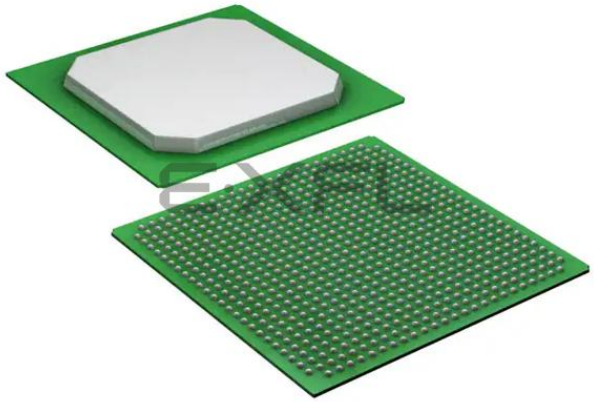


Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

Applications of Embedded - FPGAs

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

Details	
Product Status	Obsolete
Number of LABs/CLBs	1846
Number of Logic Elements/Cells	18460
Total RAM Bits	1669248
Number of I/O	426
Number of Gates	-
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	672-BBGA
Supplier Device Package	672-BGA (35x35)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep1s20b672c6n

Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Designs.*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
	The hand points to information that requires special attention.
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

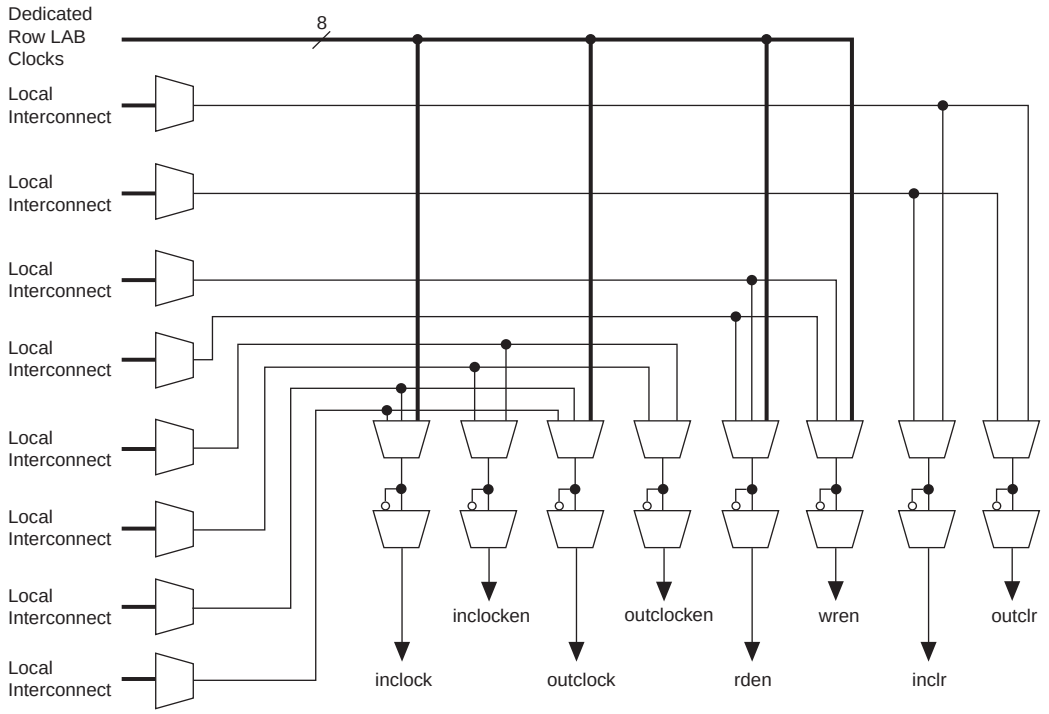
Shift Register Support

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

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

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

Figure 2–15. M512 RAM Block Control Signals



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

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

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

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

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

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

M-RAM Block

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

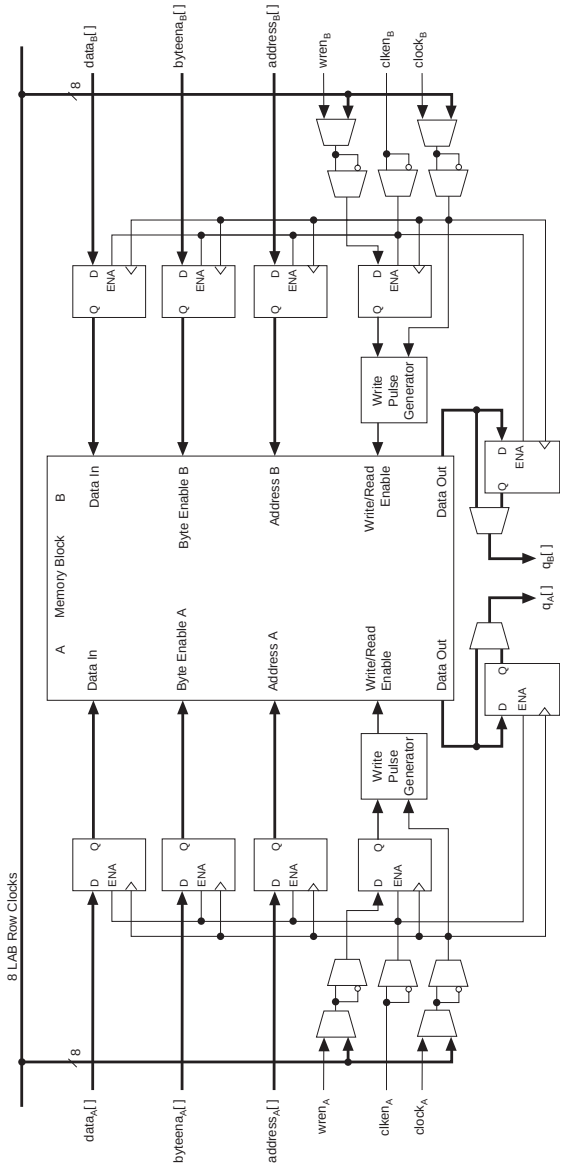
- True dual-port RAM
- Simple dual-port RAM
- Single-port RAM
- FIFO RAM

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

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

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

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



Notes to Figure 2–24

(1) All registers shown have asynchronous clear ports.

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

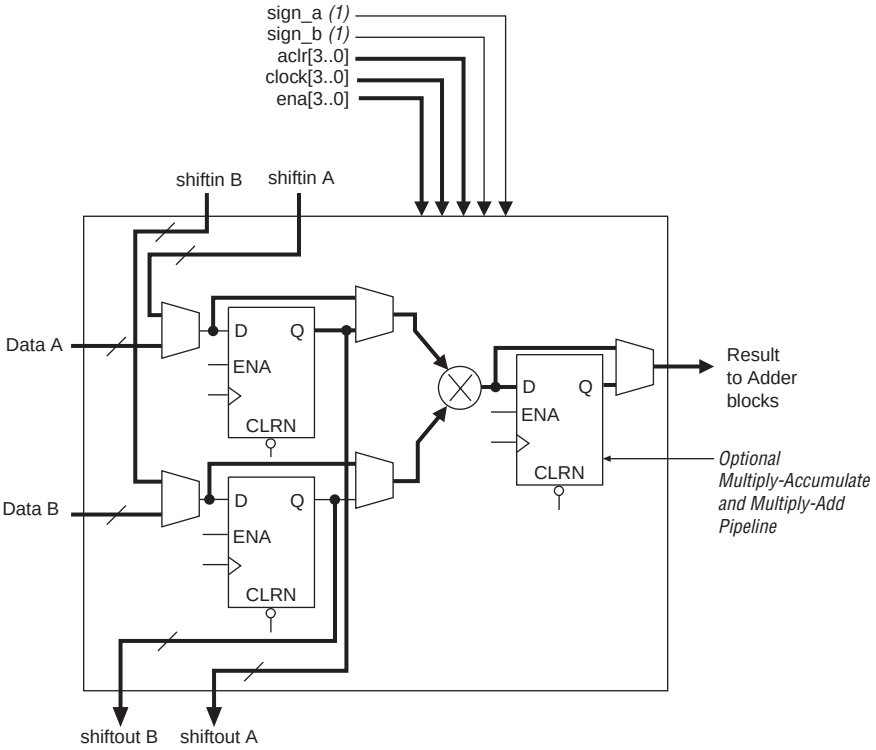
The DSP block consists of the following elements:

- Multiplier block
- Adder/output block

Multiplier Block

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

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



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

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

Table 2–17. DSP Block Signal Sources & Destinations

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

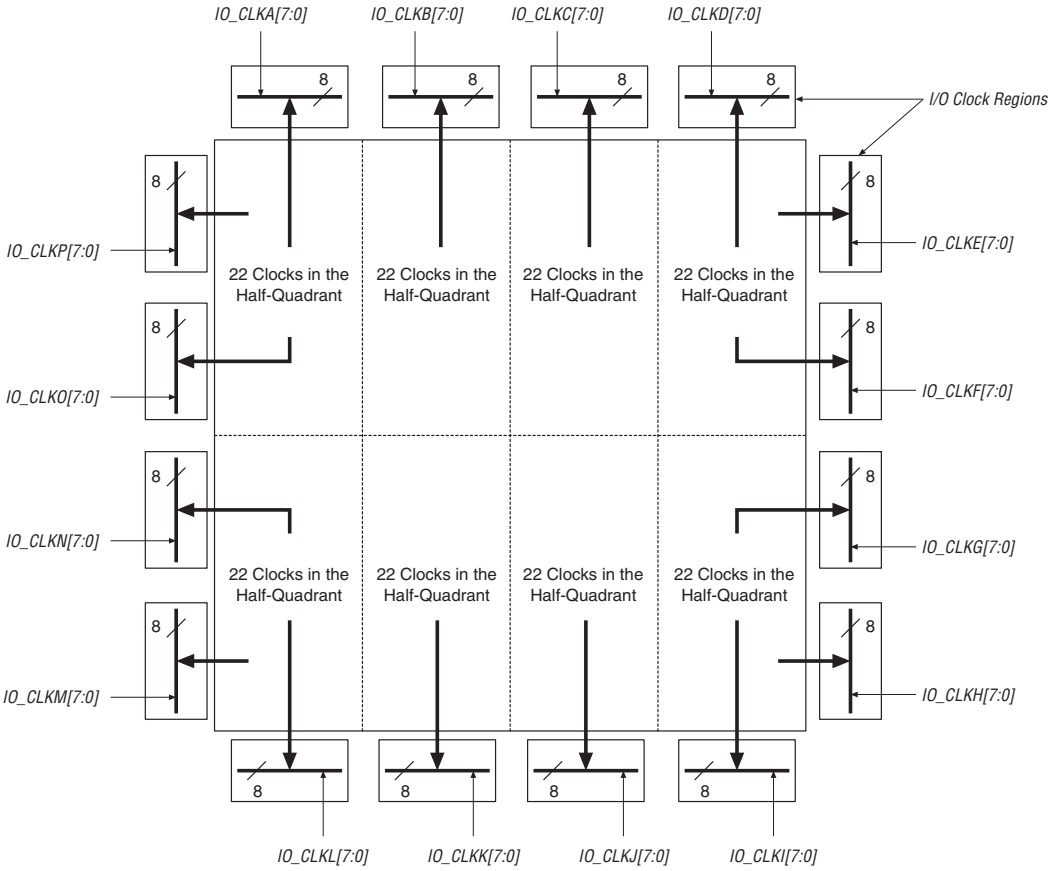
PLLs & Clock Networks

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

Global & Hierarchical Clocking

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

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



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

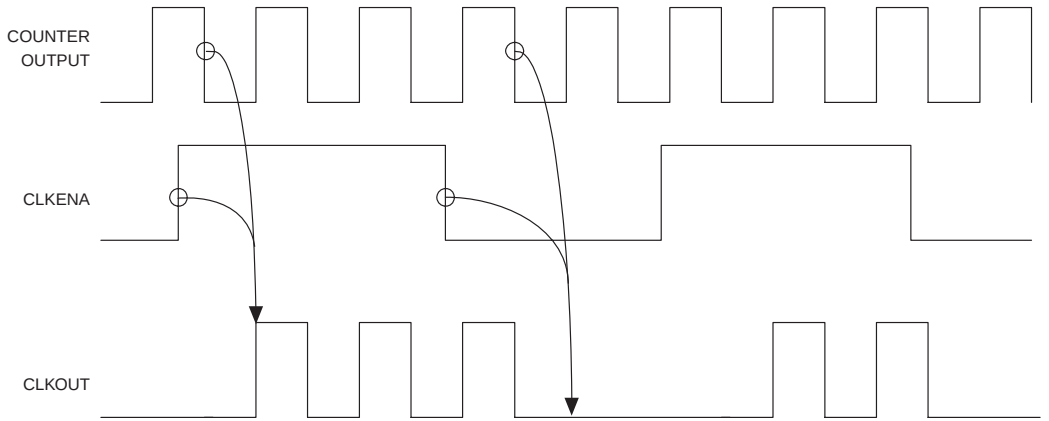
Enhanced & Fast PLLs

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

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

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

Figure 2–57. extclkena Signals

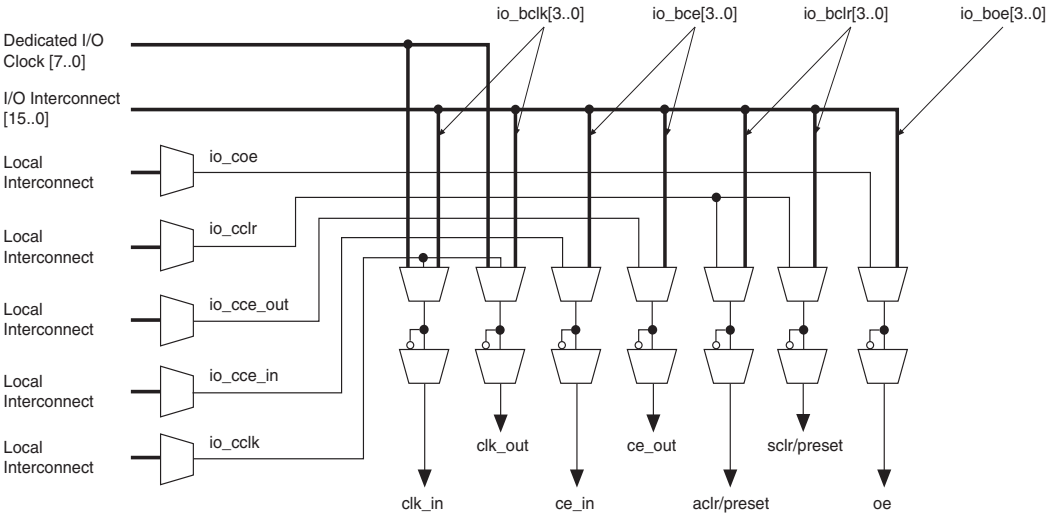


Fast PLLs

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

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

Figure 2–63. Control Signal Selection per IOE



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

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

Bus Hold

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

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

Table 2–29. Bus Hold Support

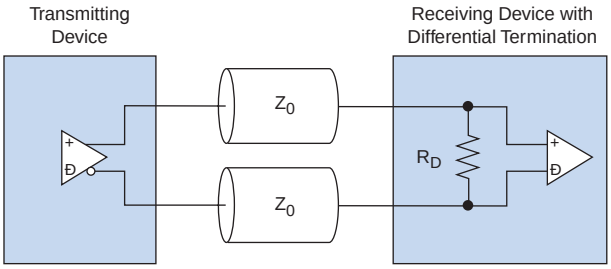
Pin Type	Bus Hold
I/O pins	✓
CLK[15..0]	
CLK[0,1,2,3,8,9,10,11]	
FCLK	✓
FPLL[7..10]CLK	

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

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

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

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



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

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

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

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

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

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

Table 2–34. Differential Termination Support Across Pin Types

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

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

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

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

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

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

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

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

Configuring Stratix FPGAs with JRunner

JRunner is a software driver that configures Altera FPGAs, including Stratix FPGAs, through the ByteBlaster II or ByteBlasterMV cables in JTAG mode. The programming input file supported is in Raw Binary File (**.rbf**) format. JRunner also requires a Chain Description File (**.cdf**) generated by the Quartus II software. JRunner is targeted for embedded JTAG configuration. The source code is developed for the Windows NT operating system (OS), but can be customized to run on other platforms. For more information on the JRunner software driver, see the JRunner Software Driver: An Embedded Solution to the JTAG Configuration White Paper and the source files on the Altera web site (**www.altera.com**).

Configuration Schemes

You can load the configuration data for a Stratix device with one of five configuration schemes (see Table 3–5), chosen on the basis of the target application. You 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 3–5. Data Sources for Configuration

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

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. You 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

Table 4–20. SSTL-2 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.18		3.0	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.35			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –8.1 mA *(3)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(3)*			V_{TT} – 0.57	V

Table 4–21. SSTL-2 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.18		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.35			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –16.4 mA *(3)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(3)*			V_{TT} – 0.76	V

Table 4–22. SSTL-3 Class I Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.2	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.4			V

Table 4–97. Output Pin Timing Skew Definitions (Part 2 of 2)	
Symbol	**Definition**
t_{LR_HIO}	Across all HIO banks (1, 2, 5, 6); across four similar type I/O banks
t_{TB_VIO}	Across all VIO banks (3, 4, 7, 8); across four similar type I/O banks
$t_{OVERALL}$	Output timing skew for all I/O pins on the device.

Notes to Table 4–97:
(1) See Figure 4–5 on page 4–57.
(2) See Figure 4–6 on page 4–58.

Table 4–98 shows the I/O skews when using the same global or regional clock to feed IOE registers in I/O banks around each device. These values can be used for calculating the timing budget on the output (write) side of a memory interface. These values already factor in the package skew.

Table 4–98. Output Skew for Stratix by Device Density			
Symbol	**Skew (ps)** *(1)*		
	EP1S10 to EP1S30	**EP1S40**	**EP1S60 & EP1S80**
t_{SB_HIO}	90	290	500
t_{SB_VIO}	160	290	500
t_{SS_HIO}	90	460	600
t_{SS_VIO}	180	520	630
t_{LR_HIO}	150	490	600
t_{TB_VIO}	190	580	670
$t_{OVERALL}$	430	630	880

Note to Table 4–98:
(1) The skew numbers in Table 4–98 account for worst case package skews.

The scaling factors for column output pin timing in Tables 4–111 to 4–113 are shown in units of time per pF unit of capacitance (ps/pF). Add this delay to the t_{CO} or combinatorial timing path for output or bidirectional pins in addition to the I/O adder delays shown in Tables 4–103 through 4–108 and the IOE programmable delays in Tables 4–109 and 4–110.

Table 4–111. Output Delay Adder for Loading on LVTTL/LVCMOS Output Buffers *Note (1)*

Conditions		Output Pin Adder Delay (ps/pF)				
Parameter	**Value**	**3.3-V LVTTL**	**2.5-V LVTTL**	**1.8-V LVTTL**	**1.5-V LVTTL**	**LVCMOS**
Drive Strength	24mA	15	–	–	-	8
	16mA	25	18	–	–	–
	12mA	30	25	25	–	15
	8mA	50	35	40	35	20
	4mA	60	–	–	80	30
	2mA	–	75	120	160	60

Note to Table 4–111:
(1) The timing information in this table is preliminary.

Table 4–112. Output Delay Adder for Loading on SSTL/HSTL Output Buffers *Note (1)*

Conditions	Output Pin Adder Delay (ps/pF)			
	SSTL-3	**SSTL-2**	**SSTL-1.8**	**1.5-V HSTL**
Class I	25	25	25	25
Class II	25	20	25	20

Note to Table 4–112:
(1) The timing information in this table is preliminary.

Table 4–113. Output Delay Adder for Loading on GTL+/GTL/CTT/PCI Output Buffers *Note (1)*

Conditions		Output Pin Adder Delay (ps/pF)				
Parameter	**Value**	**GTL+**	**GTL**	**CTT**	**PCI**	**AGP**
VCCIO Voltage Level	3.3V	18	18	25	20	20
	2.5V	15	18	-	-	-

Note to Table 4–113:
(1) The timing information in this table is preliminary.