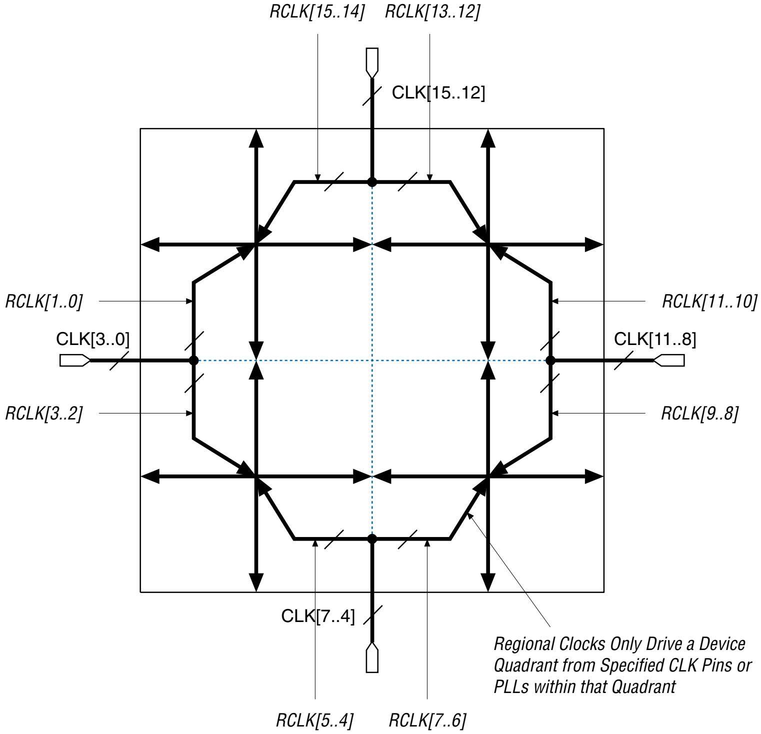
Figure 42. Global Clocking



Regional Clock Network

There are four regional clock networks `RCLK[3..0]` within each quadrant of the Stratix device that are driven by the same dedicated `CLK[15..0]` input pins or from PLL outputs. 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. The `CLK` clock pins symmetrically drive the `RCLK` networks within a particular quadrant, as shown in Figure 43.

Figure 43. Regional Clocks



Regional Clocks Only Drive a Device Quadrant from Specified CLK Pins or PLLs within that Quadrant

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 44). In EP1S30 and larger devices, there are two fast regional clock networks within each half-quadrant (see Figure 45). The `FCLK[1..0]` clocks can also be used for high fanout control signals, such as asynchronous clears, presets, clock enables, or protocol control signals such as `TRDY` and `IRDY` for PCI. Dual-purpose `FCLK` pins drive the fast clock networks. All devices have eight `FCLK` pins to drive fast regional clock networks. I/O interconnects can also drive clock or control signals onto any fast regional clock networks.

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

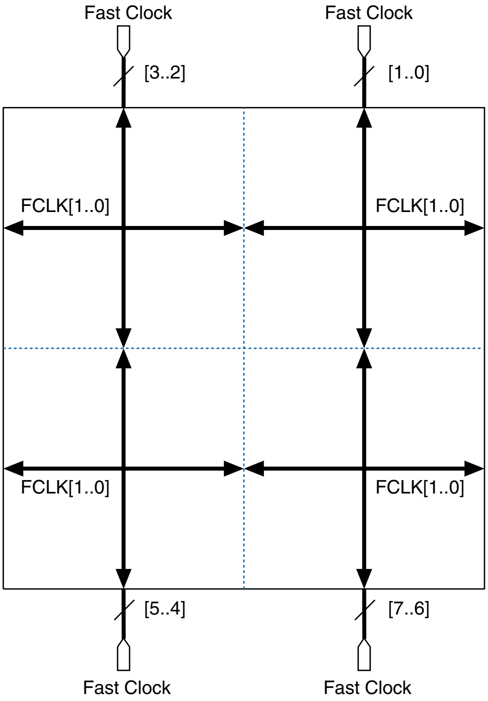
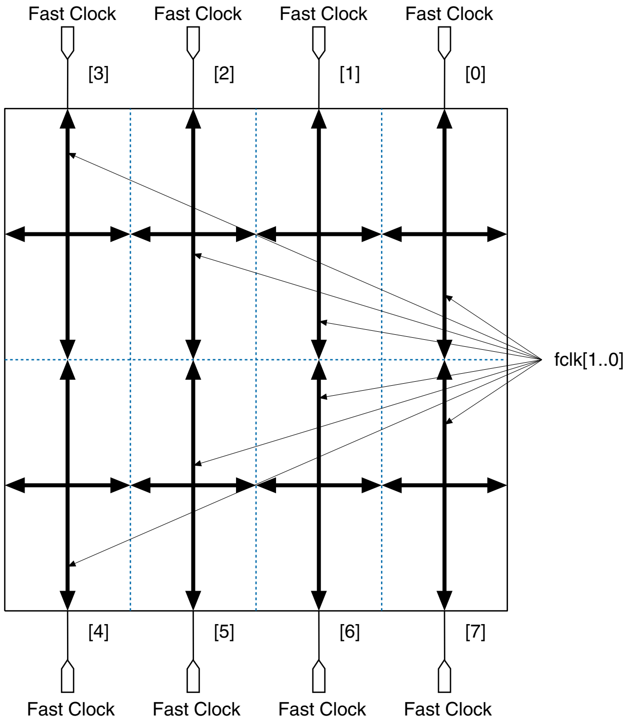


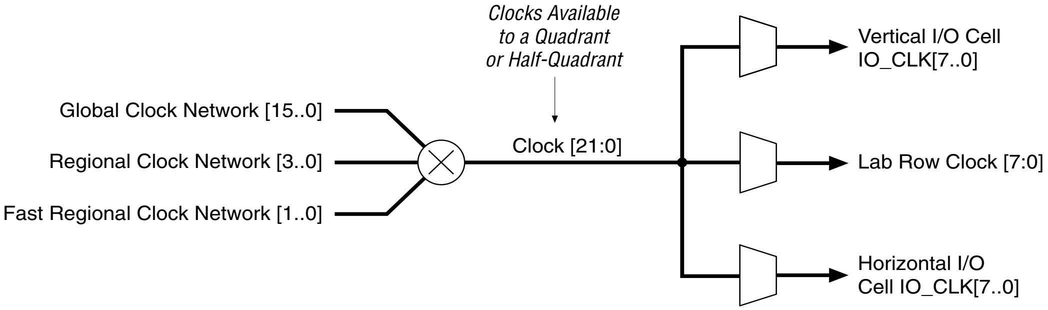
Figure 45. EP1S120 to EP1S30 Device Fast Regional Clock Pin Connections to Fast Regional Clocks



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 46.

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

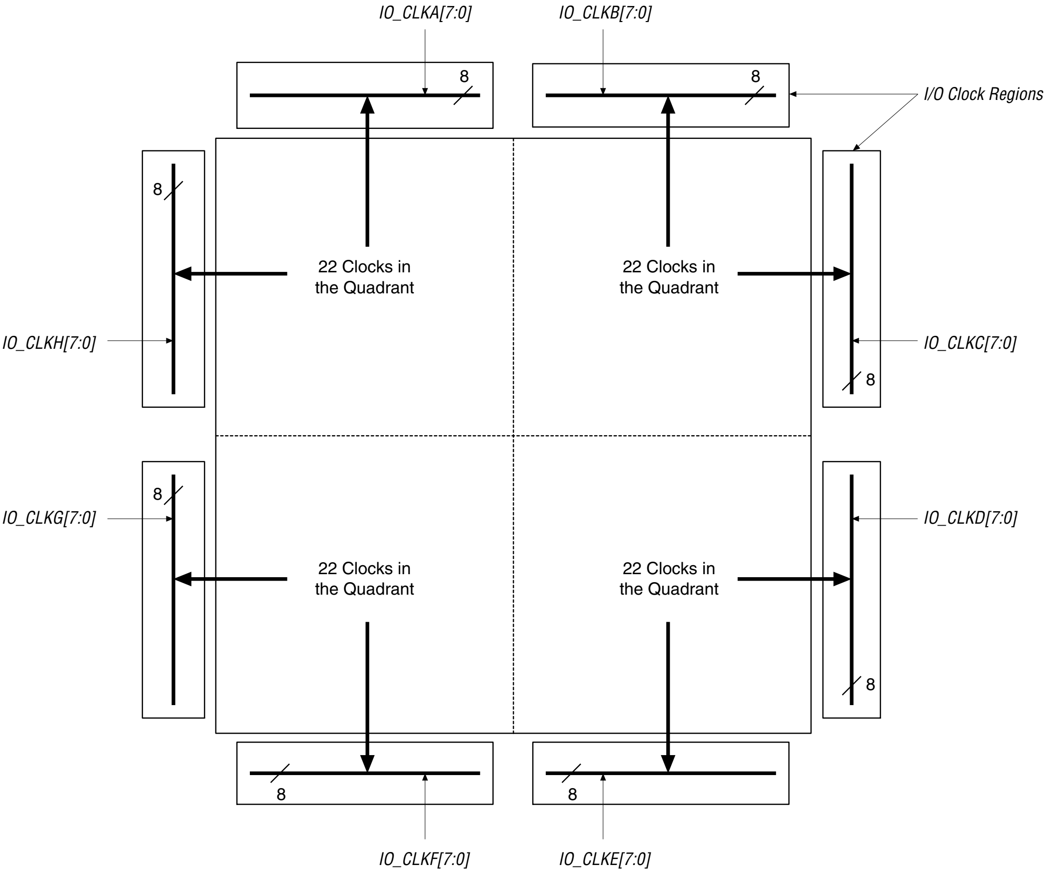
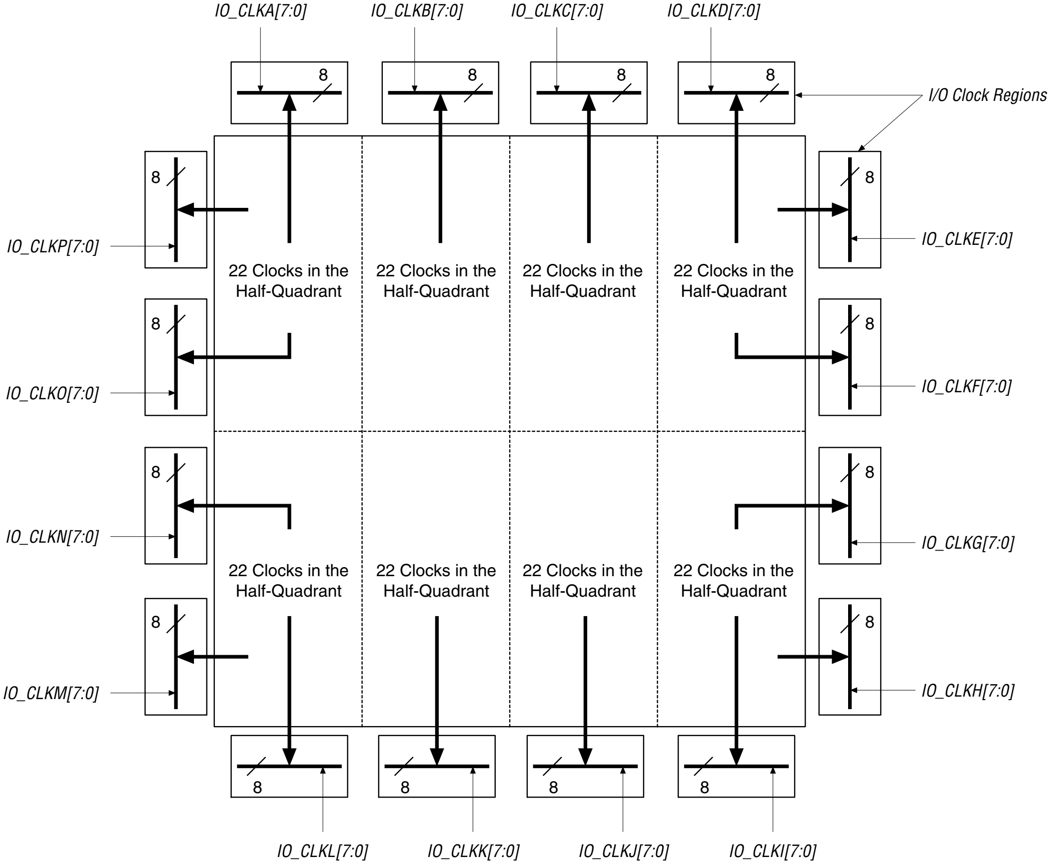


Figure 48. EP1S30, EP1S40, EP1S60, EP1S80 & EP1S120 Device I/O Clock Groups



Designers 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 designers with complete control of their 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 22 shows which PLLs are available for each Stratix device and their type. Table 23 shows the enhanced PLL and fast PLL features in Stratix devices.

Table 22. 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	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
EP1S40	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP1S60	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP1S80	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP1S120	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notes to Table 22:

(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.

Feature	Enhanced PLL	Fast PLL
Clock multiplication and division	*m*/(*n* × post-scale counter) *(1)*	*m*/(post-scale counter) *(2)*
Phase shift	Down to 160-ps increments *(3)*, *(4)*	Down to 150-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	4 *(8)*	

Table 23. Stratix PLL Features

Notes to Table 23:

(1) For enhanced PLLs, *m* and *n* are counters ranging from 1 to 512.
(2) For fast PLLs, *m* and post-scale counters range from 1 to 16.
(3) The smallest phase shift is determined by the 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 with eight single-ended or four differential outputs each. Two additional enhanced PLLs in EP1S120, EP1S80, EP1S60, and EP1S40 devices each have one single-ended output.
(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 49 shows a top-level diagram of the Stratix device and PLL floorplan.

Figure 49. PLL Locations

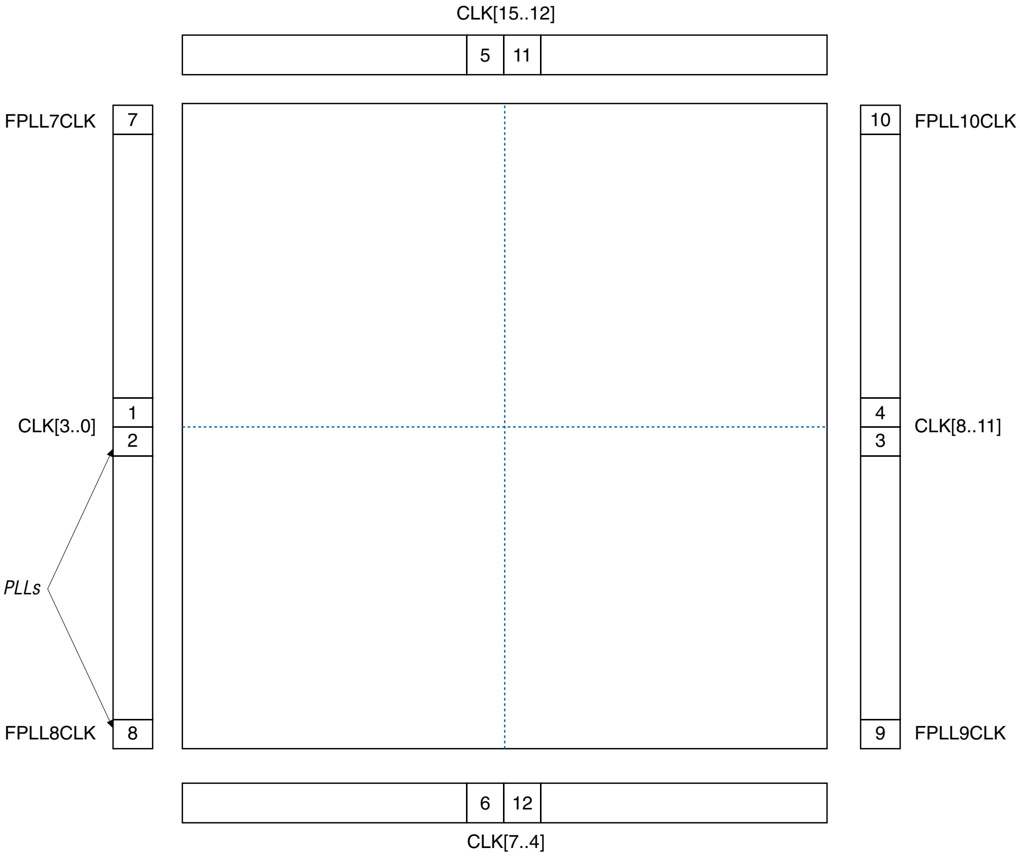
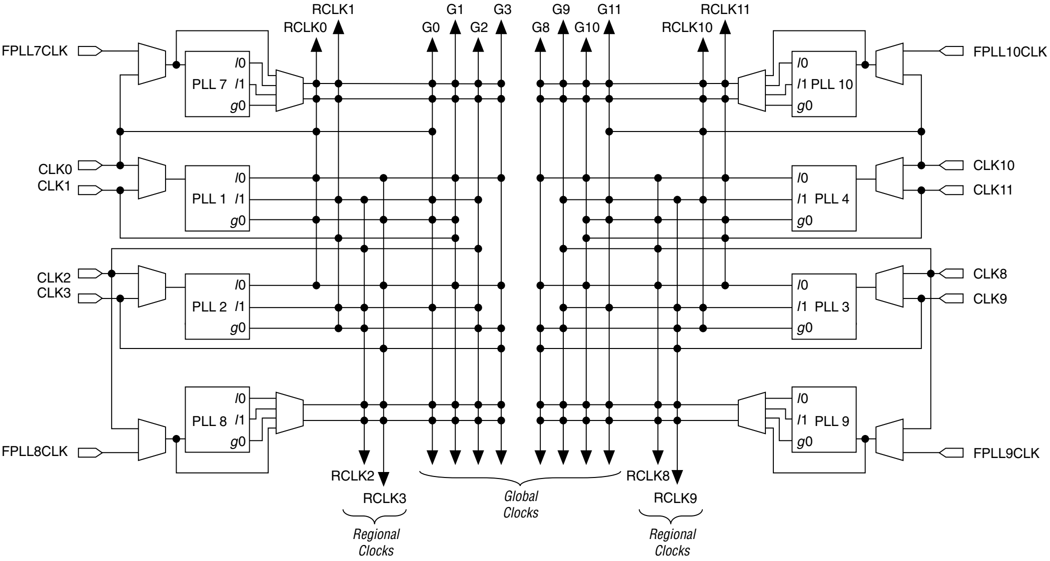


Figure 50 shows the global and regional clocking from the PLL outputs and the CLK pins.

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

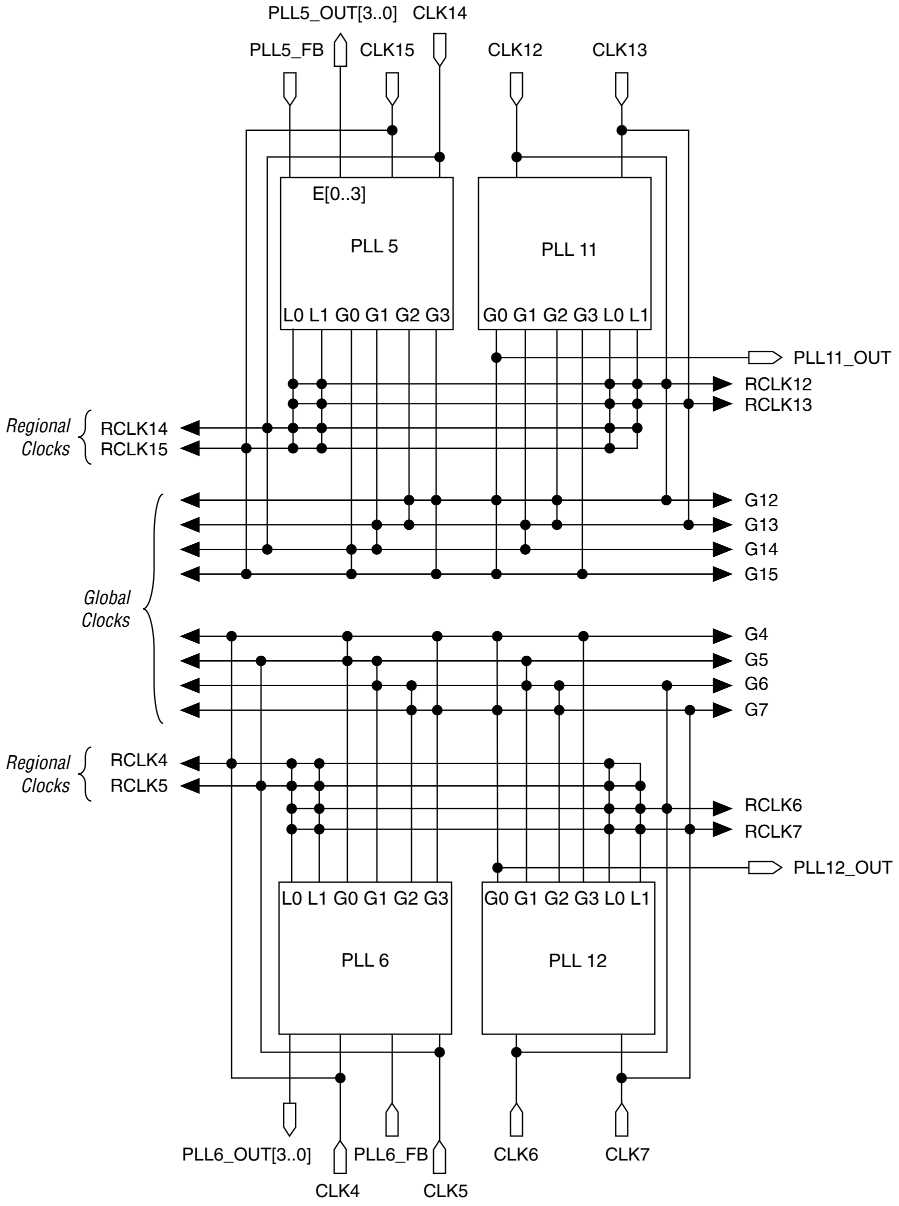


Note to Figure 50:

(1) PLLs 1 to 4 and 7 to 10 are fast PLLs. PLLs 5, 6, 11, and 12 are enhanced PLLs. PLLs 7 through 10 do not drive global clocks.

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

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



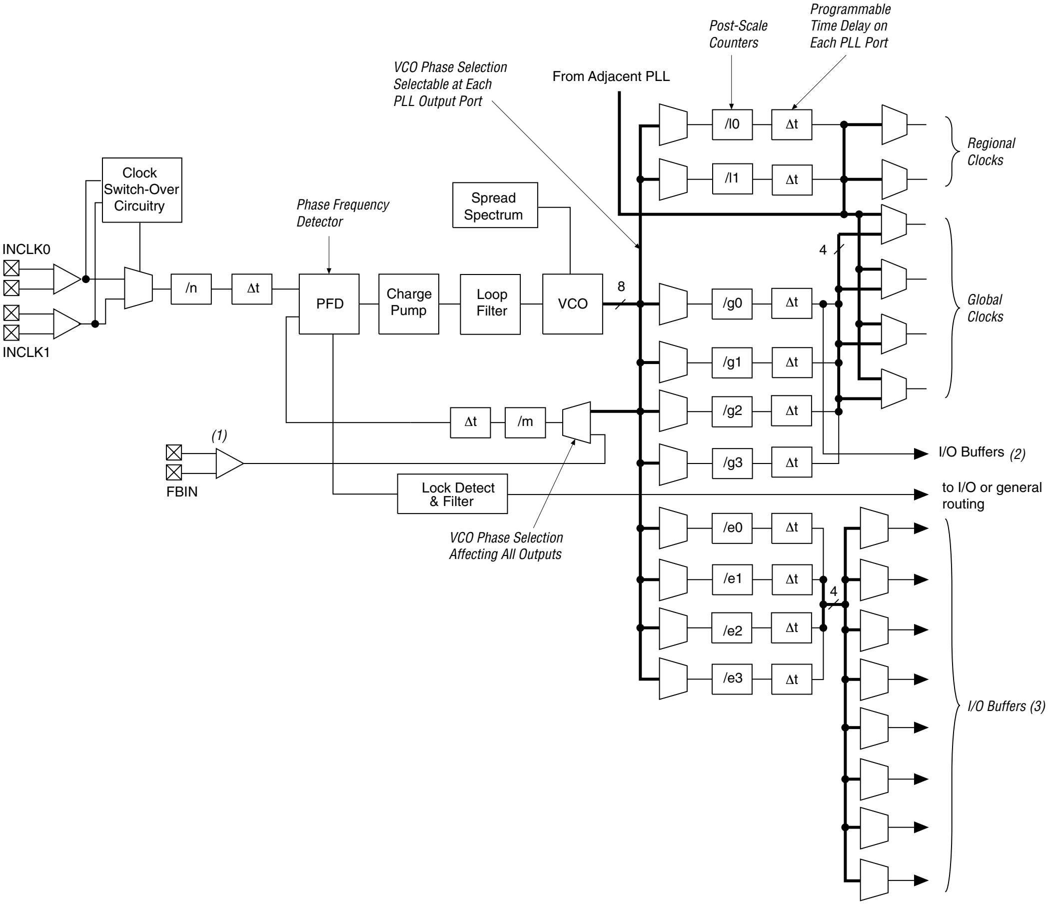
Note to Figure 51:

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

Enhanced PLLs

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

Figure 52. Stratix Enhanced PLL



Notes to Figure 52:

(1) External feedback is available in PLLs 5 and 6.
(2) This 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.

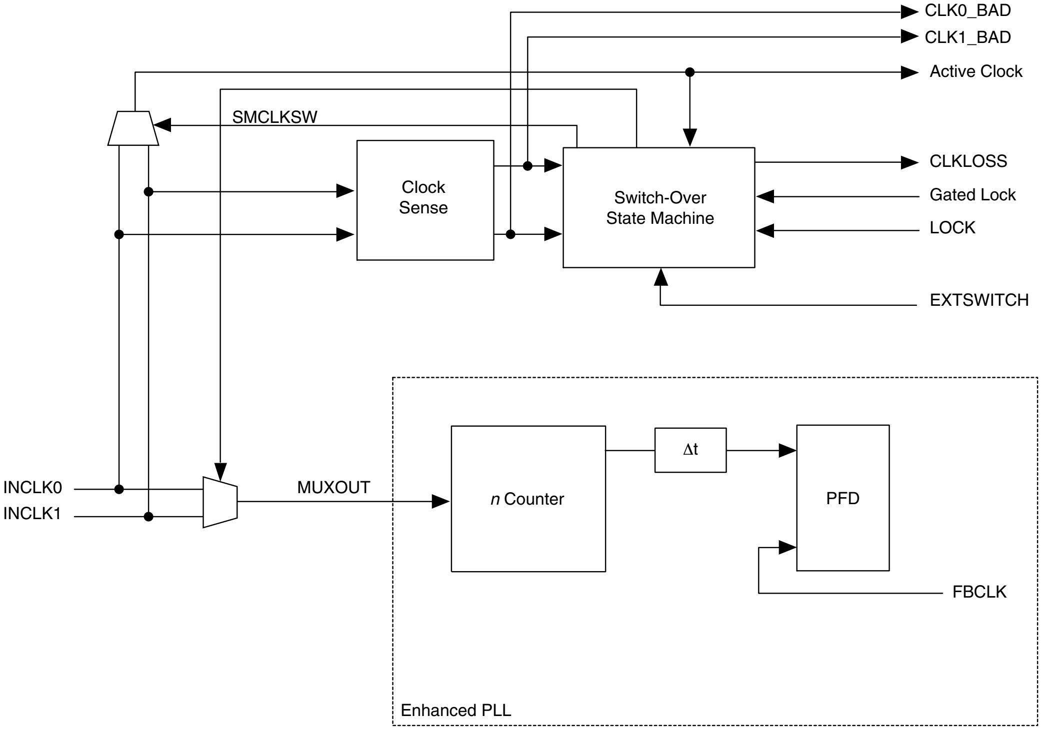
Clock Multiplication & Division

Each Stratix device enhanced PLL provides clock synthesis for PLL output ports using $m/(n \times \text{post-scale counter})$ 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 divider, n, and one multiply divider, m, per PLL, with a range of 1 to 512 on each. There are two post-scale dividers (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 512. 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 53 shows a block diagram of the switchover circuit. The switchover circuit is configurable, so the designer 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 53. Clock Switchover Circuitry



Note to Figure 53:
(1) PFD: phase frequency detector.

There are at least three possible ways to use the clock switch-over feature.

- Designers can use the switch-over 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 53. In this case, the secondary clock becomes the reference clock for the PLL.

- Designers can use the `extswitch` 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 `extclk0` is 66 MHz and `extclk1` is 100 MHz, the designer 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.

- If the PLL loses lock for some reason, designers can set the gated lock to control switchover. The gated lock signal goes low to force the switch-over state machine to switch to the secondary clock. If an external PLL is driving the Stratix device PLL, excessive jitter on the clock input could cause the PLL to lose lock. Since the switch-over circuit still senses clock edges, it might not sense a switch condition. In this case, the designer can control switchover using the gated version of the locked signal based on the loss of the primary clock.

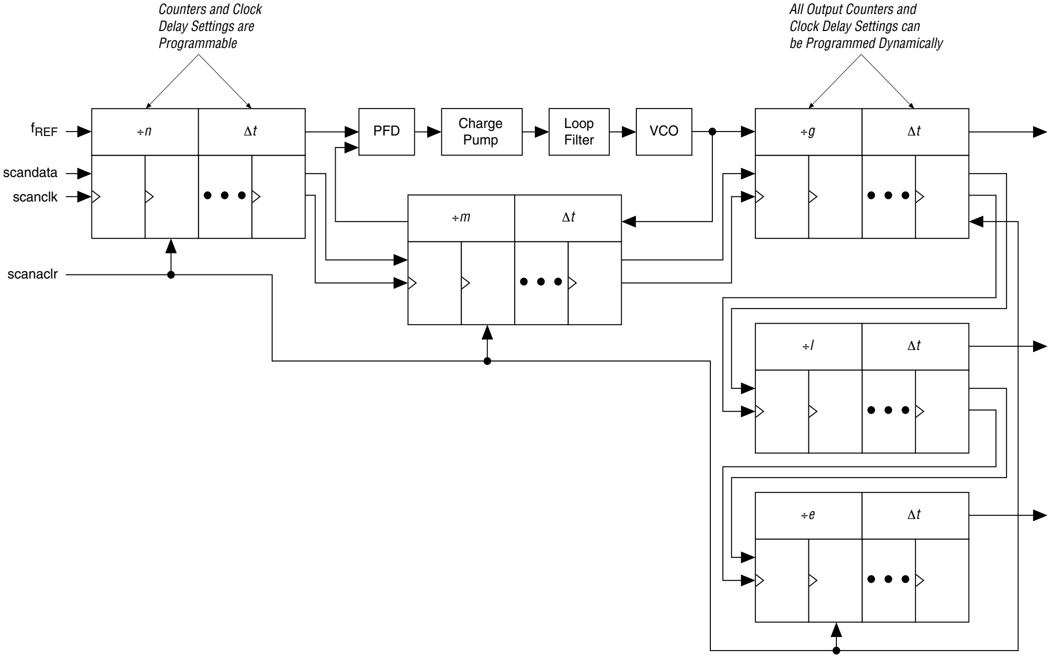
During switch over, the PLL VCO continues to run and will slow down, generating frequency drift on the PLL outputs. The clock switchover transitions without any glitches. After the switch, there is a finite resynchronization period to lock onto new clock as the VCO ramps up. The exact amount of time 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 preliminary specification for the maximum time to relock is 100 μs.

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. The designer 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 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 25 MHz. The charge pump, loop filter components, and phase shifting using VCO phase taps cannot be dynamically adjusted.

Figure 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.

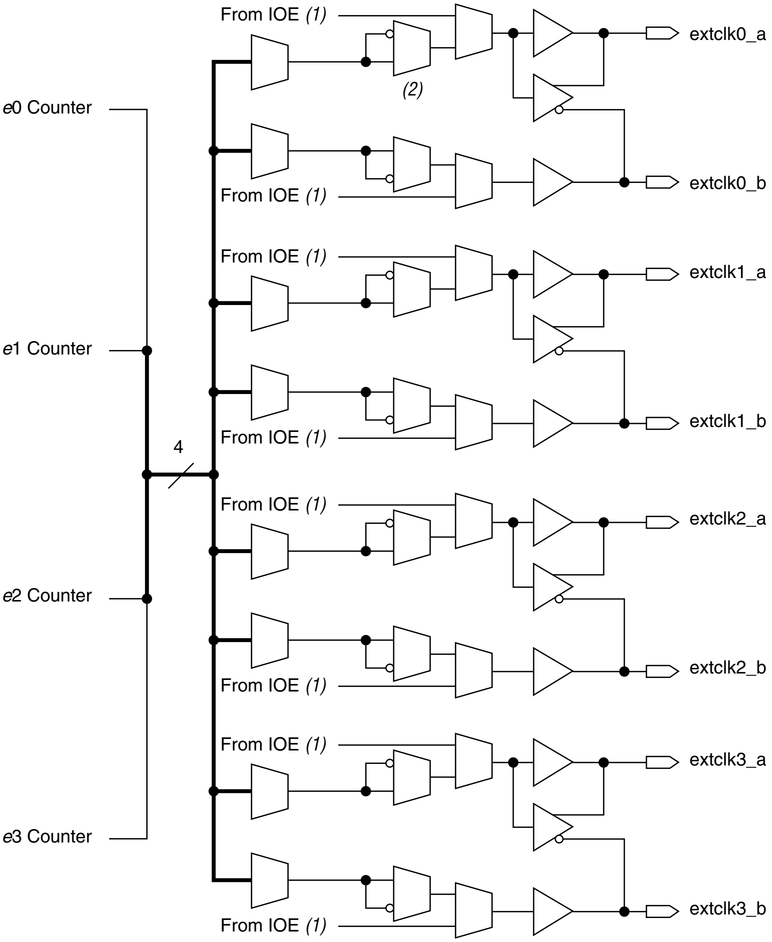
Programmable Bandwidth

The designer has 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. Designers 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). See Figure 55.

Figure 55. External Clock Outputs for PLLs 5 & 6



Notes to Figure 55:

(1) Each external clock output pin can be used 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°.

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.

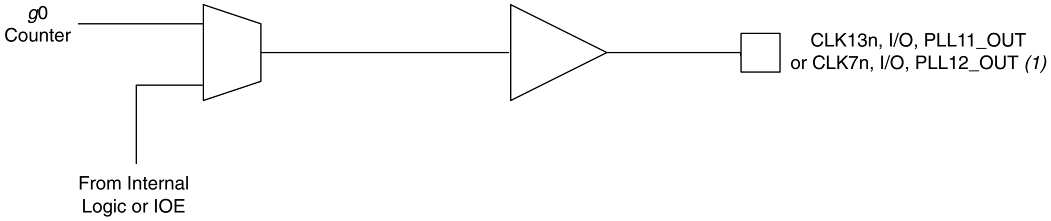
For PLLs 5 and 6, each pin of a single-ended output pair can either be in phase or 180° out of phase. The clock output pin pairs support the same I/O standards as standard output pins (in the top and bottom banks) as well as LVDS, LVPECL, PCML, HyperTransport technology, differential HSTL, and differential SSTL. Table 24 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. Designers can also use the external clock output pins as user output pins if external enhanced PLL clocking is not needed.

Table 24. I/O Standards Supported for Enhanced PLL Pins

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	✓	✓		✓
LVPECL	✓	✓		✓
PCML	✓	✓		✓
LVDS	✓	✓		✓
HyperTransport technology	✓	✓		✓
Differential HSTL	✓			✓
Differential SSTL				✓
3.3-V GTL	✓	✓		✓
3.3-V GTL+	✓	✓		✓
1.5V HSTL class I	✓	✓		✓
1.5V 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 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 56. External Clock Outputs for Enhanced PLLs 11 & 12



Note to Figure 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.
- External feedback: The external feedback input pin, `FBIN`, is phase-aligned with the clock input, `CLK`, pin. Aligning these clocks allows the designer 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.
- 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. The designer defines 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. For phase shifting, designers can specify a phase shift (in degrees or time units) for each PLL clock output port or for all outputs together in one shift. Phase-shifting values in time units are allowed with a resolution range of 160 to 420 ps. This resolution is a function of frequency input and the multiplication and division factors – i.e., it is a function of the VCO period equal to an eighth of the VCO period. Each clock output counter can choose a different phase of the VCO period from up to eight taps. Designers can use this clock output counter along with an initial setting on the post-scale counter to achieve a phase shift range for the entire period of the output clock. The phase tap feedback to the *m* counter can shift all outputs to a single phase or delay. The Quartus II software automatically sets the phase taps and counter settings according to the phase shift entered.

In addition to the phase shift feature, the fine tune clock delay shift feature provides advanced time delay shift control on each of the four PLL outputs. Each PLL output shifts in 250-ps increments for a range of –3.0 ns to +3.0 ns between any two outputs using discrete delay elements. Total delay shift between any two PLL outputs must be less than 3 ns. For example, shifts on outputs of –1 and +2 ns is allowed, but not –1 and +2.5 ns. There is some delay variation due to process, voltage, and temperature. Only the clock delay shift blocks can be controlled during system operation for dynamic clock delay control.

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 & Programmable Gated Locked

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. A designer may need to gate the lock signal for use as a system control. The enhanced PLL includes a programmable counter that holds the lock signal low for a user-selected number of input clock transitions. This allows the PLL to lock before enabling the lock signal. The designer can use the Quartus II software to set the 20-bit counter value. Either a gated lock signal or an ungated lock signal from the locked port can drive the logic array or an output pin. The device resets and enables both the counter and the PLL simultaneously upon power-up and/or assertion of `pllenable`.

Designers can also combine the lock detection with the `CONF_DONE` signal. This signal indicates that the configuration is complete. This feature holds the `CONF_DONE` signal low until the PLL(s) lock.

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. The designer can use these signals to control PLL resynchronization and the ability to gate PLL output clocks for low power applications.

The `PLLENABLE` pin is a dedicated pin that enables/disables both enhanced and fast 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.

The `areset` signals are reset/resynchronization inputs for each enhanced PLL. The Stratix device can drive these input signals from an input pin or from LEs. When driven high, the PLL counters will reset, clearing the PLL output and placing the PLL out of lock. When driven low again, the PLL will resynchronize to its input as it relocks.

The `scanaclr` signals are resets for the input shift chain registers used in PLL reconfiguration. When high, the entire register chain is cleared. When low, `scanclk` will clock in the serial data to the input shift register.

The `pfdena` signals control the PFD output with a programmable gate. If you disable the PFD, the VCO will operate at its last set value of control voltage and frequency with some drift, and the system will continue running when the PLL goes out of lock or the input clock disables. By maintaining the last locked frequency, the system has time to store its current settings before shutting down. The designer can either use their own control signal or `clk_loss` or gated locked status signals to trigger `pdfena`.

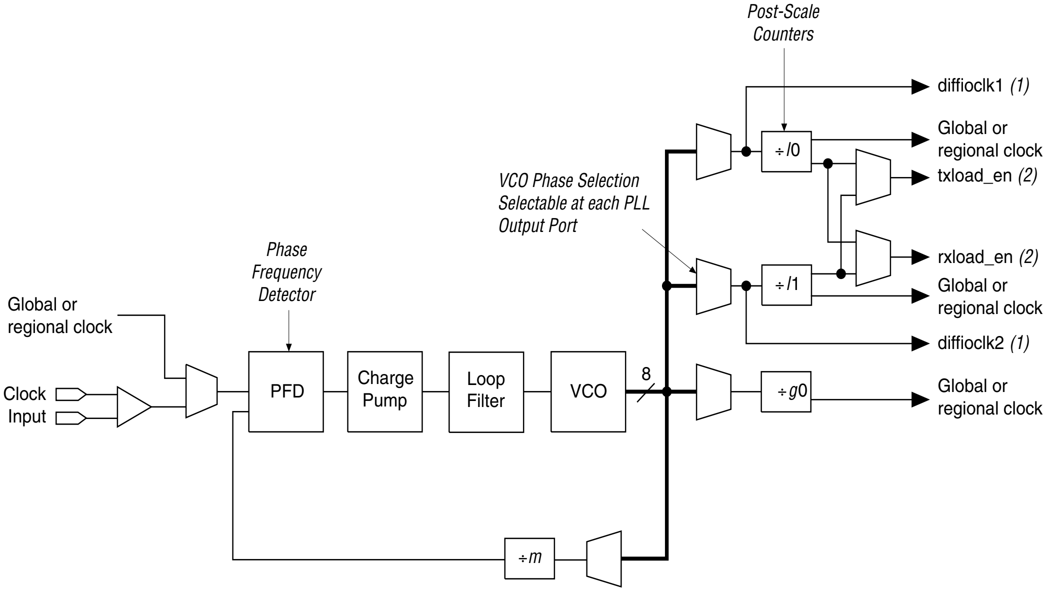
The `clkena` signals control the regional, global, and external outputs of the enhanced PLL. Each regional, global, and external output port has its own `clkena` signal. The `clkena` signals synchronously disable or enable the clock at the PLL output port so the PLL can maintain lock independent of the `clkena` signals. This feature is useful for applications that require 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.

Fast PLLs

Stratix devices contain up to eight enhanced PLLs with high-speed serial interfacing ability, along with general-purpose features. Figure 57 shows a diagram of the fast PLL.

Figure 57. Stratix Device Fast PLL



Notes to Figure 57:

(1) 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.

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

Clock Multiplication & Division

Stratix device enhanced 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, the designer can set the output counter to 1 to allow the high-speed VCO frequency to drive the SERDES.

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.

Table 25 shows the I/O standards supported by fast PLL input pins.

Table 25. Fast PLL Port I/O Standards

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		
LVPECL	✓	
PCML	✓	
LVDS	✓	
HyperTransport technology	✓	
Differential HSTL		
Differential SSTL		
3.3-V GTL	✓	
3.3-V GTL+	✓	
1.5V HSTL class I	✓	
1.5V 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	✓	

Phase Shifting

Stratix device fast PLLs have advanced clock shift capability that enables programmable phase shifts. Designers can enter a phase shift (in degrees or time units) for each PLL clock output port or for all outputs together in one shift. Designers can perform phase shifting in time units with a resolution range of 150 to 400 ps. This resolution is a function 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. Unlike enhanced PLLs, fast PLLs do not have a programmable gated lock signal.

For more information on high-speed differential I/O support, see "High-Speed Differential I/O Support" on page 125.

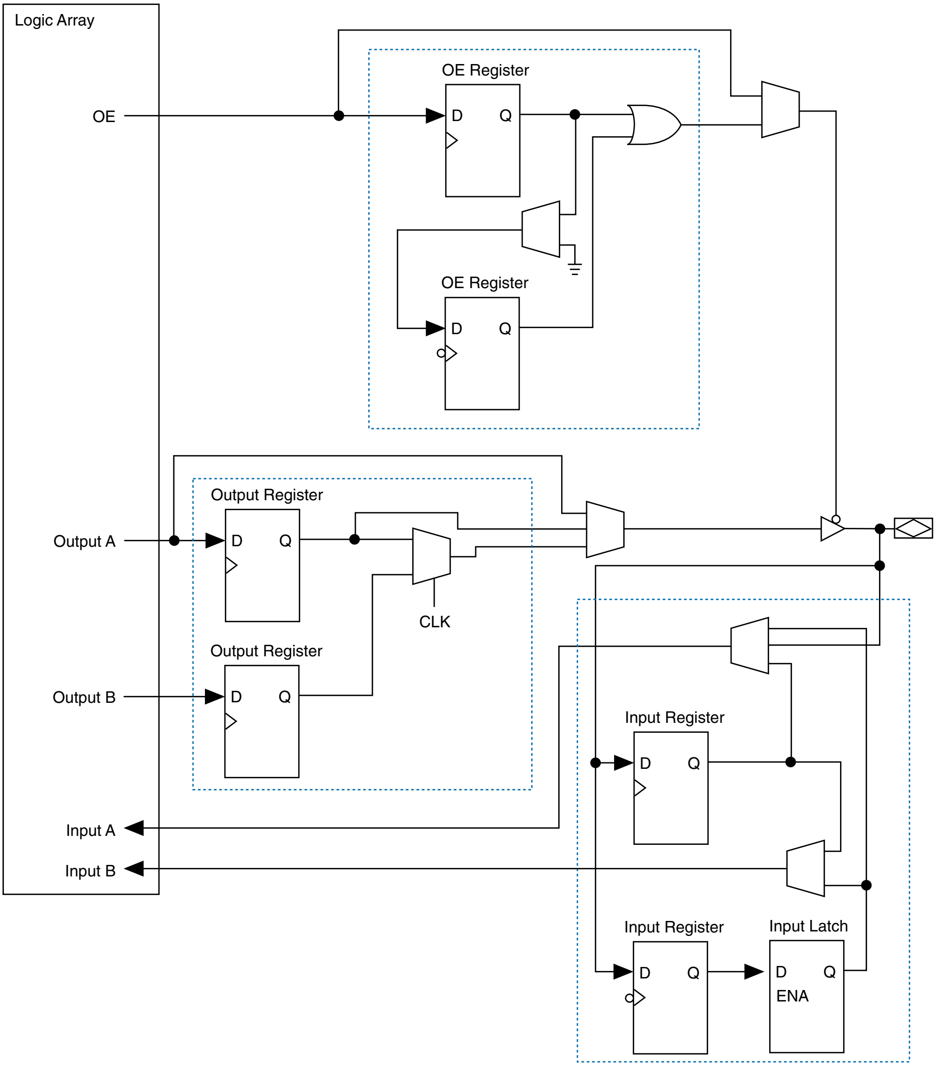
I/O Structure

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 compliance
- Joint Test Action Group (JTAG) boundary-scan test (BST) support
- Driver impedance matching
- On-chip termination for differential and single-ended standards
- 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

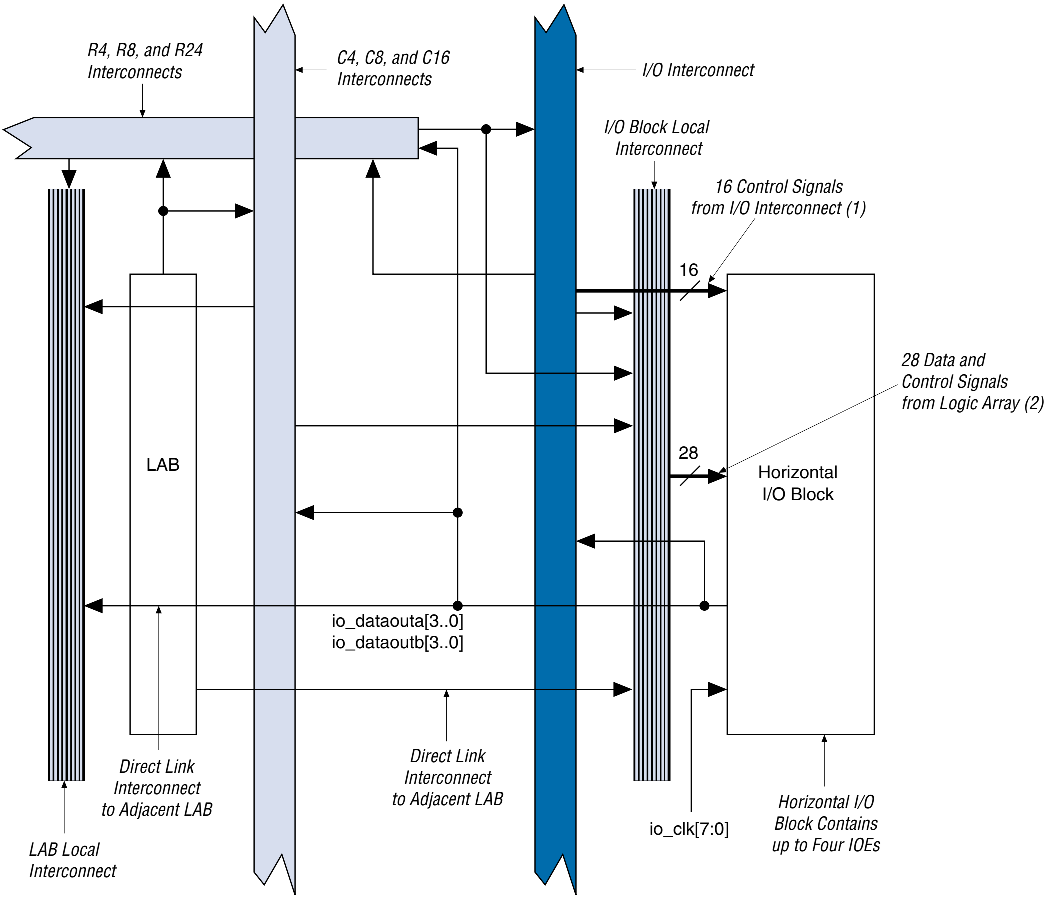
The IOE in Stratix devices contains a bidirectional I/O buffer, six registers, and a latch for a complete embedded bidirectional single data rate or DDR transfer. Figure 58 shows the Stratix IOE structure. The IOE contains two input registers (plus a latch), two output registers, and two output enable 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 58. 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 59 shows how a row I/O block connects to the logic array. Figure 60 shows how a column I/O block connects to the logic array.

Figure 59. Row I/O Block Connection to the Interconnect

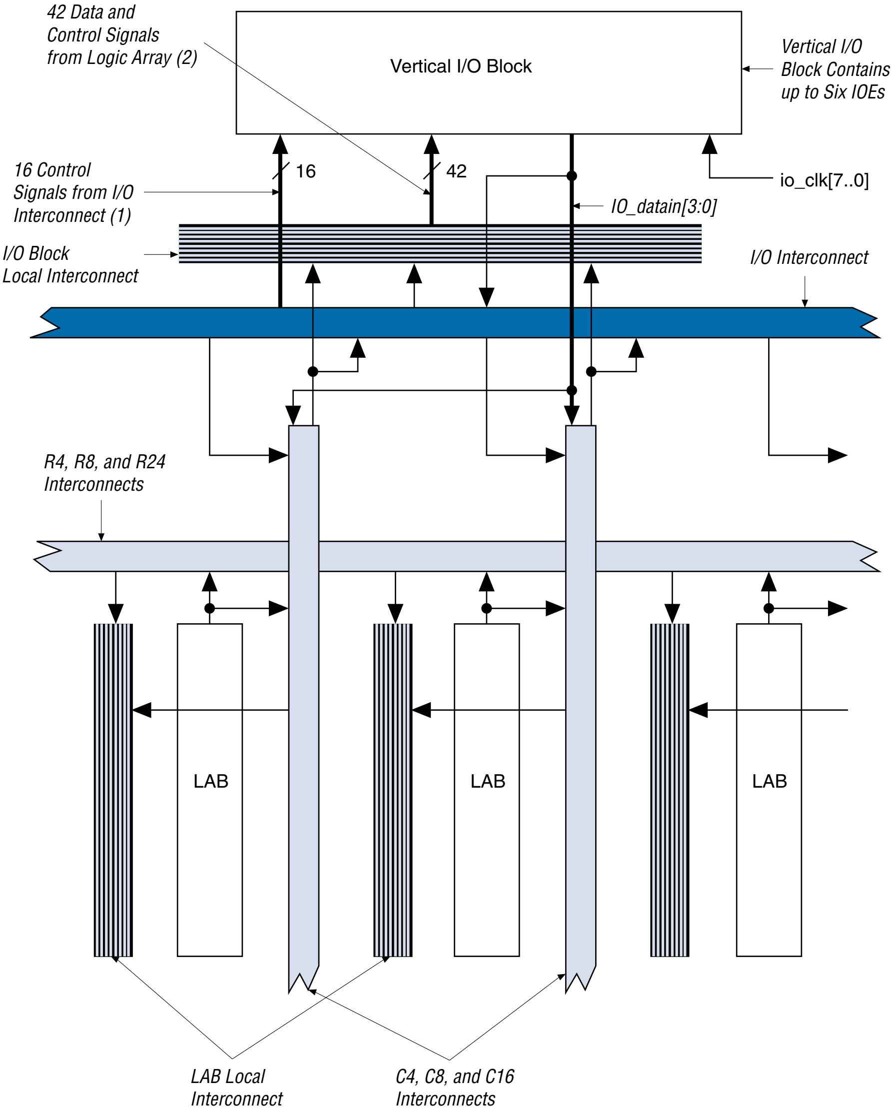


Notes to Figure 59:

(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 60. Column I/O Block Connection to the Interconnect



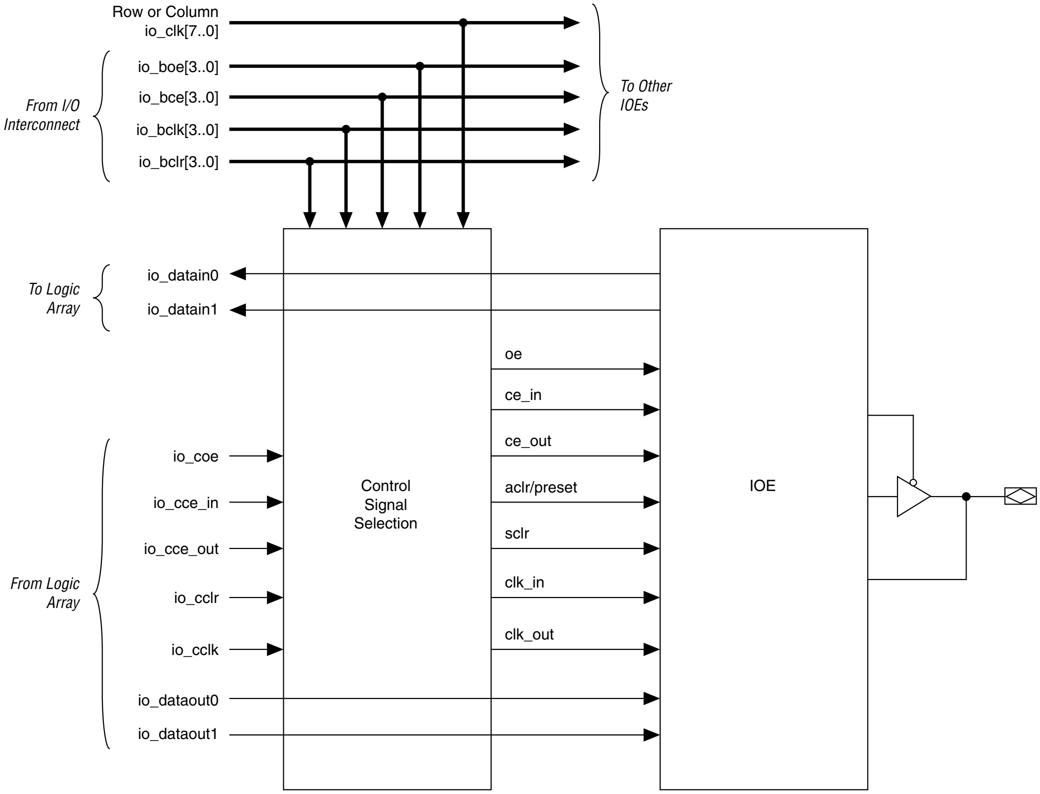
Notes to Figure 60:

(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_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]`.

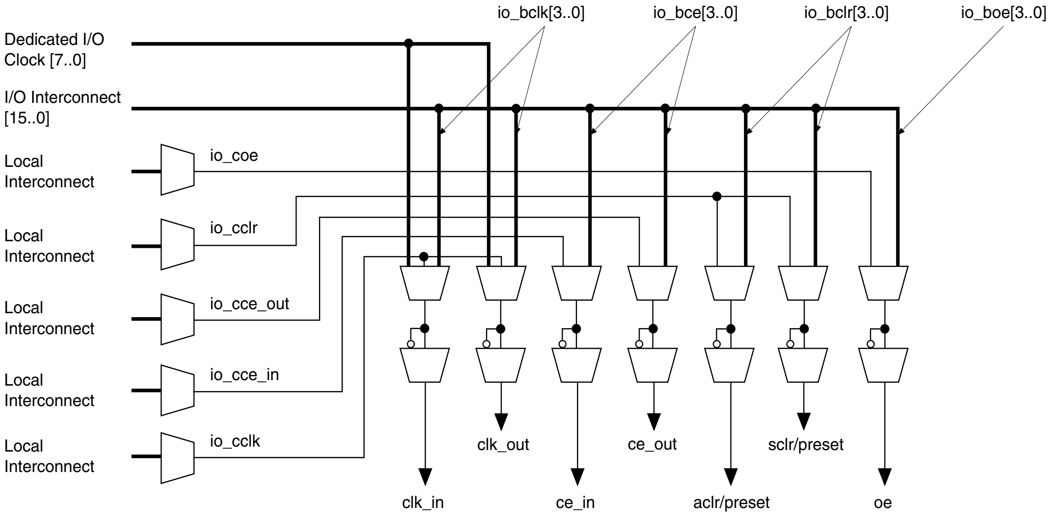
Stratix devices have an I/O interconnect similar to the R4 and C4 interconnect to drive high-fanout signals to and from the I/O blocks. There are 16 signals that drive into the I/O blocks composed of four output enables `io_boe[3..0]`, four clock enables `io_bce[3..0]`, four clocks `io_bclk[3..0]`, and 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 76). Figure 61 illustrates the signal paths through the I/O block.

Figure 61. Signal Path through the I/O Block



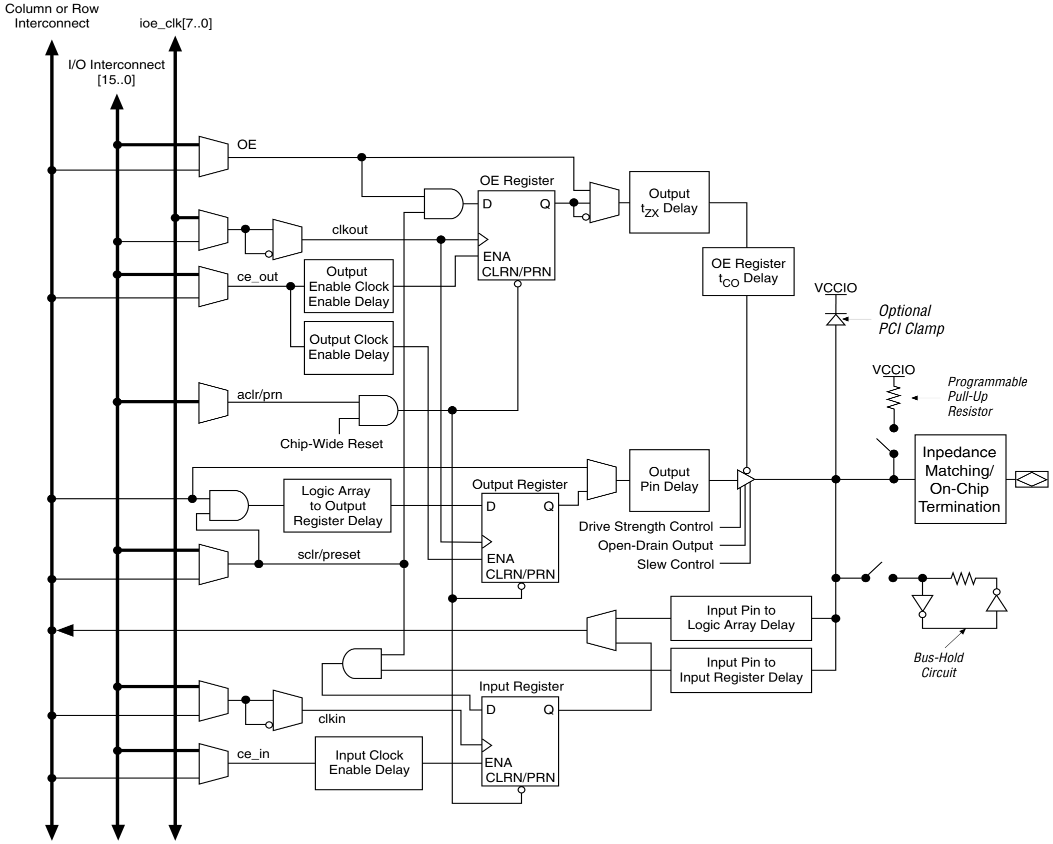
Each IOE contains its own control signal selection for the following control signals: `oe`, `ce_in`, `ce_out`, `aclr/preset`, `sclr/preset`, `clk_in`, and `clk_out`. Figure 62 illustrates the control signal selection.

Figure 62. 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 63 shows the IOE in bidirectional configuration.

Figure 63. Stratix IOE in Bidirectional I/O Configuration



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 26 shows the programmable delays for Stratix devices.

Table 26. 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

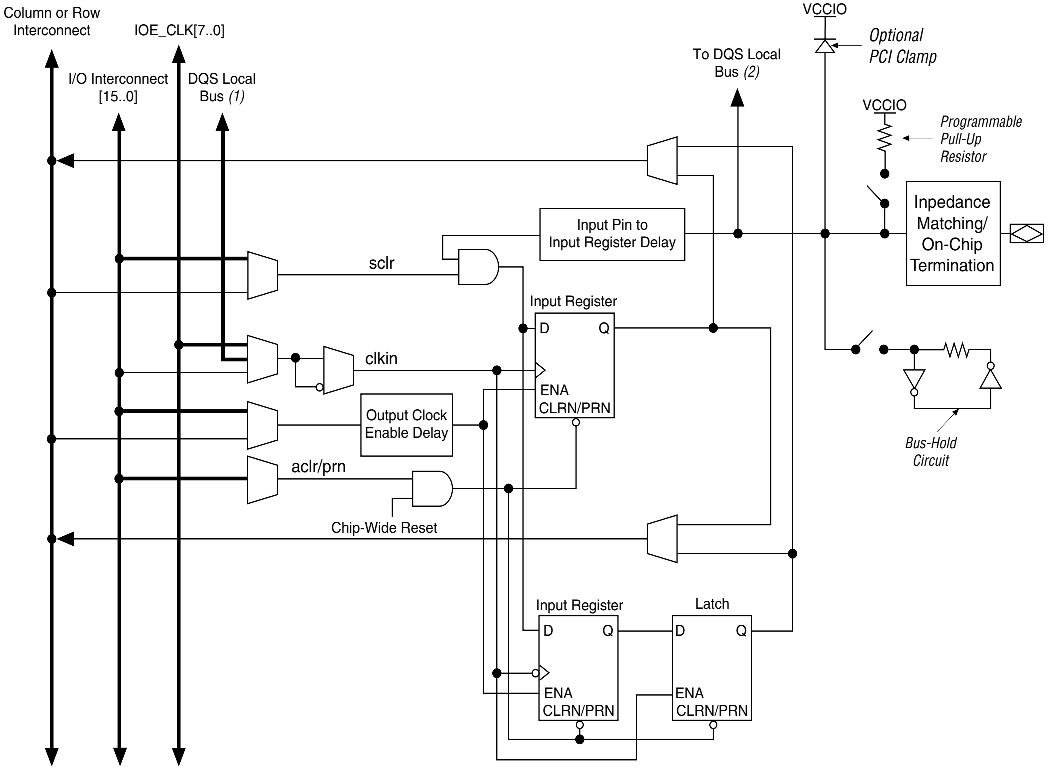
The IOE registers in Stratix devices share the same source for clear or preset. The designer can program preset or clear for each individual IOE. The designer can also program the registers to power up high or low after configuration is complete. If programmed to power up low, an asynchronous clear can 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 to the designer 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 64 shows an IOE configured for DDR input.

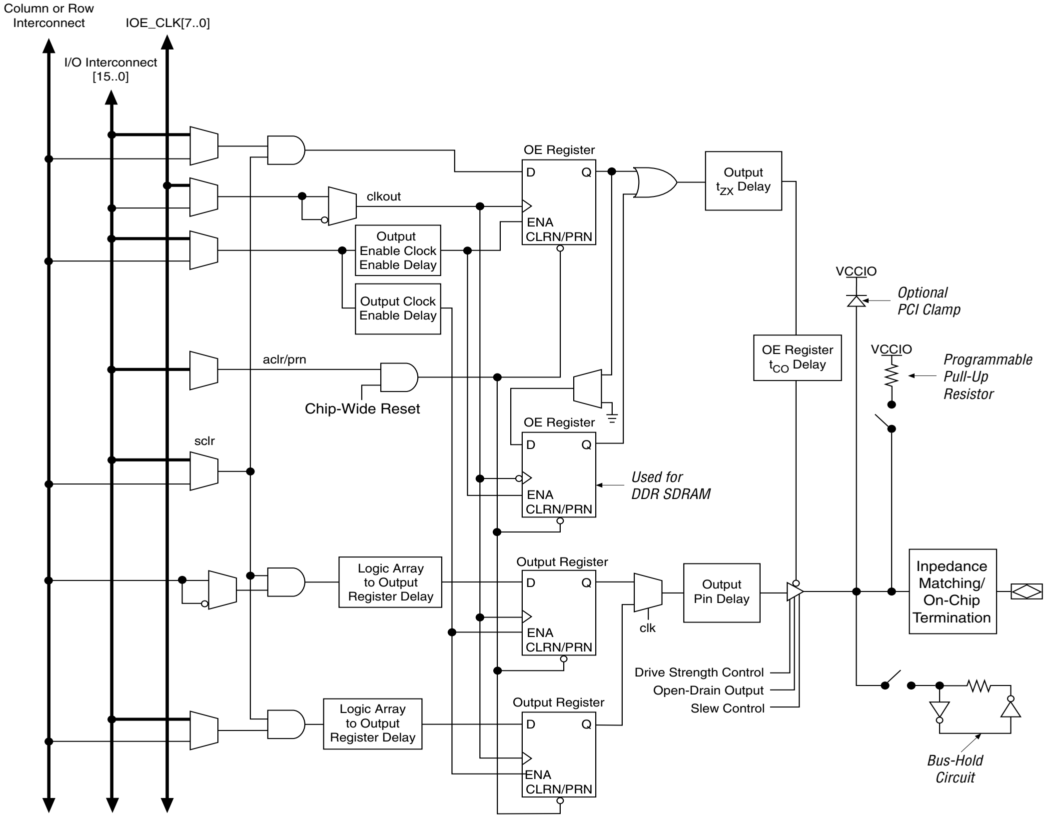
Figure 64. Stratix IOE in DDR Input I/O Configuration



Notes to Figure 64:
(1) This signal connection is only allowed on dedicated DQ function pins.
(2) This signal is for dedicated DQS function pins only.

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 65 shows the IOE configured for DDR output.

Figure 65. Stratix IOE in DDR Output I/O Configuration



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, fast cycle RAM (FCRAM) at up to 200 MHz through dedicated circuitry, and QDR and QDRII SRAM interfaces up to 167 MHz.

DDR SDRAM & FCRAM

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

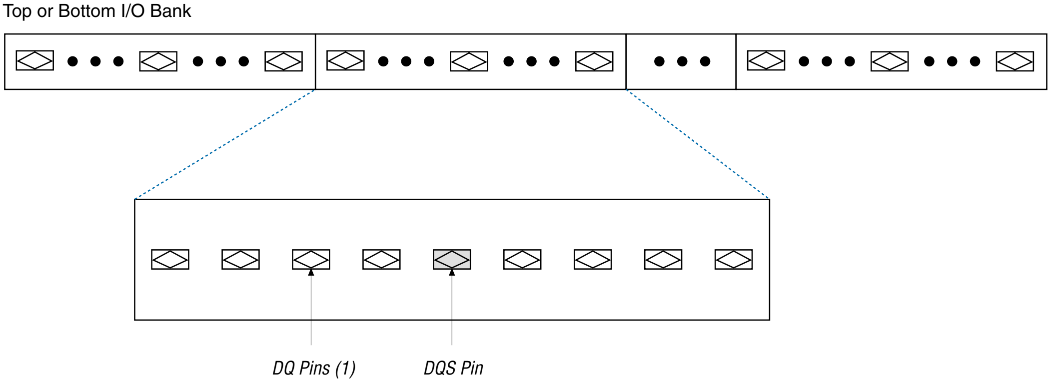
For ×8 mode, there are 20 groups of programmable DQS and DQ pins—10 groups in the top banks and 10 groups in the bottom banks. Each group consists of one DQS pin and a set of at least eight DQ pins (see Figure 66). Each DQS pin drives the set of eight DQ pins within that group.

For ×16 mode, there are eight groups of programmable DQS and DQ pins in the top and bottom banks that are a subset of ×8 DQ pins. Each group consists of one DQS and at least 16 DQ pins.

For ×32 mode, there are four groups of programmable DQS and DQ pins in the top and bottom banks that are also subset of ×8 DQ pins. Each group consists of one DQS and at least 32 DQ pins.

A self-compensated delay element on each DQS pin allows for a 90° phase shift or a 72° phase shift when interfacing to FCRAM. The DQS signals drive onto a local DQS bus within the top and bottom I/O banks. This bus is an additional resource to the IO Clocks and is used to clock DQ input registers with the DQS signal. See Figure 66. These dedicated circuits combine with enhanced PLL clocking and phase shift ability to provide a complete hardware solution for interfacing to high-speed memory.

Figure 66. Stratix Device DQ & DQS Groups in ×8 Mode



Note to Figure 66:

(1) There are at least eight `DQ` pins per group. Some devices may have more.

Zero Bus Turnaround SRAM Interface Support

In addition to DDR SDRAM support, Stratix device I/O pins can also interface with ZBT SRAM devices at up to 200 MHz. ZBT SRAM blocks are designed to eliminate dead bus cycles when turning a bidirectional bus around between reads and writes, or writes and reads. ZBT allows for 100% bus utilization because ZBT SRAM can be read or written on every clock cycle.

To avoid bus contention, the output clock-to-low-impedance time (t_{ZX}) delay ensures that the t_{ZX} is greater than the clock-to-high-impedance time (t_{XZ}). Stratix devices can meet ZBT t_{CO} and t_{SU} times by controlling phase delay in clocks to the OE/output and input registers using an enhanced PLL.

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 27 shows the possible settings for the I/O standards with drive strength control.

Table 27. Programmable Drive Strength

I/O Standard	I_{OH}/I_{OL} Current Strength Setting (mA)
LVTTL (3.3 V)	4
	8
	12
	16
	24
LVCMOS (3.3 V)	2
	4
	8
	12
	24 *(1)*
LVTTL (2.5 V)	2
	8
	12
	16
LVTTL (1.8 V)	2
	8
	12
LVTTL (1.5 V)	2
	4
	8

Note to Table 27:

(1) Banks 1, 2, 5, and 6 do not support this setting.

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 the designer 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, you do not need an external pull-up or pull-down resistor to hold a signal level when the bus is tri-stated.

The bus-hold circuitry also pulls undriven pins away from the input threshold voltage where noise can cause unintended high-frequency switching. The designer can select this feature individually for each I/O pin. The bus-hold output will drive 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. Table 68 on page 154 gives 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 you enable this feature 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.

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
- 3.3-V AGP (1× and 2×)
- LVDS
- LVPECL
- PCML
- HyperTransport
- Differential HSTL (on input/output clocks only)
- Differential SSTL (on output clocks only)
- GTL/GTL+
- 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 28 describes the I/O standards supported by Stratix devices.

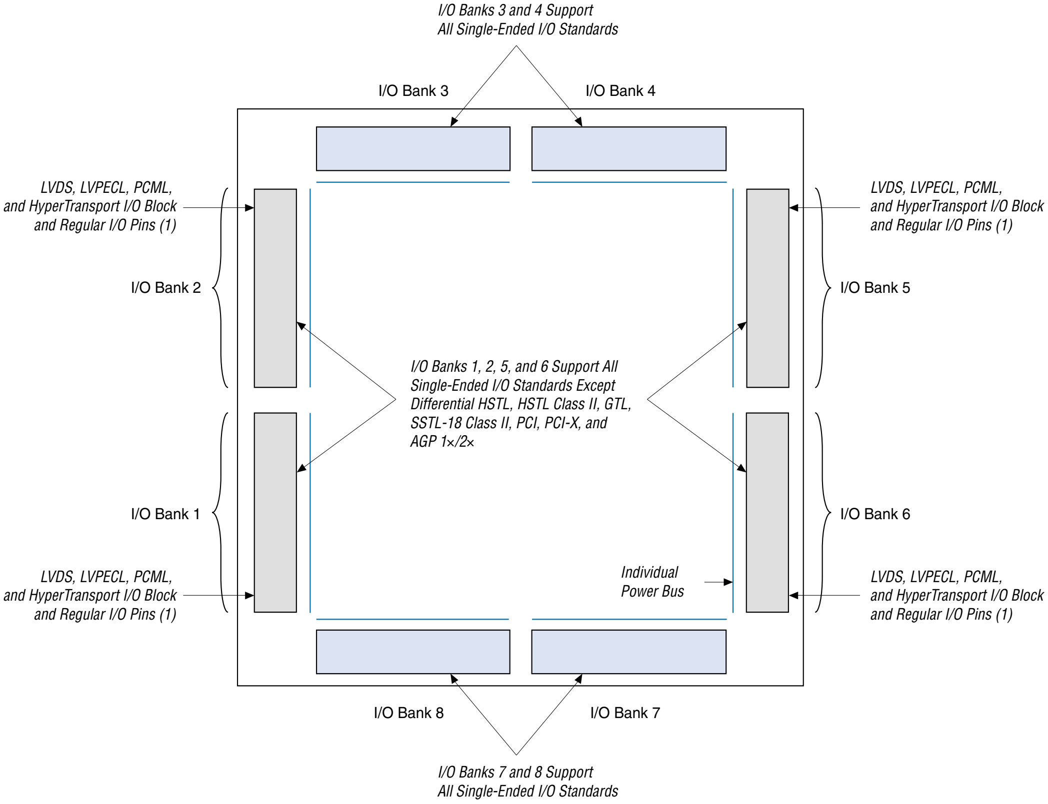
Table 28. Stratix Supported I/O Standards

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	Single-ended	N/A	3.3	N/A
LVDS	Differential	N/A	3.3	N/A
LVPECL	Differential	N/A	3.3	N/A
PCML	Differential	N/A	3.3	N/A
HyperTransport	Differential	N/A	2.5	N/A
Differential HSTL	Differential	N/A	1.5	N/A
Differential SSTL	Differential	N/A	2.5	N/A
GTL / GTL+	Voltage-referenced	1.0	N/A	1.5
HSTL class I and II	Voltage-referenced	0.75	1.5	0.75
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

For more information on I/O standards supported by Stratix devices, see *Application Note 201 (Using Selectable I/O Standards in Stratix Devices)*.

Stratix devices contain eight I/O banks, as shown in Figure 67. The four I/O banks on the right and left of the device contain circuitry to support high-speed differential I/O support, LVPECL, PCML, and HyperTransport inputs and other outputs. These banks support all I/O standards listed in Table 28 except PCI I/O pins or PCI-X, GTL, SSTL-18 Class II, and HSTL Class II outputs. The top and bottom I/O banks support all single-ended I/O standards. Table 29 shows I/O standard support for each I/O bank.

Figure 67. Stratix I/O Banks



Note to Figure 67:

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

Table 29. I/O Support by Bank

I/O Standard	Top & Bottom Banks (3, 4, 7 & 8)	Left & Right Banks (1, 2, 5 & 6)
LVTTL	✓	✓
LVCMOS	✓	✓
2.5 V	✓	✓
1.8 V	✓	✓
1.5 V	✓	✓
3.3-V PCI	✓	
3.3-V PCI-X	✓	
LVPECL		✓
PCML		✓
LVDS		✓
HyperTransport		✓
Differential HSTL (clock inputs/outputs)	✓	
Differential SSTL (clock outputs)	✓	
3.3-V GTL	✓	*(1)*
3.3-V GTL+	✓	✓
1.5-V HSTL class I	✓	✓
1.5-V HSTL class II	✓	*(1)*
SSTL-18 class I	✓	✓
SSTL-18 class II	✓	*(1)*
SSTL-2 class I	✓	✓
SSTL-2 class II	✓	✓
SSTL-3 class I	✓	✓
SSTL-3 class II	✓	✓
AGP (1× and 2×)	✓	*(1)*
CTT	✓	✓

Note to Table 29:

(1) These I/O standards are only supported for input pins.

Each I/O bank has its own `VCCIO` pins. A single device can support 1.5-V, 1.8-V, 2.5-V, 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.

Terminator Technology

Terminator technology provides on-chip parallel and differential termination and impedance matching (series termination) to reduce reflections and maintain signal integrity. Terminator technology simplifies board design by minimizing the number of external termination resistors required. These resistors can be placed inside the package, eliminating small stubs that can still lead to reflections. Additionally, the terminator technology provides constant calibration of the internal resistor values after configuration and during normal operation via two external reference resistors. The constant calibration allows the termination resistors to compensate for process, temperature, and voltage variation, providing a robust termination scheme. There is one set of reference resistors for each I/O bank.

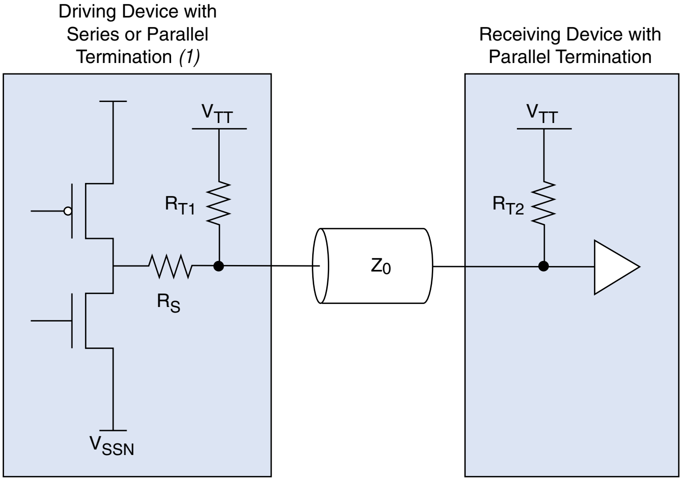
Three types of termination are available in the device:

- Series Termination (R_S) and Impedance Matching
- Parallel Termination (R_T)
- Differential Termination (R_D)

Stratix devices support series termination for SSTL-3 and SSTL-2 signals to meet SSTL specifications. Stratix devices also support driver impedance matching through series termination for LVTTL and LVCMOS signals to match the impedance of the transmission lines, typically 25 or 50 Ω. When used with the output drivers, the terminator technology sets the output driver impedance to 25 or 50 Ω as specified by the external reference resistors, resulting in significantly reduced reflections.

Parallel termination is supported for SSTL-3, SSTL-2, HSTL, GTL, GTL+, and CTT signals as defined by the respective I/O standards. Figure 68 illustrates the possible termination schemes for single-ended I/O pins.

Figure 68. Termination Schemes for Single-Ended I/O Pins

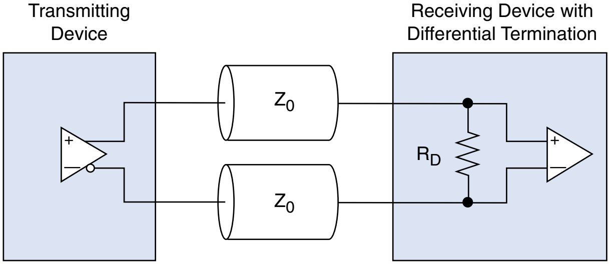


Note to Figure 68:

(1) In the transmitting device, only one type of termination: series or parallel termination is possible. For standards that require both terminations, such as SSTL 2 Class II, an external parallel termination resistor must be provided.

Stratix devices support differential termination with a 100-Ω resistor for LVDS signals. Figure 69 shows the device with differential termination.

Figure 69. Differential LVDS Input On-Chip Termination



Terminator technology can only support one type of termination per I/O bank, although some different I/O standards can be mixed within a given I/O bank. I/O banks at the top and bottom of the device support series termination and impedance matching and parallel termination. I/O banks on the left and right side of the device support series termination and impedance matching and LVDS far-end differential termination. Each I/O bank utilizing on-chip termination must connect two external reference resistors, R_{UP} and R_{DN}, to the designated pins in the I/O bank. The designer sets which pins are terminated and match the reference resistors. After configuration and during normal operation, the device periodically samples the external resistor values and updates the internal resistor values. Table 30 shows the Terminator technology support within each I/O bank.

Table 30. Terminator Technology Support by I/O Banks

Terminator Technology Support	Top & Bottom Banks (3, 4, 7 & 8)	Left & Right Banks (1, 2, 5 & 6)
Series termination	✓	✓
Impedance matching (LVTTL/LVCMOS)	✓	✓
Parallel termination *(1)*	✓	
Differential termination		✓

Note to Table 32:

(1) Clock pins CLK[0..3] and CLK[8..11] do not support parallel termination.

Table 31 summarizes the external resistor values required for terminator technology.

Table 31. External Resistor Values

Parameter	R_{UP}	R_{DN}
Series termination	250 Ω	250 Ω
Impedance matching (LVTTL/LVCMOS)	250 Ω / 500 Ω	250 Ω / 500 Ω
Parallel termination *(1)*	1,000 Ω	1,000 Ω
Differential termination	*(2)*	*(2)*

Notes to Table 31:

(1) Stratix devices support parallel termination on the top and bottom I/O banks only.
(2) No external resistor is necessary.

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 devices have one set of `VCC` pins for internal operation and input buffers (`VCCINT`), and eight sets for I/O output drivers (`VCCIO`).

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 32 summarizes Stratix MultiVolt I/O support.

Table 32. Stratix MultiVolt I/O Support *Note (1)*

VCCIO (V)	Input Signal					Output Signal				
	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	✓	✓	✓	✓		✓				
1.8		✓	✓	✓		✓ *(2)*	✓			
2.5			✓	✓		✓ *(3)*	✓ *(3)*	✓		
3.3			✓	✓	✓ *(4)*	✓ *(5)*	✓ *(5)*	✓ *(5)*	✓	✓

Notes to Table 32:

(1) The PCI clamping diode must be disabled to drive an input with voltages higher than V_{CCIO}.
(2) When V_{CCIO} = 1.8 V, a Stratix device can drive a 1.5-V device with 1.8-V tolerant inputs.
(3) When V_{CCIO} = 2.5 V, a Stratix device can drive a 1.5-V or 1.8-V device with 2.5-V tolerant inputs.
(4) Stratix devices can be 5.0-V tolerant with the use of an external resistor and internal PCI clamp diode.
(5) When V_{CCIO} = 3.3 V, a Stratix device can drive a 1.5-V, 1.8-V, or 2.5-V device with 3.3-V tolerant inputs.

Stratix devices include a dedicated pin, `VCCSEL`, to choose whether the configuration and JTAG pins can accept 3.3 V/2.5 V or 1.8 V during configuration. Setting the pin to high (1.5/1.8/2.5/3.3 V) sets the configuration and JTAG I/O pins to accept 1.8 V, while low (0 V) sets them to accept 3.3/2.5 V.

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 PCML.

There are four dedicated high-speed PLLs in the EP1S10 to EP1S25 devices and eight dedicated high-speed PLLs in the EP1S30 to EP1S120 devices to multiply reference clocks and drive high-speed differential SERDES channels. Table 33 shows high-speed differential I/O support in Stratix devices.

Table 33. Differential I/O Support in Stratix Devices

Device	Number of Fast PLLs	Number of Receiver Channels *(1)*	Number of Transmitter Channels *(1)*
EP1S10	4	44	44
EP1S20	4	66	66
EP1S25	4	78	78
EP1S30	8	82 *(2)*	82 *(2)*
EP1S40	8	90 *(2)*	90 *(2)*
EP1S60	8	116 *(2)*	116 *(2)*
EP1S80	8	80/56 *(3)*	40/112 *(4)*
EP1S120	8	80/56 *(3)*	40/112 *(4)*

Notes to Table 33:

(1) Table 33 provides the maximum number of channels supported for each device. Some devices may have additional packages that offer fewer channels than what is listed. The number of channels available in each package is shown in *Application Note 202 (Using High-Speed I/O Interfaces in Stratix Devices)*.

(2) A small number of these channels operate at 462 Mbps; the remaining channels operate at 840 Mbps. The number of channels operating at 462 Mbps is shown in *Application Note 202 (Using High-Speed I/O Interfaces in Stratix Devices)*.

(3) These devices support 80 receiver channels at 840 Mbps and up to 56 receiver channels at 462 Mbps.

(4) These devices support at least 40 transmitter channels at 840 Mbps and up to 112 transmitter channels at 462 Mbps.

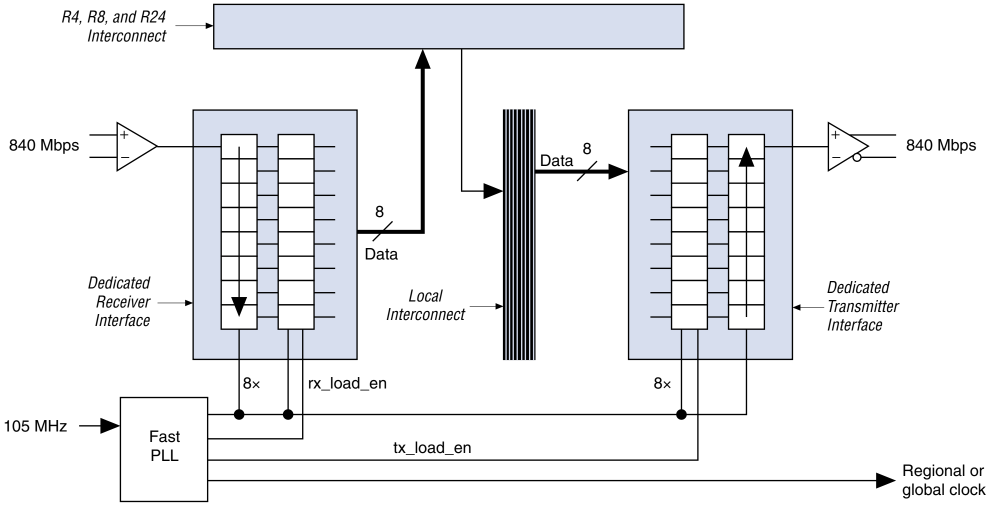
The high-speed differential I/O circuitry supports the following high speed I/O interconnect standards and applications:

- 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, 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 multiplies the clock by a factor of 1, 2, 4, 8, or 10. The serialization/deserialization factor can be any number from 1, 2, 4, 8, or 10 and does not have to equal the clock-multiplication value. For a SERDES factor of 1, the Stratix device bypasses the SERDES. For a SERDES factor of 2, the DDR input and output is used in the IOE. Figure 70 illustrates the dedicated receiver and transmitter interface.

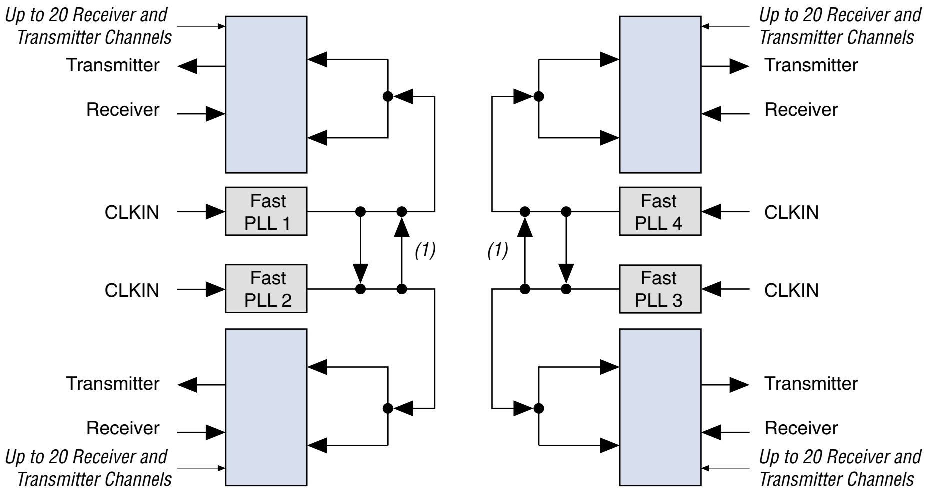
Figure 70. High-Speed Differential I/O Receiver / Transmitter Interface Example



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.

Each fast PLL can support up to 20 high-speed 840-Mbps receiver and transmitter high-speed differential I/O channels. 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 row. Figure 71 shows the fast PLL and channel layout in EP1S10, EP1S20, and EP1S25 devices. Figure 72 shows the fast PLL and channel layout in the EP1S30 to EP1S120 devices.

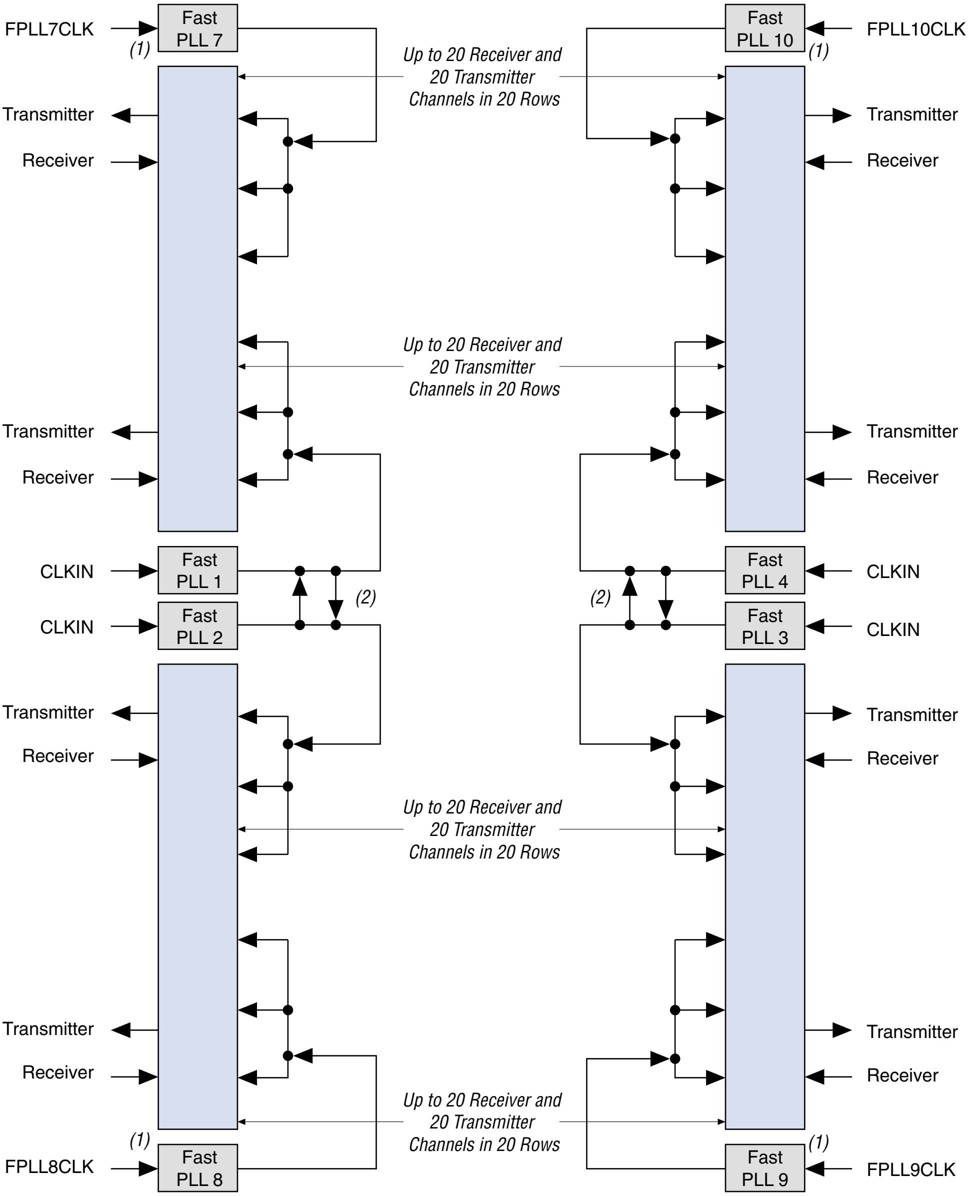
Figure 71. Fast PLL & Channel Layout in the EP1S10, EP1S20 or EP1S25 Devices



Note to Figure 71:

(1) 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 462 Mbps. For example, if PLL 2 clocks PLL 1’s channel region, then those channels support up to 462 Mbps.

Figure 72. Fast PLL & Channel Layout in the EP1S30 to EP1S120 Devices



Notes to Figure 72:

(1) For EP1S80 and EP1S120 devices, the fast PLLs located at the corners of the device support up to 462 Mbps.

(2) 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 462 Mbps. For example, if PLL 2 clocks PLL 1's channel region, then those channels support up to 462 Mbps.

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, 1/8, 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 DDR.

Byte Alignment

For high-speed source synchronous interfaces such as POS-PHY 4, XSBI, RapidIO, and HyperTransport, 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 bit slipping 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. The designer 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.

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.

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All Stratix devices provide JTAG BST circuitry that complies with the IEEE Std. 1149.1a-1990 specification. JTAG boundary-scan testing can be performed either before or after, but not during configuration. Stratix devices can also use the JTAG port for configuration together with either the Quartus II software or hardware using either Jam Files (**.jam**) or Jam Byte-Code Files (**.jbc**).

Stratix devices support IOE I/O standard setting reconfiguration through the JTAG BST chain. The JTAG chain can update the I/O standard for all input and output pins any time before or during user mode. Designers can use this ability for JTAG testing before configuration when some of the Stratix pins drive or receive from other devices on the board using voltage-referenced standards. Since the Stratix device may not be configured before JTAG testing, the I/O pins may not be configured for appropriate electrical standards for chip-to-chip communication. Programming those I/O standards via JTAG allows full designers to fully test I/O connection to other devices.

The enhanced PLL reconfiguration bits are part of the JTAG chain before configuration and after power-up. After device configuration, the PLL reconfiguration bits are not part of the JTAG chain.

Stratix devices also use the JTAG port to monitor the logic operation of the device with the SignalTap embedded logic analyzer. Stratix devices support the JTAG instructions shown in Table 34.

Table 34. Stratix JTAG Instructions

JTAG Instruction	Description
`SAMPLE/PRELOAD`	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation, and permits an initial data pattern to be output at the device pins. Also used by the SignalTap embedded logic analyzer.
`EXTEST` *(1)*	Allows the external circuitry and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
`BYPASS`	Places the 1-bit bypass register between the `TDI` and `TDO` pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.
`USERCODE`	Selects the 32-bit USERCODE register and places it between the `TDI` and `TDO` pins, allowing the USERCODE to be serially shifted out of `TDO`.
`IDCODE`	Selects the IDCODE register and places it between `TDI` and `TDO`, allowing the IDCODE to be serially shifted out of `TDO`.
`HIGHZ` *(1)*	Places the 1-bit bypass register between the `TDI` and `TDO` pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation, while tri-stating all of the I/O pins.
`CLAMP` *(1)*	Places the 1-bit bypass register between the `TDI` and `TDO` pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding I/O pins to a state defined by the data in the boundary-scan register.
ICR instructions	Used when configuring an Stratix device via the JTAG port with a MasterBlaster™ or ByteBlasterMV™ download cable, or when using a Jam File or Jam Byte-Code File via an embedded processor.
`PULSE_NCONFIG`	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is unaffected.
`CONFIG_IO`	Allows the IOE standards to be configured through the JTAG chain. Stops configuration if executed during configuration. Can be executed before or after configuration.
SignalTap instructions	Monitors internal device operation with the SignalTap embedded logic analyzer.

Note to Table 34:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of `HIGHZ`, `CLAMP`, and `EXTEST`.

The Stratix device instruction register length is 10 bits and the USERCODE register length is 32 bits. Tables 35 and 36 show the boundary-scan register length and device IDCODE information for Stratix devices.

Table 35. Stratix Boundary-Scan Register Length

Device	Boundary-Scan Register Length
EP1S10	1,317
EP1S20	1,797
EP1S25	2,157
EP1S30	2,253
EP1S40	2,637
EP1S60	3,129
EP1S80	3,777
EP1S120	(1)

Table 36. 32-Bit Stratix Device IDCODE

Device	IDCODE (32 Bits) (2)			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) (3)
EP1S10	0000	0010 0000 0000 0001	000 0110 1110	1
EP1S20	0000	0010 0000 0000 0010	000 0110 1110	1
EP1S25	0000	0010 0000 0000 0011	000 0110 1110	1
EP1S30	0000	0010 0000 0000 0100	000 0110 1110	1
EP1S40	0000	0010 0000 0000 0101	000 0110 1110	1
EP1S60	0000	0010 0000 0000 0110	000 0110 1110	1
EP1S80	0000	0010 0000 0000 0111	000 0110 1110	1
EP1S120	0000	(1)	000 0110 1110	1

Notes to Tables 35 and 36:

(1) Contact Altera Applications for up-to-date information on this device.
(2) The most significant bit (MSB) is on the left.
(3) The IDCODE's least significant bit (LSB) is always 1.

Figure 73 shows the timing requirements for the JTAG signals.

Figure 73. Stratix JTAG Waveforms

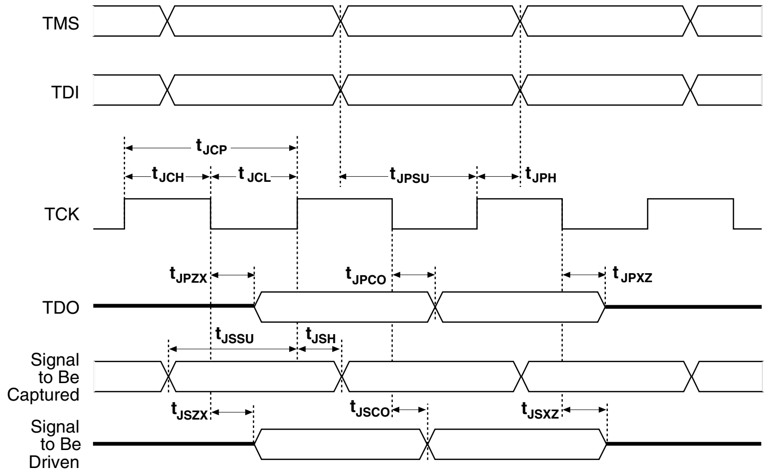


Table 37 shows the JTAG timing parameters and values for Stratix devices.

Table 37. Stratix JTAG Timing Parameters & Values

Symbol	Parameter	Min	Max	Unit
t_{JCP}	`TCK` clock period	100		ns
t_{JCH}	`TCK` clock high time	50		ns
t_{JCL}	`TCK` clock low time	50		ns
t_{JPSU}	JTAG port setup time	20		ns
t_{JPH}	JTAG port hold time	45		ns
t_{JPCO}	JTAG port clock to output		25	ns
t_{JPZX}	JTAG port high impedance to valid output		25	ns
t_{JPXZ}	JTAG port valid output to high impedance		25	ns
t_{JSSU}	Capture register setup time	20		ns
t_{JSH}	Capture register hold time	45		ns
t_{JSCO}	Update register clock to output		35	ns
t_{JSZX}	Update register high impedance to valid output		35	ns
t_{JSXZ}	Update register valid output to high impedance		35	ns

For more information on JTAG, see the following documents:

- *Application Note 39 (IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices)*
- *Jam Programming & Test Language Specification*

SignalTap Embedded Logic Analyzer

Stratix devices feature the SignalTap embedded logic analyzer, which monitors design operation over a period of time through the IEEE Std. 1149.1 (JTAG) circuitry. A designer can analyze internal logic at speed without bringing internal signals to the I/O pins. This feature is particularly important for advanced packages, such as FineLine BGA packages, because it can be difficult to add a connection to a pin during the debugging process after a board is designed and manufactured.

Configuration

The logic, circuitry, and interconnects in the Stratix architecture are configured with CMOS SRAM elements. Stratix devices are reconfigurable and are 100% tested prior to shipment. As a result, the designer does not have to generate test vectors for fault coverage purposes, and can instead focus on simulation and design verification. In addition, the designer does not need to manage inventories of different ASIC designs. Stratix devices can be configured on the board for the specific functionality required.

Stratix devices are configured at system power-up with data stored in an Altera serial configuration device or provided by a system controller. Altera offers in-system programmability (ISP)-capable configuration devices that configure Stratix devices via a serial data stream. Stratix devices can be configured in under 100 ms using 8-bit parallel data at 100 MHz. The Stratix device's optimized interface allows microprocessors to configure it serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat Stratix devices as memory and configure them by writing to a virtual memory location, making reconfiguration easy. After a Stratix device has been configured, it can be reconfigured in-circuit by resetting the device and loading new data. Real-time changes can be made during system operation, enabling innovative reconfigurable computing applications.

Operating Modes

The Stratix architecture uses SRAM configuration elements that require configuration data to be loaded each time the circuit powers up. The process of physically loading the SRAM data into the device is called configuration. During initialization, which occurs immediately after configuration, the device resets registers, enables I/O pins, and begins to operate as a logic device. The I/O pins are tri-stated during power-up, and before and during configuration. Together, the configuration and initialization processes are called command mode. Normal device operation is called user mode.

A built-in weak pull-up resistor pulls all user I/O pins to V_{CCIO} before and during device configuration.

SRAM configuration elements allow Stratix devices to be reconfigured in-circuit by loading new configuration data into the device. With real-time reconfiguration, the device is forced into command mode with a device pin. The configuration process loads different configuration data, reinitializes the device, and resumes user-mode operation. Designers can perform in-field upgrades by distributing new configuration files either within the system or remotely.

Configuration Schemes

Designers can load the configuration data for a Stratix device with one of five configuration schemes (see Table 38), chosen on the basis of the target application. Designers can use a configuration device, intelligent controller, or the JTAG port to configure a Stratix device. A configuration device can automatically configure a Stratix device at system power-up.

Multiple Stratix devices can be configured in any of five configuration schemes by connecting the configuration enable (`nCE`) and configuration enable output (`nCEO`) pins on each device.

Table 38. Data Sources for Configuration

Configuration Scheme	Data Source
Configuration device	Enhanced or EPC2 configuration device
Passive serial (PS)	ByteBlasterMV or MasterBlaster download cable or serial data source
Passive parallel asynchronous (PPA)	Parallel data source
Fast passive parallel	Parallel data source
JTAG	MasterBlaster or ByteBlasterMV download cable or a microprocessor with a Jam or JBC file

Partial Reconfiguration

The enhanced PLLs within the Stratix device family support partial reconfiguration of their multiply, divide, and time delay settings without reconfiguring the entire device. Designers can use either serial data from the logic array or regular I/O pins to program the PLL's counter settings in a serial chain. This option provides considerable flexibility for frequency synthesis, allowing real-time variation of the PLL frequency and delay. The rest of the device is functional while reconfiguring the PLL. See "Enhanced PLLs" on page 90 for more information on Stratix PLLs.

Remote Update Configuration Modes

Stratix devices also support remote configuration using an Altera enhanced configuration device (e.g., EPC16, EPC8, and EPC4 devices) with page mode selection. Factory configuration data is stored in the default page of the configuration device. This is the default configuration which contains the design required to control remote updates and handle or recover from errors. The designer writes the factory configuration once into the flash memory or configuration device. Remote update data can update any of the remaining pages of the configuration device. If there is an error or corruption in a remote update configuration, the configuration device reverts back to the factory configuration information.

There are two remote configuration modes: remote and local configuration. Designers can use the remote update configuration mode for all three configuration modes: serial, parallel synchronous, and parallel asynchronous. Configuration devices (e.g., EPC16 devices) only support serial and parallel synchronous modes. Asynchronous parallel mode allows remote updates when an intelligent host is used to configure the Stratix device. This host must support page mode settings similar to an EPC16 device.

Remote Update Mode

When the Stratix device is first powered up in remote update programming mode, it loads the configuration located at page address "`000`." The factory configuration should always be located at page address "`000`," and should never be remotely updated. The factory configuration contains the required logic to perform the following operations:

- Determine the page address/load location for the next application's configuration data
- Recover from a previous configuration error
- Receive new configuration data and write it into the configuration device

The factory configuration is the default and takes control if an error occurs while loading the application configuration.

While in the factory configuration, the factory-configuration logic performs the following operations:

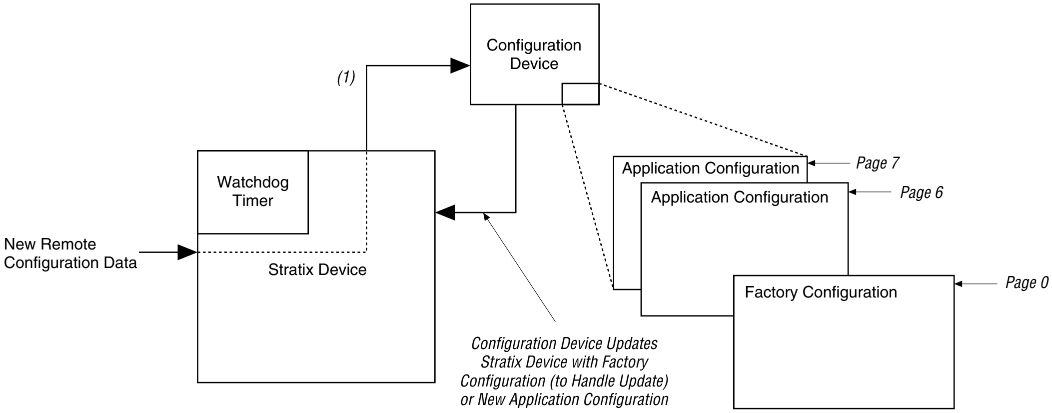
- Loads a remote update-control register to determine the page address of the new application configuration
- Determines whether to enable a user watchdog timer for the application configuration
- Determines what the watchdog timer setting should be if it is enabled

The user watchdog timer is a counter that must be continually reset within a specific amount of time in the user mode of an application configuration to ensure that valid configuration occurred during a remote update. Only valid application configurations designed for remote update can reset the user watchdog timer in user mode. If a valid application configuration does not reset the user watchdog timer in a specific amount of time, the timer updates a status register and loads the factory configuration. The user watchdog timer is automatically disabled for factory configurations.

If an error occurs in loading the application configuration, the configuration logic writes a status register to specify the cause of the reconfiguration. Once this occurs, the Stratix device automatically loads the factory configuration, which reads the status register and determines the reason for reconfiguration. Based on the reason, the factory configuration will take appropriate steps and will write the remote update control register to specify the next application configuration page to be loaded.

When the Stratix device successfully loads the application configuration, it enters into user mode. The Stratix device then executes the main application of the user. Intellectual property (IP), such as a Nios™ embedded processor, can help the Stratix device determine when remote update is coming. The Nios embedded processor or user logic receives incoming data, writes it to the configuration device, and loads the factory configuration. The factory configuration will read the remote update status register and determine the valid application configuration to load. Figure 74 shows the Stratix remote update. Figure 75 shows the transition diagram for remote update mode.

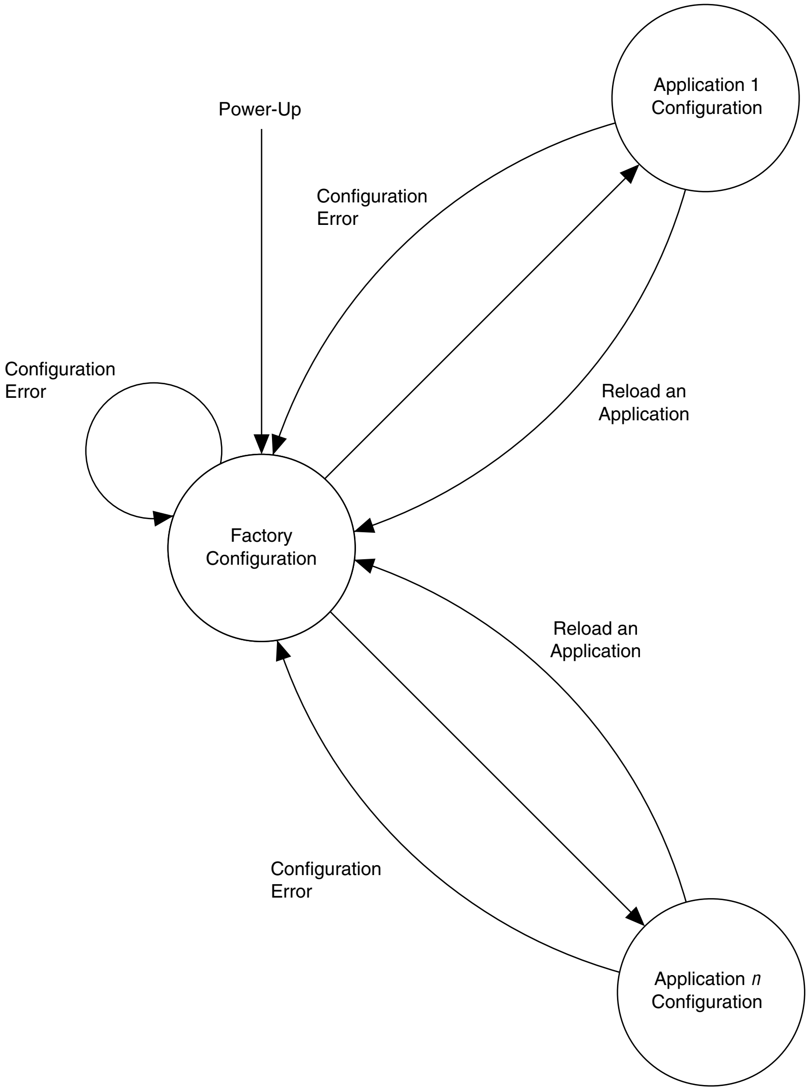
Figure 74. Stratix Device Remote Update



Note to Figure 74:

(1) When the Stratix device is configured with the factory configuration, it can handle update data from EPC16, EPC8, or EPC4 configuration device pages and point to the next page in the configuration device.

Figure 75. Remote Update Transition Diagram Notes (1), (2)



Notes to Figure 75:

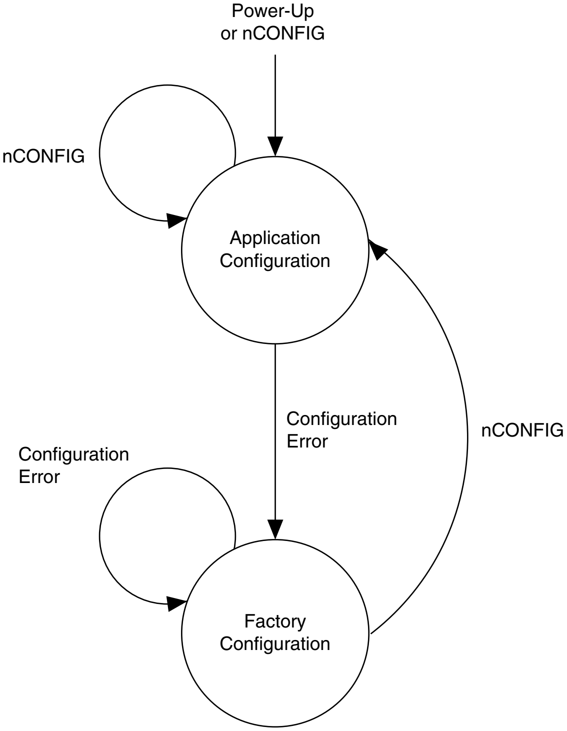
(1) Remote update of Application Configuration is controlled by a Nios embedded processor or user logic programmed in the Factory or Application configurations.

(2) Up to seven pages can be specified allowing up to seven different configuration applications.

Local Update Mode

Local update mode is a simplified version of the remote update. This feature is intended for simple systems that need to load a single application configuration immediately upon power up without loading the factory configuration first. Local update designs have only one application configuration to load, so it does not require a factory configuration to determine which application configuration to use. Figure 76 shows the transition diagram for local update mode.

Figure 76. Local Update Transition Diagram



Temperature Sensing Diode

Stratix devices include a diode-connected transistor for use as a temperature sensor in power management. This diode is used with an external digital thermometer device such as a MAX1617A or MAX1619 from MAXIM Integrated Products. These devices steer bias current through the Stratix diode, measuring forward voltage and converting this reading to temperature in the form of an 8-bit signed number (7 bits plus sign). The external device's output represents the package temperature of the Stratix device and can be used for intelligent power management.

The diode requires two pins (`tempdiodep` and `tempdioden`) on the Stratix device to connect to the external temperature-sensing device, as shown in Figure 77. The temperature sensing diode is a passive element and therefore can be used before the Stratix device is powered. Table 39 shows the specifications for bias voltage and current of the Stratix temperature sensing diode.

Figure 77. External Temperature-Sensing Diode

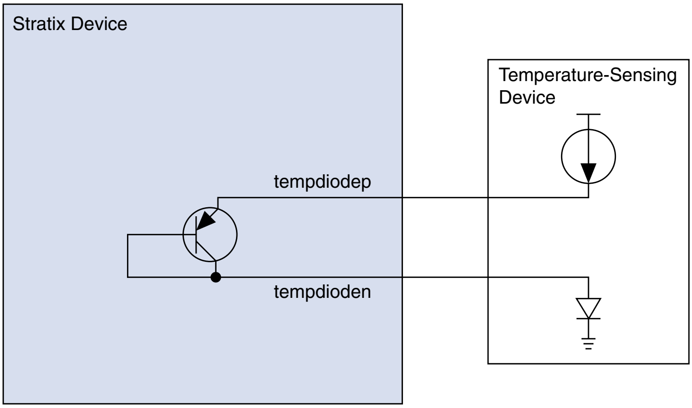
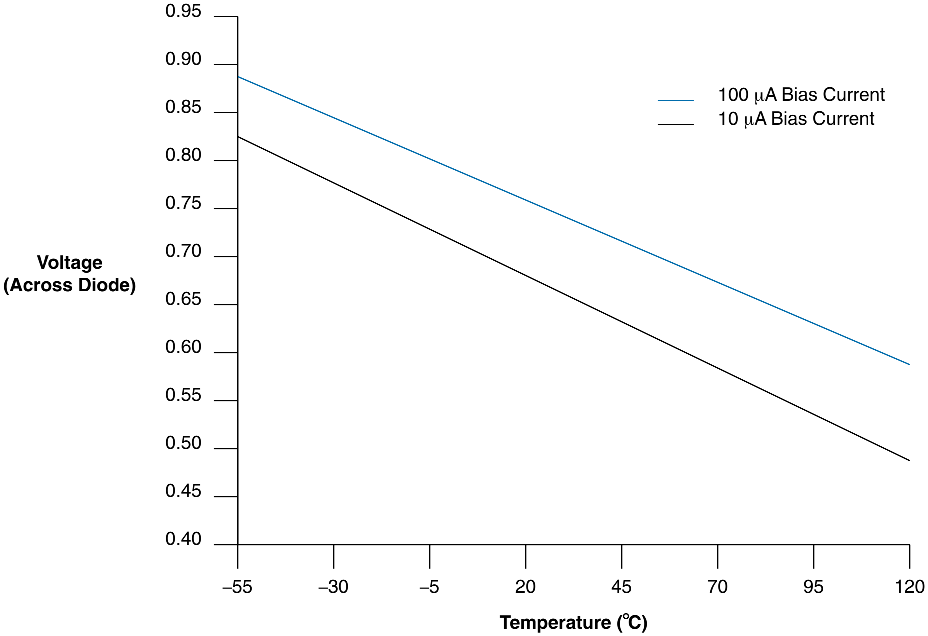


Table 39. Temperature-Sensing Diode Electrical Characteristics				
Parameter	**Minimum**	**Typical**	**Maximum**	**Units**
I_{BIAS} high	80	100	120	μA
I_{BIAS} low	8	10	12	μA
$V_{BP} - V_{BN}$	0.3		0.9	V
V_{BN}		0.7		V
Series resistance			3	Ω

The temperature-sensing diode works for the entire operating range shown in Figure 78.

Figure 78. Temperature vs. Temperature-Sensing Diode Voltage



Operating Conditions

Stratix devices are offered in both commercial and industrial grades. However, industrial-grade devices may have limited speed-grade availability.

Tables 40 through 69 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 1.5-V Stratix devices.

Table 40. Stratix Device Absolute Maximum Ratings *Notes (1), (2)*

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage	With respect to ground *(3)*	–0.5	2.4	V
V_{CCIO}			–0.5	4.6	V
V_I	DC input voltage		–0.5	4.6	V
I_{OUT}	DC output current, per pin		–25	25	mA
T_{STG}	Storage temperature	No bias	–65	150	° C
T_{AMB}	Ambient temperature	Under bias	–65	135	° C
T_J	Junction temperature	BGA packages under bias		135	° C

Table 41. Stratix Device Recommended Operating Conditions

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	*(4)*	1.425	1.575	V
V_{CCIO}	Supply voltage for output buffers, 3.3-V operation	*(4)*, *(5)*	3.00 (3.135)	3.60 (3.465)	V
	Supply voltage for output buffers, 2.5-V operation	*(4)*	2.375	2.625	V
	Supply voltage for output buffers, 1.8-V operation	*(4)*	1.71	1.89	V
	Supply voltage for output buffers, 1.5-V operation	*(4)*	1.4	1.6	V
V_I	Input voltage	*(3)*, *(6)*	–0.5	4.1	V
V_O	Output voltage		0	V_{CCIO}	V
T_J	Operating junction temperature	For commercial use	0	85	° C
		For industrial use	–40	100	° C
t_R	Input rise time			40	ns
t_F	Input fall time			40	ns

Table 42. Stratix Device DC Operating Conditions *Note (7)*

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
I_I	Input pin leakage current	$V_I = V_{CCIOmax}$ to 0 V *(8)*	–10		10	μA
I_{OZ}	Tri-stated I/O pin leakage current	$V_O = V_{CCIOmax}$ to 0 V *(8)*	–10		10	μA
I_{CC0}	V_{CC} supply current (standby) (All ESBs in power-down mode)	V_I = ground, no load, no toggling inputs				mA
R_{CONF}	Value of I/O pin pull-up resistor before and during configuration	V_{CCIO} = 3.0 V *(9)*	20		50	kΩ
		V_{CCIO} = 2.375 V *(9)*	30		80	kΩ
		V_{CCIO} = 1.71 V *(9)*	60		150	kΩ

Table 43. LVTTL Specifications					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Units**
V_{CCIO}	Output supply voltage		3.0	3.6	V
V_{IH}	High-level input voltage		1.7	4.1	V
V_{IL}	Low-level input voltage		–0.5	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –4 to –24 mA *(10)*	2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 4 to 24 mA *(10)*		0.45	V

Table 44. LVCMOS Specifications					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Units**
V_{CCIO}	Output supply voltage		3.0	3.6	V
V_{IH}	High-level input voltage		1.7	4.1	V
V_{IL}	Low-level input voltage		–0.5	0.7	V
V_{OH}	High-level output voltage	V_{CCIO} = 3.0, I_{OH} = –0.1 mA	V_{CCIO} – 0.2		V
V_{OL}	Low-level output voltage	V_{CCIO} = 3.0, I_{OL} = 0.1 mA		0.2	V

Table 45. 2.5-V I/O Specifications *Note (10)*					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Units**
V_{CCIO}	Output supply voltage		2.375	2.625	V
V_{IH}	High-level input voltage		1.7	4.1	V
V_{IL}	Low-level input voltage		–0.5	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –0.1 mA	2.1		V
		I_{OH} = –1 mA	2.0		V
		I_{OH} = –2 to –16 mA *(10)*	1.7		V
V_{OL}	Low-level output voltage	I_{OL} = 0.1 mA		0.2	V
		I_{OH} = 1 mA		0.4	V
		I_{OH} = 2 to 16 mA *(10)*		0.7	V

Table 46. 1.8-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Units
V_{CCIO}	Output supply voltage		1.65	1.95	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	2.25	V
V_{IL}	Low-level input voltage		–0.3	0.35 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 to –8 mA *(10)*	V_{CCIO} – 0.45		V
V_{OL}	Low-level output voltage	I_{OL} = 2 to 8 mA *(10)*		0.45	V

Table 47. 1.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Units
V_{CCIO}	Output supply voltage		1.4	1.6	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3	0.35 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 mA *(10)*	0.75 × V_{CCIO}		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(10)*		0.25 × V_{CCIO}	V

Table 48. 3.3-V LVDS I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{OD}	Differential output voltage	R_L = 100 Ω	250		450	mV
ΔV_{OD}	Change in V_{OD} between high and low	R_L = 100 Ω			50	mV
V_{OS}	Output offset voltage	R_L = 100 Ω	1.125	1.25	1.375	V
ΔV_{OS}	Change in V_{OS} between high and low	R_L = 100 Ω			50	mV
V_{TH}	Differential input threshold	V_{CM} = 1.2 V	–100		100	mV
V_{IN}	Receiver input voltage range		0.0		2.4	V
R_L	Receiver differential input resistor		90	100	110	Ω

Table 49. PCML Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{IL}	Low-level input voltage				V_{CCIO} – 0.3	V
V_{IH}	High-level input voltage		V_{CCIO}			V
V_{OL}	Low-level output voltage				V_{CCIO} – 0.3	V
V_{OH}	High-level output voltage		V_{CCIO}			V
V_T	Output termination voltage			V_{CCIO}		V
V_{OD}	Differential output voltage		300	450	600	mV
V_{TH}	Differential input threshold		100			mV
R_O	Output load			100		Ω
R_L	Receiver pull-up resistor		45	50	55	Ω

Table 50. LVPECL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{IL}	Low-level input voltage		0		2,200	mV
V_{IH}	High-level input voltage		100		2,880	mV
V_{OL}	Low-level output voltage		1,450		1,650	mV
V_{OH}	High-level output voltage		2,175		2,420	mV
V_{ID}	Differential input voltage		100	600	970	mV
V_{OD}	Differential output voltage		525	800	970	mV

Table 51. HyperTransport Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		2.375	2.5	2.625	V
V_{OD}	Differential output voltage		380	600	820	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	500	600	700	mV
V_{ID}	Differential input voltage		300	600	900	mV
V_{ICM}	Input common mode voltage		450	600	750	mV
R_L	Receiver differential input resistor		90	100	110	Ω

Table 52. 3.3-V PCI Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		3.0	3.3	3.6	V
V_{IH}	High-level input voltage		0.5 × V_{CCIO}		V_{CCIO} + 0.5	V
V_{IL}	Low-level input voltage		–0.5		0.3 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OUT} = –500 μA	0.9 × V_{CCIO}			V
V_{OL}	Low-level output voltage	I_{OUT} = 1,500 μA			0.1 × V_{CCIO}	V

Table 53. PCI-X Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		3.0		3.6	V
V_{IH}	High-level input voltage		0.5 × V_{CCIO}		V_{CCIO} + 0.5	V
V_{IL}	Low-level input voltage		–0.5		0.35 × V_{CCIO}	V
V_{IPU}	Input pull-up voltage		0.7 × V_{CCIO}			V
V_{OH}	High-level output voltage	I_{OUT} = –500 μA	0.9 × V_{CCIO}			V
V_{OL}	Low-level output voltage	I_{OUT} = 1,500 μA			0.1 × V_{CCIO}	V

Table 54. GTL+ I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{TT}	Termination voltage		1.35	1.5	1.65	V
V_{REF}	Reference voltage		0.88	1.0	1.12	V
V_{IH}	High-level input voltage		V_{REF} + 0.1			V
V_{IL}	Low-level input voltage				V_{REF} – 0.1	V
V_{OL}	Low-level output voltage	I_{OL} = 36 mA *(10)*			0.65	V

Table 55. GTL I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{TT}	Termination voltage		1.14	1.2	1.26	V
V_{REF}	Reference voltage		0.74	0.8	0.86	V
V_{IH}	High-level input voltage		V_{REF} + 0.05			V
V_{IL}	Low-level input voltage				V_{REF} – 0.05	V
V_{OL}	Low-level output voltage	I_{OL} = 40 mA *(10)*			0.4	V

Table 56. SSTL-18 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		1.65	1.8	1.95	V
V_{REF}	Reference voltage		0.8	0.9	1.0	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.125			V
$V_{IL(DC)}$	Low-level DC input voltage				V_{REF} – 0.125	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.275			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.275	V
V_{OH}	High-level output voltage	I_{OH} = –6.7 mA *(10)*	V_{TT} + 0.475			V
V_{OL}	Low-level output voltage	I_{OL} = 6.7 mA *(10)*			V_{TT} – 0.475	V

Table 57. SSTL-18 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		1.65	1.8	1.95	V
V_{REF}	Reference voltage		0.8	0.9	1.0	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.125			V
$V_{IL(DC)}$	Low-level DC input voltage				V_{REF} – 0.125	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.275			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.275	V
V_{OH}	High-level output voltage	I_{OH} = –13.4 mA *(10)*	V_{TT} + 0.630			V
V_{OL}	Low-level output voltage	I_{OL} = 13.4 mA *(10)*			V_{TT} – 0.630	V

Table 58. SSTL-2 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		2.375	2.5	2.625	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage		1.15	1.25	1.35	V
V_{IH}	High-level input voltage		V_{REF} + 0.18		3.0	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.18	V
V_{OH}	High-level output voltage	I_{OH} = –8.1 mA *(10)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(10)*			V_{TT} – 0.57	V

Table 59. SSTL-2 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		2.3	2.5	2.7	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage		1.15	1.25	1.35	V
V_{IH}	High-level input voltage		V_{REF} + 0.18		V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.18	V
V_{OH}	High-level output voltage	I_{OH} = –16.4 mA *(10)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(10)*			V_{TT} – 0.76	V

Table 60. SSTL-3 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		3.0	3.3	3.6	V
V_{TT}	Termination voltage		V_{REF} – 0.05	V_{REF}	V_{REF} + 0.05	V
V_{REF}	Reference voltage		1.3	1.5	1.7	V
V_{IH}	High-level input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA *(10)*	V_{TT} + 0.6			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA *(10)*			V_{TT} – 0.6	V

Table 61. SSTL-3 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		3.0	3.3	3.6	V
V_{TT}	Termination voltage		V_{REF} – 0.05	V_{REF}	V_{REF} + 0.05	V
V_{REF}	Reference voltage		1.3	1.5	1.7	V
V_{IH}	High-level input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = –16 mA *(10)*	V_{TT} + 0.8			V
V_{OL}	Low-level output voltage	I_{OL} = 16 mA *(10)*			V_{TT} – 0.8	V

Table 62. 3.3-V AGP 2× Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		3.15	3.3	3.45	V
V_{REF}	Reference voltage		$0.39 \times V_{CCIO}$		$0.41 \times V_{CCIO}$	V
V_{IH}	High-level input voltage *(11)*		$0.5 \times V_{CCIO}$		$V_{CCIO} + 0.5$	V
V_{IL}	Low-level input voltage *(11)*				$0.3 \times V_{CCIO}$	V
V_{OH}	High-level output voltage	$I_{OUT} = -0.5$ mA	$0.9 \times V_{CCIO}$		3.6	V
V_{OL}	Low-level output voltage	$I_{OUT} = 1.5$ mA			$0.1 \times V_{CCIO}$	V

Table 63. 3.3-V AGP 1× Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		3.15	3.3	3.45	V
V_{IH}	High-level input voltage *(11)*		$0.5 \times V_{CCIO}$		$V_{CCIO} + 0.5$	V
V_{IL}	Low-level input voltage *(11)*				$0.3 \times V_{CCIO}$	V
V_{OH}	High-level output voltage	$I_{OUT} = -0.5$ mA	$0.9 \times V_{CCIO}$		3.6	V
V_{OL}	Low-level output voltage	$I_{OUT} = 1.5$ mA			$0.1 \times V_{CCIO}$	V

Table 64. 1.5-V HSTL Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		1.4	1.5	1.6	V
V_{REF}	Input reference voltage		0.68	0.75	0.9	V
V_{TT}	Termination voltage		0.7	0.75	0.8	V
V_{IH} (DC)	DC high-level input voltage		$V_{REF} + 0.1$			V
V_{IL} (DC)	DC low-level input voltage		–0.3		$V_{REF} - 0.1$	V
V_{IH} (AC)	AC high-level input voltage		$V_{REF} + 0.2$			V
V_{IL} (AC)	AC low-level input voltage				$V_{REF} - 0.2$	V
V_{OH}	High-level output voltage	$I_{OH} = 8$ mA *(10)*	$V_{CCIO} - 0.4$			V
V_{OL}	Low-level output voltage	$I_{OH} = -8$ mA *(10)*			0.4	V

Table 65. 1.5-V HSTL Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		1.4	1.5	1.6	V
V_{REF}	Input reference voltage		0.68	0.75	0.9	V
V_{TT}	Termination voltage		0.7	0.75	0.8	V
V_{IH} (DC)	DC high-level input voltage		$V_{REF} + 0.1$			V
V_{IL} (DC)	DC low-level input voltage		–0.3		$V_{REF} - 0.1$	V
V_{IH} (AC)	AC high-level input voltage		$V_{REF} + 0.2$			V
V_{IL} (AC)	AC low-level input voltage				$V_{REF} - 0.2$	V
V_{OH}	High-level output voltage	I_{OH} = 16 mA *(10)*	$V_{CCIO} - 0.4$			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(10)*			0.4	V

Table 66. 1.5-V Differential HSTL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		1.4	1.5	1.6	V
V_{DIF} (DC)	DC input differential voltage		0.2			V
V_{CM} (DC)	DC common mode input voltage		0.68		0.9	V
V_{DIF} (AC)	AC differential input voltage		0.4			V

Table 67. CTT I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	Output supply voltage		3.0	3.3	3.6	V
V_{TT}/V_{REF}	Termination and input reference voltage		1.35	1.5	1.65	V
V_{IH}	High-level input voltage		$V_{REF} + 0.2$			V
V_{IL}	Low-level input voltage				$V_{REF} - 0.2$	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA	$V_{REF} + 0.4$			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA			$V_{REF} - 0.4$	V
I_O	Output leakage current (when output is high *Z*)	GND ≤ V_{OUT} ≤ V_{CCIO}	–10		10	µA

Table 68. Bus Hold Parameters

Parameter	Conditions	V_{CCIO} Level								Units
		1.5 V		1.8 V		2.5 V		3.3 V		
		Min	Max	Min	Max	Min	Max	Min	Max	
Low sustaining current	$V_{IN} > V_{IL}$ (maximum)			30		50		70		μA
High sustaining current	$V_{IN} < V_{IH}$ (minimum)			–30		–50		–70		μA
Low overdrive current	$0 V < V_{IN} < V_{CCIO}$				200		300		500	μA
High overdrive current	$0 V < V_{IN} < V_{CCIO}$				–200		–300		–500	μA

Table 69. Stratix Device Capacitance

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
C_{IN}	Input capacitance on I/O pins in banks 1, 2, 5, and 6	$V_{IN} = 0$ V, f = 1.0 MHz			pF
	Input capacitance on I/O pins in banks 3, 4, 7, and 8	$V_{IN} = 1.0$ V, f = 1.0 MHz			
C_{INCLK}	Input capacitance on dedicated clock pin	$V_{IN} = 0$ V, f = 1.0 MHz		12	pF
C_{OUT}	Output capacitance	$V_{IN} = 0$ V, f = 1.0 MHz		10	pF

Notes to Tables 40 – 69:

(1) See the *Operating Requirements for Altera Devices Data Sheet*.
(2) Conditions beyond those listed in Table 40 may cause permanent damage to a device. Additionally, device operation at the absolute maximum ratings for extended periods of time may have adverse affects on the device.
(3) Minimum DC input is –0.5 V. During transitions, the inputs may undershoot to –0.5 V or overshoot to 4.6 V for input currents less than 100 mA and periods shorter than 20 ns.
(4) Maximum V_{CC} rise time is 100 ms, and V_{CC} must rise monotonically.
(5) V_{CCIO} maximum and minimum conditions for LVPECL, LVDS, and PCML are shown in parentheses.
(6) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT} and V_{CCIO} are powered.
(7) Typical values are for $T_A = 25°$ C, $V_{CCINT} = 1.5$ V, and $V_{CCIO} = 1.5$ V, 1.8 V, 2.5 V, and 3.3 V.
(8) This value is specified for normal device operation. The value may vary during power-up. This applies for all V_{CCIO} settings (3.3, 2.5, 1.8, and 1.5 V)
(9) Pin pull-up resistance values will lower if an external source drives the pin higher than V_{CCIO}.
(10) Drive strength is programmable according to values in Table 27 on page 116.
(11) V_{REF} specifies the center point of the switching range.

Power Consumption

Detailed power consumption information for Stratix devices will be released when available.

Timing Model

The DirectDrive technology and MultiTrack interconnect ensure predictable performance, accurate simulation, and accurate timing analysis across all Stratix device densities and speed grades. This section describes and specifies the performance, internal, external, and PLL timing specifications.

All specifications are representative of worst-case supply voltage and junction temperature conditions.

Preliminary & Final Timing

Timing models can have either preliminary or final status. The Quartus II software issues an informational message during the design compilation if the timing models are preliminary. Table 70 shows the status of the Stratix device timing models.

Preliminary status means the timing model is subject to change. Initially, timing numbers are created using simulation results, process data, and other known parameters. These tests are used to make the preliminary numbers as close to the actual timing parameters as possible.

Final timing numbers are based on actual device operation and testing. These numbers reflect the actual performance of the device under worst-case voltage and junction temperature conditions.

Table 70. Stratix Device Timing Model Status

Device	Preliminary	Final
EP1S10	✓	
EP1S20	✓	
EP1S25	✓	
EP1S30	✓	
EP1S40	✓	
EP1S60	✓	
EP1S80	✓	
EP1S120	✓	

Performance

Table 71 shows Stratix performance for some common designs. All performance values were obtained with Quartus II software compilation of LPM, or MegaCore functions for the FIR and FFT designs.

Table 71. Stratix Performance (Part 1 of 2) *Notes (1), (2)*

Applications		Resources Used			Performance			
		LEs	TriMatrix Memory Blocks	DSP Blocks	-5 Speed Grade	-6 Speed Grade	-7 Speed Grade	Units
LE	16-to-1 multiplexer *(3)*							ns
	32-to-1 multiplexer *(3)*							ns
	16-to-1 multiplexer *(4)*							MHz
	32-to-1 multiplexer *(4)*							MHz
	16-bit counter							MHz
	64-bit counter							MHz
TriMatrix memory M512 block	Simple dual-port RAM 32 × 18 bit							MHz
	FIFO 32 × 18 bit							MHz
TriMatrix memory M4K block	Simple dual-port RAM 128 × 36 bit							MHz
	True dual-port RAM 128 × 36 bit							MHz
	FIFO 128 × 36 bit							MHz
TriMatrix memory MegaRAM block	Simple dual-port RAM 4K × 144 bit							MHz
	True dual-port RAM 8K × 72 bit							MHz

Table 71. Stratix Performance (Part 2 of 2) *Notes (1), (2)*

Applications		Resources Used			Performance			
		LEs	TriMatrix Memory Blocks	DSP Blocks	-5 Speed Grade	-6 Speed Grade	-7 Speed Grade	Units
DSP block	9 × 9-bit multiplier *(5)*	0	0	1		281		MHz
	18 × 18-bit multiplier *(5)*	0	0	1		232		MHz
	36 × 36-bit multiplier *(5)*	0	0	1		135		MHz
	36 × 36-bit multiplier *(6)*	0	0	1		232		MHz
	18-bit, 4-tap FIR filter	0	0	1				
Larger designs	8-bit, 16-tap parallel FIR filter							MHz
	8-bit, 512-point FFT function							MHz

Notes to Table 71:
(1) These design performance numbers were obtained using the Quartus II software.
(2) Numbers not listed will be included in a future version of this data sheet.
(3) This performance number reflects the combinatorial input and output path without pins.
(4) This application uses registered inputs and outputs.
(5) This application uses registered input and output stages within the DSP block.
(6) This application uses registered input, pipeline, and output stages within the DSP block.

Internal Timing Parameters

Internal timing parameters are specified on a speed grade basis independent of device density. Tables 72 through 78 describe the Stratix device internal timing microparameters for LEs, IOEs, TriMatrix memory structures, DSP blocks, and MultiTrack interconnects.

Table 72. LE Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{SU}	LE register setup time before clock
t_{H}	LE register hold time after clock
t_{CO}	LE register clock-to-output delay
t_{LUT}	LE combinatorial LUT delay for data-in to data-out
t_{CLR}	Minimum clear pulse width
t_{PRE}	Minimum preset pulse width
t_{CLKHL}	Minimum clock high or low time

Table 73. IOE Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{SU}	IOE input and output register setup time before clock
t_{H}	IOE input and output register hold time after clock
t_{CO}	IOE input and output register clock-to-output delay
t_{COOUT}	IOE output register clock-to-output delay
$t_{PIN2COMBOUT}$	Input pin to IOE combinatorial output
$t_{COMBIN2PIN}$	IOE data input to combinatorial output
t_{CLR}	Minimum clear pulse width
t_{PRE}	Minimum preset pulse width
t_{CLKHL}	Minimum clock high or low time

Table 74. DSP Block Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{SU}	Input, pipeline, and output register setup time before clock
t_{H}	Input, pipeline, and output register hold time after clock
t_{CO}	Input, pipeline, and output register clock-to-output delay
$t_{INREG2PIPE18}$	Input Register to DSP Block pipeline register in 18 × 18-bit mode
$t_{INREG2PIPE9}$	Input Register to DSP Block pipeline register in 9 × 9-bit mode
$t_{PIPE2OUTREG2ADD}$	DSP Block Pipeline Register to output register delay in Two-Multipliers Adder mode
$t_{PIPE2OUTREG4ADD}$	DSP Block Pipeline Register to output register delay in Four-Multipliers Adder mode
t_{PD9}	Combinatorial input to output delay for 9 × 9
t_{PD18}	Combinatorial input to output delay for 18 × 18
t_{PD36}	Combinatorial input to output delay for 36 × 36
t_{CLR}	Minimum clear pulse width
t_{PRE}	Minimum preset pulse width
t_{CLKHL}	Minimum clock high or low time

Table 75. M512 Block Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{M512RC}	Synchronous read cycle time
t_{M512WC}	Synchronous write cycle time
$t_{M512WERESU}$	Write or read enable setup time before clock
$t_{M512WEREH}$	Write or read enable hold time after clock
$t_{M512DATASU}$	Data setup time before clock
$t_{M512DATAH}$	Data hold time after clock
$t_{M512WADDRSU}$	Write address setup time before clock
$t_{M512WADDRH}$	Write address hold time after clock
$t_{M512RADDRSU}$	Read address setup time before clock
$t_{M512RADDRH}$	Read address hold time after clock
$t_{M512DATACO1}$	Clock-to-output delay when using output registers
$t_{M512DATACO2}$	Clock-to-output delay without output registers
$t_{M512CLKHL}$	Minimum clock high or low time
$t_{M512CLR}$	Minimum clear pulse width
$t_{M512PRE}$	Minimum preset pulse width

Table 76. M4K Block Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{M4KRC}	Synchronous read cycle time
t_{M4KWC}	Synchronous write cycle time
$t_{M4KWERESU}$	Write or read enable setup time before clock
$t_{M4KWEREH}$	Write or read enable hold time after clock
$t_{M4KBESU}$	Byte enable setup time before clock
t_{M4KBEH}	Byte enable hold time after clock
$t_{M4KDATASU}$	Data setup time before clock
$t_{M4KDATAH}$	Data hold time after clock
$t_{M4KWADDRSU}$	Write address setup time before clock
$t_{M4KWADDRH}$	Write address hold time after clock
$t_{M4KRADDRSU}$	Read address setup time before clock
$t_{M4KRADDRH}$	Read address hold time after clock
$t_{M4KDATACO1}$	Clock-to-output delay when using output registers
$t_{M4KDATACO2}$	Clock-to-output delay without output registers
$t_{M4KCLKHL}$	Minimum clock high or low time
t_{M4KCLR}	Minimum clear pulse width
t_{M4KPRE}	Minimum preset pulse width

Table 77. MegaRAM Block Internal Timing Microparameter Descriptions

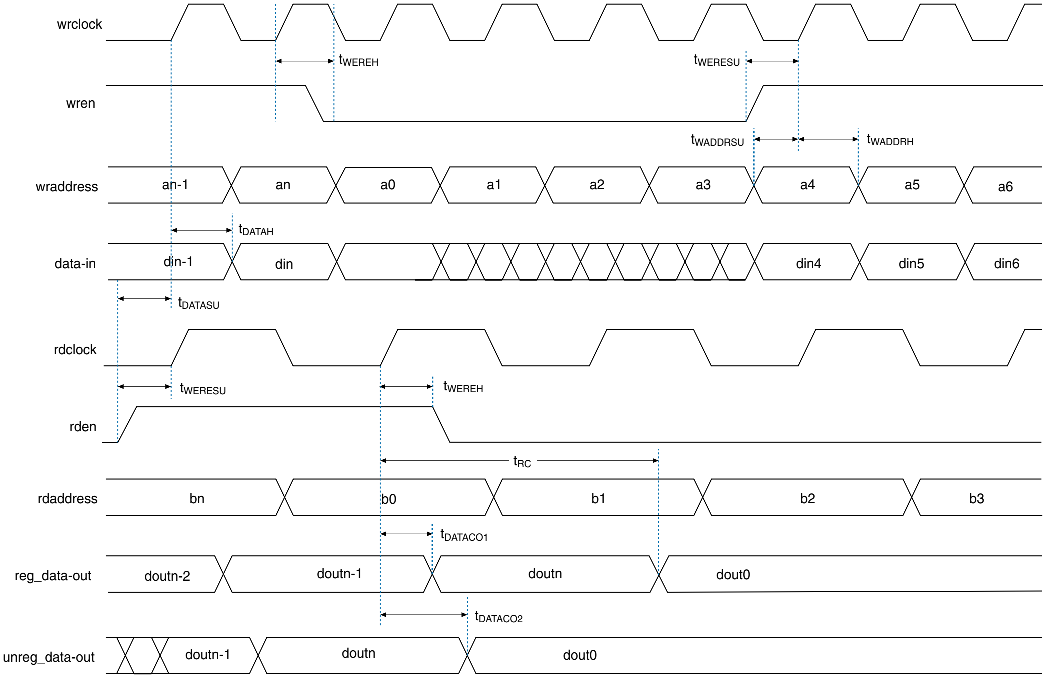
Symbol	Parameter
t_{MEGARC}	Synchronous read cycle time
t_{MEGAWC}	Synchronous write cycle time
$t_{MEGAWERESU}$	Write or read enable setup time before clock
$t_{MEGAWEREH}$	Write or read enable hold time after clock
$t_{MEGABESU}$	Byte enable setup time before clock
$t_{MEGABEH}$	Byte enable hold time after clock
$t_{MEGADATASU}$	Data setup time before clock
$t_{MEGADATAH}$	Data hold time after clock
$t_{MEGAWADDRSU}$	Write address setup time before clock
$t_{MEGAWADDRH}$	Write address hold time after clock
$t_{MEGARADDRSU}$	Read address setup time before clock
$t_{MEGARADDRH}$	Read address hold time after clock
$t_{MEGADATACO1}$	Clock-to-output delay when using output registers
$t_{MEGADATACO2}$	Clock-to-output delay without output registers
$t_{MEGACLKHL}$	Minimum clock high or low time
$t_{MEGACLR}$	Minimum clear pulse width
$t_{MEGAPRE}$	Minimum preset pulse width

Table 78. Routing Delay Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{R4}	Delay for an R4 line with one load; covers a distance of four LAB columns
t_{R8}	Delay for an R8 line with one load; covers a distance of eight LAB columns
t_{R24}	Delay for an R24 line with one load; covers a distance of 24 LAB columns
t_{C4}	Delay for an C4 line with one load; covers a distance of four LAB rows
t_{C8}	Delay for an C8 line with one load; covers a distance of eight LAB rows
t_{C16}	Delay for an C24 line with one load; covers a distance of 16 LAB rows
t_{LOCAL}	Local interconnect delay

Figure 79 shows the TriMatrix memory waveforms for the M512, M4K, and MegaRAM timing parameters shown in Tables 75 through 77 above.

Figure 79. Dual-Port RAM Timing Microparameter Waveform



Tables 79 through 85 show the internal timing microparameters for all Stratix devices.

Table 79. LE Internal Timing Microparameters

Symbol	-5		-6		-7	
	Min	Max	Min	Max	Min	Max
t_{SU}						
t_H						
t_{CO}						
t_{LUT}						
t_{CLR}						
t_{PRE}						
t_{CLKHL}						

Table 80. IOE Internal Timing Microparameters

Symbol	-5		-6		-7	
	Min	Max	Min	Max	Min	Max
t_{SU}						
t_{H}						
t_{CO}						
t_{COOUT}						
$t_{PIN2COMBOUT}$						
$t_{COMBIN2PIN}$						
t_{CLR}						
t_{PRE}						
t_{CLKHL}						

Table 81. DSP Block Internal Timing Microparameters

Symbol	-5		-6		-7	
	Min	Max	Min	Max	Min	Max
t_{SU}						
t_{H}						
t_{CO}						
$t_{INREG2PIPE18}$						
$t_{INREG2PIPE9}$						
$t_{PIPE2OUTREG2ADD}$						
$t_{PIPE2OUTREG4ADD}$						
t_{PD9}						
t_{PD18}						
t_{PD36}						
t_{CLR}						
t_{PRE}						
t_{CLKHL}						

Table 82. M512 Block Internal Timing Microparameters

Symbol	-5		-6		-7	
	Min	Max	Min	Max	Min	Max
$t_{M512WESU}$						
$t_{M512RESU}$						
$t_{M512BESU}$						
$t_{M512DATASU}$						
$t_{M512RDADDRSU}$						
$t_{M512WRADDRSU}$						
$t_{M512DATACO1}$						
$t_{M512DATACO2}$						
$t_{M512CLKHL}$						
$t_{M512CLR}$						
$t_{M512PRE}$						

Table 83. M4K Block Internal Timing Microparameters

Symbol	-5		-6		-7	
	Min	Max	Min	Max	Min	Max
$t_{M4KWESU}$						
$t_{M4KRESU}$						
$t_{M4KBESU}$						
$t_{M4KDATASU}$						
$t_{M4KRDADDRSU}$						
$t_{M4KWRADDRSU}$						
$t_{M4KDATACO1}$						
$t_{M4KDATACO2}$						
$t_{M4KCLKHL}$						
t_{M4KCLR}						
t_{M4KPRE}						

Table 84. MegaRAM Block Internal Timing Microparameters

Symbol	-5		-6		-7	
	Min	Max	Min	Max	Min	Max
$t_{MEGAWESU}$						
$t_{MEGARESU}$						
$t_{MEGABESU}$						
$t_{MEGADATASU}$						
$t_{MEGARDADDRSU}$						
$t_{MEGAWRADDRSU}$						
$t_{MEGADATACO1}$						
$t_{MEGADATACO2}$						
$t_{MEGACLKHL}$						
$t_{MEGACLR}$						
$t_{MEGAPRE}$						

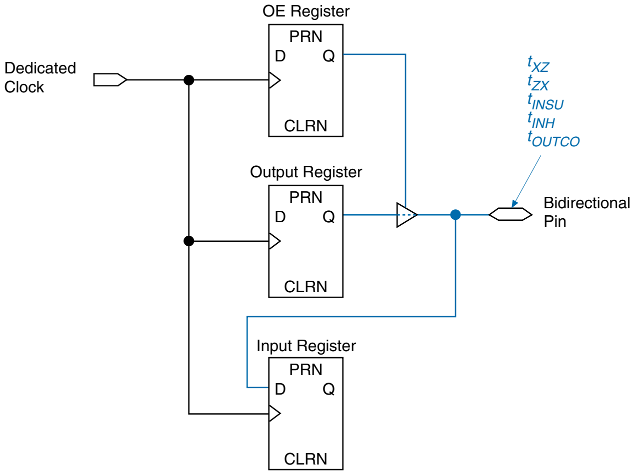
Table 85. Routing Delay Internal Timing Microparameters

Symbol	-5		-6		-7	
	Min	Max	Min	Max	Min	Max
t_{R4}						
t_{R8}						
t_{R24}						
t_{C4}						
t_{C8}						
t_{C16}						
t_{LOCAL}						

External Timing Parameters

External timing parameters are specified by device density and speed grade. Figure 80 shows the timing model for bidirectional IOE pin timing. All registers are within the IOE.

Figure 80. External Timing in Stratix Devices



All external I/O timing parameters shown are for 3.3-V LVTTL or LVCMOS I/O standards with the maximum current strength. For external I/O timing using standards other than LVTTL or LVCMOS use the I/O standard input and output delay adders in Tables 113 through 115.

Table 86 shows the external I/O timing parameters when using fast regional clock networks.

Table 86. Stratix Fast Regional Clock External I/O Timing Parameters *Notes (1), (2)*

Symbol	Parameter	Conditions
t_{INSU}	Setup time for input or bidirectional pin using column IOE input register with fast regional clock fed by `FCLK` pin	
t_{INH}	Hold time for input or bidirectional pin using column IOE input register with fast regional clock fed by `FCLK` pin	
t_{OUTCO}	Clock-to-output delay output or bidirectional pin using column IOE output register with fast regional clock fed by `FCLK` pin	
t_{XZ}	Synchronous column IOE output enable register to output pin disable delay using fast regional clock fed by `FCLK` pin	
t_{ZX}	Synchronous column IOE output enable register to output pin enable delay using fast regional clock fed by `FCLK` pin	

Notes to Table 86:

(1) These timing parameters are sample-tested only.

(2) These timing parameters are for column IOE pins. Row IOE pins are 100- to 250-ps slower depending on device and speed grade and whether it is t_{CO} or t_{SU}. Designers should use the Quartus II software to verify the external timing for any pin.

Table 87 shows the external I/O timing parameters when using regional clock networks.

Table 87. Stratix Regional Clock External I/O Timing Parameters Notes (1), (2)

Symbol	Parameter	Conditions
t_{INSU}	Setup time for input or bidirectional pin using column IOE input register with regional clock fed by `CLK` pin	
t_{INH}	Hold time for input or bidirectional pin using column IOE input register with regional clock fed by `CLK` pin	
t_{OUTCO}	Clock-to-output delay output or bidirectional pin using column IOE output register with regional clock fed by `CLK` pin	
t_{XZ}	Synchronous column IOE output enable register to output pin disable delay using regional clock fed by `CLK` pin	
t_{ZX}	Synchronous column IOE output enable register to output pin enable delay using regional clock fed by `CLK` pin	
$t_{INSUPLL}$	Setup time for input or bidirectional pin using column IOE input register with regional clock fed by Enhanced PLL with default phase setting	
t_{INHPLL}	Hold time for input or bidirectional pin using column IOE input register with regional clock fed by Enhanced PLL with default phase setting	
$t_{OUTCOPLL}$	Clock-to-output delay output or bidirectional pin using column IOE output register with regional clock Enhanced PLL with default phase setting	
t_{XZPLL}	Synchronous column IOE output enable register to output pin disable delay using regional clock fed by Enhanced PLL with default phase setting	
t_{ZXPLL}	Synchronous column IOE output enable register to output pin enable delay using regional clock fed by Enhanced PLL with default phase setting	

Notes to Table 87:

(1) These timing parameters are sample-tested only.

(2) These timing parameters are for column IOE pins. Row IOE pins are 100- to 250-ps slower depending on device, speed grade, and the specific parameter in question. Designers should use the Quartus II software to verify the external timing for any pin.

Table 88 shows the external I/O timing parameters when using global clock networks.

Table 88. Stratix Global Clock External I/O Timing Parameters *Notes (1), (2)*

Symbol	Parameter	Conditions
t_{INSU}	Setup time for input or bidirectional pin using column IOE input register with global clock fed by `CLK` pin	
t_{INH}	Hold time for input or bidirectional pin using column IOE input register with global clock fed by `CLK` pin	
t_{OUTCO}	Clock-to-output delay output or bidirectional pin using column IOE output register with global clock fed by `CLK` pin	
t_{XZ}	Synchronous column IOE output enable register to output pin disable delay using global clock fed by `CLK` pin	
t_{ZX}	Synchronous column IOE output enable register to output pin enable delay using global clock fed by `CLK` pin	
$t_{INSUPLL}$	Setup time for input or bidirectional pin using column IOE input register with global clock fed by Enhanced PLL with default phase setting	
t_{INHPLL}	Hold time for input or bidirectional pin using column IOE input register with global clock fed by enhanced PLL with default phase setting	
$t_{OUTCOPLL}$	Clock-to-output delay output or bidirectional pin using column IOE output register with global clock enhanced PLL with default phase setting	
t_{XZPLL}	Synchronous column IOE output enable register to output pin disable delay using global clock fed by enhanced PLL with default phase setting	
t_{ZXPLL}	Synchronous column IOE output enable register to output pin enable delay using global clock fed by enhanced PLL with default phase setting	

Notes to Table 88:
(1) These timing parameters are sample-tested only.
(2) These timing parameters are for column IOE pins. Row IOE pins are 100- to 250-ps slower depending on device, speed grade, and the specific parameter in question. Designers should use the Quartus II software to verify the external timing for any pin.

Tables 89 through 91 show the external timing parameters for EP1S10 devices.

Table 89. EP1S10 Fast Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 90. EP1S10 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 91. EP1S10 Global Clock External I/O Timing Parameters							
Symbol	**-5 Speed Grade**		**-6 Speed Grade**		**-7 Speed Grade**		**Unit**
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Tables 92 through 94 show the external timing parameters for EP1S20 devices.

Table 92. EP1S20 Fast Regional Clock External I/O Timing Parameters							
Symbol	**-5 Speed Grade**		**-6 Speed Grade**		**-7 Speed Grade**		**Unit**
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 93. EP1S20 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 94. EP1S20 Global Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Tables 95 through 97 show the external timing parameters for EP1S25 devices.

Table 95. EP1S25 Fast Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 96. EP1S25 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 97. EP1S25 Global Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Tables 98 through 100 show the external timing parameters for EP1S30 devices.

Table 98. EP1S30 Fast Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 99. EP1S30 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 100. EP1S30 Global Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Tables 101 through 103 show the external timing parameters for EP1S40 devices.

Table 101. EP1S40 Fast Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 102. EP1S40 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 103. EP1S40 Global Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Tables 104 through 106 show the external timing parameters for EP1S60 devices.

Table 104. EP1S60 Fast Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 105. EP1S60 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 106. EP1S60 Global Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Tables 107 through 109 show the external timing parameters for EP1S80 devices.

Table 107. EP1S80 Fast Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 108. EP1S80 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 109. EP1S80 Global Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Tables 110 through 112 show the external timing parameters for EP1S120 devices.

Table 110. EP1S120 Fast Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							

Table 111. EP1S120 Regional Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

Table 112. EP1S120 Global Clock External I/O Timing Parameters

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}							
t_{INH}							
t_{OUTCO}							
t_{XZ}							
t_{ZX}							
$t_{INSUPLL}$							
t_{INHPLL}							
$t_{OUTCOPLL}$							
t_{XZPLL}							
t_{ZXPLL}							

External Delay Parameters

External delay parameters for I/O standard input and output adders and programmable input and output delays are specified by speed grade independent of device density.

Tables 113 through 115 show the adder delays associated with I/O standards. If you select an I/O standard other than LVTTL/LVCMOS, add the selected delay to the external t_{CO} and t_{SU} I/O parameters shown in Tables 79 through 85.

Table 113. Stratix I/O Standard Input Delay Adders

I/O Standard	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
LVCMOS							
3.3-V LVTTL							
2.5-V LVTTL							
1.8-V LVTTL							
1.5-V LVTTL							
GTL LVTTL							
GTL+							
3.3-V PCI							
3.3-V PCI-X							
Compact PCI							
AGP 1×							
AGP 2×							
CTT							
SSTL-3 class I							
SSTL-3 class II							
SSTL-2 class I							
SSTL-2 class II							
SSTL-18 class I							
SSTL-18 class II							
HSTL class I							
HSTL class II							
LVDS							
LVPECL							
PCML							
HyperTransport							

Table 114. Stratix I/O Standard Output Delay Adders for Fast Slew Rate (Part 1 of 2)

I/O Standard		-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
LVCMOS	2 mA							
	4 mA							
	8 mA							
	12 mA							
	24 mA							
3.3-V LVTTL	4 mA							
	8 mA							
	12 mA							
	16 mA							
	24 mA							
2.5-V LVTTL	2 mA							
	8 mA							
	12 mA							
	16 mA							
1.8-V LVTTL	2 mA							
	8 mA							
	12 mA							
1.5-V LVTTL	2 mA							
	4 mA							
	8 mA							
GTL								
GTL+								
3.3-V PCI								
3.3-V PCI-X								
Compact PCI								
AGP 1×								
AGP 2×								
CTT								
SSTL-3 class I								
SSTL-3 class II								
SSTL-2 class I								
SSTL-2 class II								
SSTL-18 class I								
SSTL-18 class II								
HSTL class I								

Table 114. Stratix I/O Standard Output Delay Adders for Fast Slew Rate (Part 2 of 2)

I/O Standard	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
HSTL class II							
LVDS							
LVPECL							
PCML							
HyperTransport							

Table 115. Stratix I/O Standard Output Delay Adders for Slow Slew Rate (Part 1 of 2)

I/O Standard		-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
LVCMOS	2 mA							
	4 mA							
	8 mA							
	12 mA							
	24 mA							
3.3-V LVTTL	4 mA							
	8 mA							
	12 mA							
	16 mA							
	24 mA							
2.5-V LVTTL	2 mA							
	8 mA							
	12 mA							
	16 mA							
1.8-V LVTTL	2 mA							
	8 mA							
	12 mA							
1.5-V LVTTL	2 mA							
	4 mA							
	8 mA							
GTL								
GTL+								
3.3-V PCI								
3.3-V PCI-X								
Compact PCI								

Table 115. Stratix I/O Standard Output Delay Adders for Slow Slew Rate (Part 2 of 2)

I/O Standard	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
AGP 1×							
AGP 2×							
CTT							
SSTL-3 class I							
SSTL-3 class II							
SSTL-2 class I							
SSTL-2 class II							
SSTL-18 class I							
SSTL-18 class II							
HSTL class I							
HSTL class II							
LVDS							
LVPECL							
PCML							
HyperTransport							

Table 116 shows the adder delays for the IOE programmable delays. These delays are controlled with the Quartus II software options listed in the Parameter column.

Table 116. Stratix IOE Programmable Delays

Parameter	Setting	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
Decrease input delay to internal cells	On							
	Small							
	Medium							
	Large							
Decrease input delay to input register	On							
Decrease input delay to output register	On							
Increase delay to output pin	On							
Increase delay to output enable pin	On							
Increase output clock enable delay	On							
Increase input clock enable delay	On							
Increase t_{ZX} delay to output pin	On							
Increase output enable clock enable delay	On							

PLL Timing

Table 117 describes the Stratix device enhanced PLL specifications.

Table 117. Enhanced PLL Specifications

Symbol	Parameter	Min	Typ	Max	Unit
f_{IN}	Input frequency *(1)*	3		462	MHz
$f_{IN\ DUTY}$	Input clock duty cycle	40		60	%
$f_{EIN\ DUTY}$	External feedback clock input duty cycle	40		60	%
$t_{IN\ JITTER}$	Input clock cycle-to-cycle jitter			±150	ps
$t_{EIN\ JITTER}$	External feedback clock cycle-to-cycle jitter			±150	ps
t_{FCOMP}	External feedback clock compensation time *(2)*			6	ns
f_{OUT}	PLL output frequency	0.6		462	MHz
$t_{OUT\ DUTY}$	Duty cycle for external clock output (when set to 50%)	45		55	%
t_{JITTER}	PLL external clock cycle-to-cycle output jitter *(3)*			±100	ps
t_{DLOCK}	Time required to lock dynamically (after switchover or counter value change)	*(4)*		100	μs
t_{LOCK}	Time required to lock from end of device configuration	10		1,000	μs
f_{VCO}	PLL internal VCO operating range	300		800	MHz
t_{LSKEW}	Clock skew between two external clock outputs driven by the same counter		±50		ps
t_{SKEW}	Clock skew between two external clock outputs driven by the different counters with the same settings		±75		ps
f_{SS}	Spread spectrum modulation frequency	150		500	kHz
% spread	Percentage spread for spread spectrum frequency *(5)*		0.5		%

Notes to Table 117:

(1) Actual PLL performance depends on the settings made.
(2) t_{FCOMP} can also equal 50% of the input clock period multiplied by n.
(3) Actual jitter performance may vary based on the system configuration.
(4) Lock time is a function of PLL configuration and may be significantly faster.
(5) Exact, user-controllable value depends on the PLL settings.

Table 118 describes the Stratix device fast PLL specifications.

Table 118. Fast PLL Specifications

Symbol	Parameter	Min	Max	Unit
f_{IN} (1)	`CLKIN` frequency (for *m* = 9, 10)	33	84	MHz
	`CLKIN` frequency (for *m* = 7, 8)	43	105	MHz
	`CLKIN` frequency (for *m* = 4, 5, 6)	75	140	MHz
	`CLKIN` frequency (for *m* = 2)	150	420	MHz
	`CLKIN` frequency (for *m* = 1)	300	644.5	MHz
t_{INDUTY}	`CLKIN` duty cycle	40	60	%
$t_{INJITTER}$	Cycle-to-cycle jitter for `CLKIN` pin		±100	ps
t_{DUTY}	Duty cycle for `DFFIO` 1× `CLKOUT` pin	40	60	%
t_{JITTER}	Cycle-to-cycle jitter for DIFFIO clock out(2)		±100	ps
	Cycle-to-cycle jitter for internal global or regional clock		±100	ps
t_{LOCK}	Time required for PLL to acquire lock	10	100	µs
f_{VCO}	VCO operating frequency	300	840	MHz
m (3)	Multiplication factors for *m* counter	1	32	Integer
*l*0, *l*1, *g*0 (3)	Multiplication factors for *l*0, *l*1, and *g*0 counter	1	32	Integer

Notes to Table 118:

(1) PLLs 7, 8, 9, and 10 support up to 462-MHz input clock frequency. PLLs 1, 2, 3, and 4 support up to 644.5-MHz input clock frequency on the `CLK0`, `CLK2`, `CLK9`, and `CLK11` pins only.

(2) This parameter is for high-speed differential I/O mode only.

(3) These counters have a maximum of 32 if programmed for 50/50 duty cycle. Otherwise, they have a maximum of 16. High-speed differential I/O mode supports 1, 2, 4, 8, or 10.

Software

Stratix devices are supported by the Altera Quartus II design software, which provides a comprehensive environment for system-on-a-programmable-chip (SOPC) design. The Quartus II software includes HDL and schematic design entry, compilation and logic synthesis, full simulation and advanced timing analysis, SignalTap® logic analysis, and device configuration. See the *Design Software Selector Guide* for more details on the Quartus II software features.

The Quartus II software supports the Windows 2000/NT/98, Sun Solaris, Linux Red Hat v6.2 and HP-UX operating systems. It also supports seamless integration with industry-leading EDA tools through the NativeLink® interface.

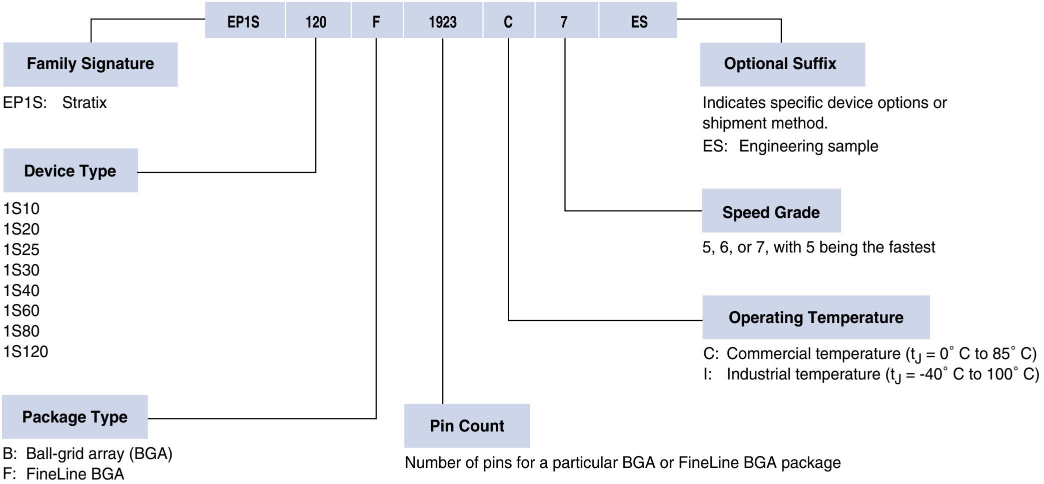
Device Pin-Outs

Device pin-outs for Stratix devices will be released on the Altera web site (**http://www.altera.com**).

Ordering Information

Figure 81 describes the ordering codes for Stratix devices. For more information on a specific package, refer to the *Altera Device Package Information Data Sheet*.

Figure 81. Stratix Device Packaging Ordering Information



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
lit_req@altera.com

I.S. EN ISO 9001