Intel - EP1S80F1508C7N Datasheet

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	7904
Number of Logic Elements/Cells	79040
Total RAM Bits	7427520
Number of I/O	1203
Number of Gates	-
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	1508-BBGA, FCBGA
Supplier Device Package	1508-FBGA, FC (40x40)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep1s80f1508c7n

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: **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.

Chapter	Date/Version	Changes Made
4	January 2005, 3.2	- Updated rise and fall input values.
	September 2004, v3.1	- Updated Note 3 in Table 4–8 on page 4–4. - Updated Table 4–10 on page 4–6. - Updated Table 4–20 on page 4–12 through Table 4–23 on page 4–13. Added rows $V_{IL(AC)}$ and $V_{IH(AC)}$ to each table. - Updated Table 4–26 on page 4–14 through Table 4–29 on page 4–15. - Updated Table 4–31 on page 4–16. - Updated description of “External Timing Parameters” on page 4–33. - Updated Table 4–36 on page 4–20. - Added signals t_{OUTCO}, T_{XZ}, and T_{ZX} to Figure 4–4 on page 4–33. - Added rows $t_{M512CLKENSU}$ and $t_{M512CLKENH}$ to Table 4–40 on page 4–24. - Added rows $t_{M4CLKENSU}$ and $t_{M4CLKENH}$ to Table 4–41 on page 4–24. - Updated Note 2 in Table 4–54 on page 4–35. - Added rows $t_{MRAMCLKENSU}$ and $t_{MRAMCLKENH}$ to Table 4–42 on page 4–25. - Updated Table 4–46 on page 4–29. - Updated Table 4–47 on page 4–29.

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

C16 column interconnects span a length of 16 LABs and provide the fastest resource for long column connections between LABs, TriMatrix memory blocks, DSP blocks, and IOEs. C16 interconnects can cross M-RAM blocks and also drive to row and column interconnects at every fourth LAB. C16 interconnects drive LAB local interconnects via C4 and R4 interconnects and do not drive LAB local interconnects directly.

All embedded blocks communicate with the logic array similar to LAB-to-LAB interfaces. Each block (i.e., TriMatrix memory and DSP blocks) connects to row and column interconnects and has local interconnect regions driven by row and column interconnects. These blocks also have direct link interconnects for fast connections to and from a neighboring LAB. All blocks are fed by the row LAB clocks, `labclk[7..0]`.

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

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

Notes to Table 2–3:

(1) See Table 4–36 for maximum performance information.

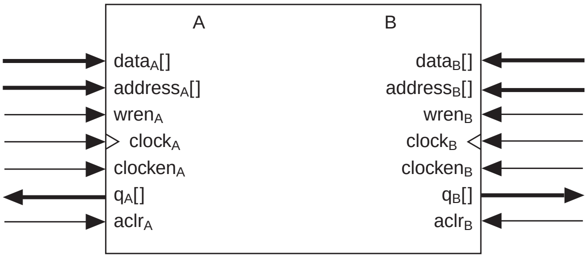
(2) The M-RAM block does not support memory initializations. However, the M-RAM block can emulate a ROM function using a dual-port RAM bock. The Stratix device must write to the dual-port memory once and then disable the write-enable ports afterwards.

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

Memory Modes

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

Figure 2–12. True Dual-Port Memory Configuration



M512 RAM Block

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

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

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

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

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

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

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

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

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

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

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

Notes to Table 2–7:

(1) Any combination of byte enables is possible.

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

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

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

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

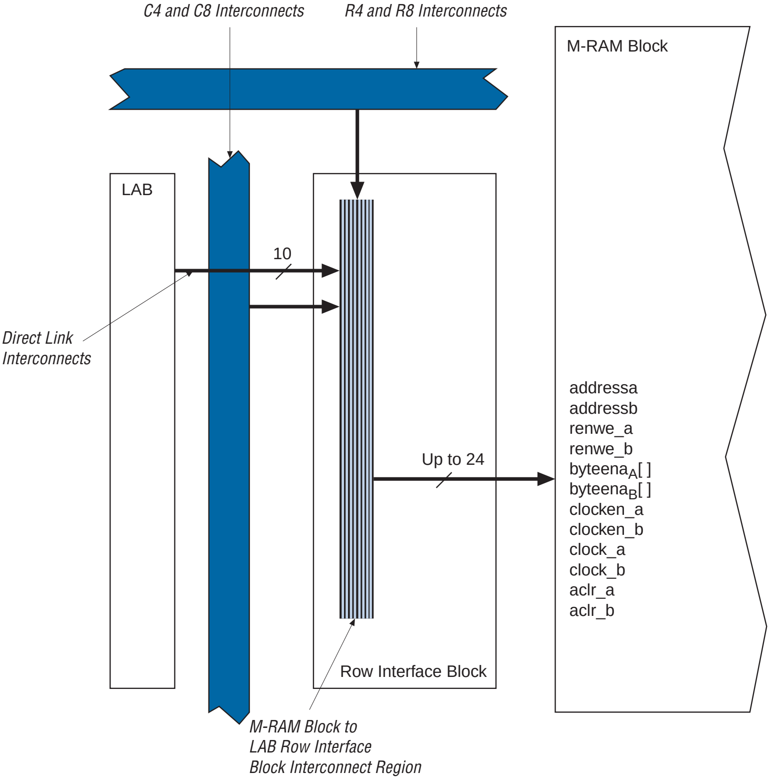


Table 2–14 shows the summary of input register modes for the DSP block.

Table 2–14. Input Register Modes

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

Multiplier

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

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

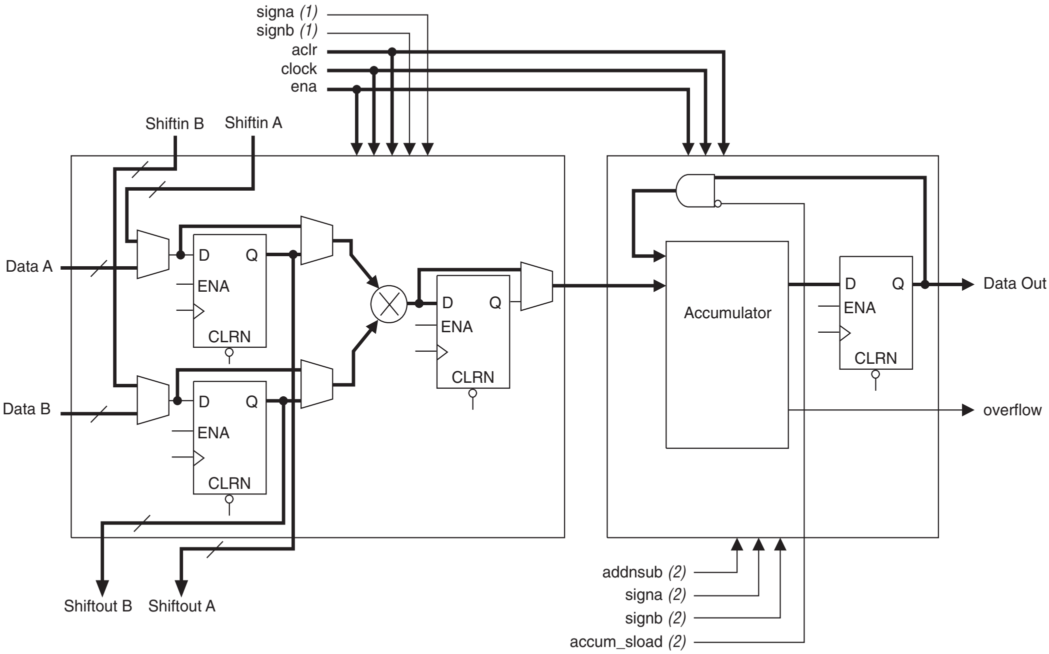
Table 2–15. Multiplier Signed Representation

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

Multiply-Accumulator Mode

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

Figure 2–37. Multiply-Accumulate Mode



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

Two-Multipliers Adder Mode

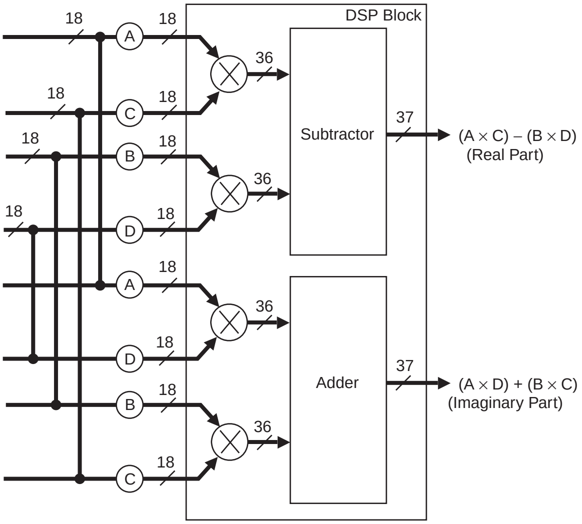
The two-multipliers adder mode uses the adder/subtractor/accumulator block to add or subtract the outputs of the multiplier block, which is useful for applications such as FFT functions and complex FIR filters. A single DSP block can implement two sums or differences from two 18 × 18-bit multipliers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

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

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

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

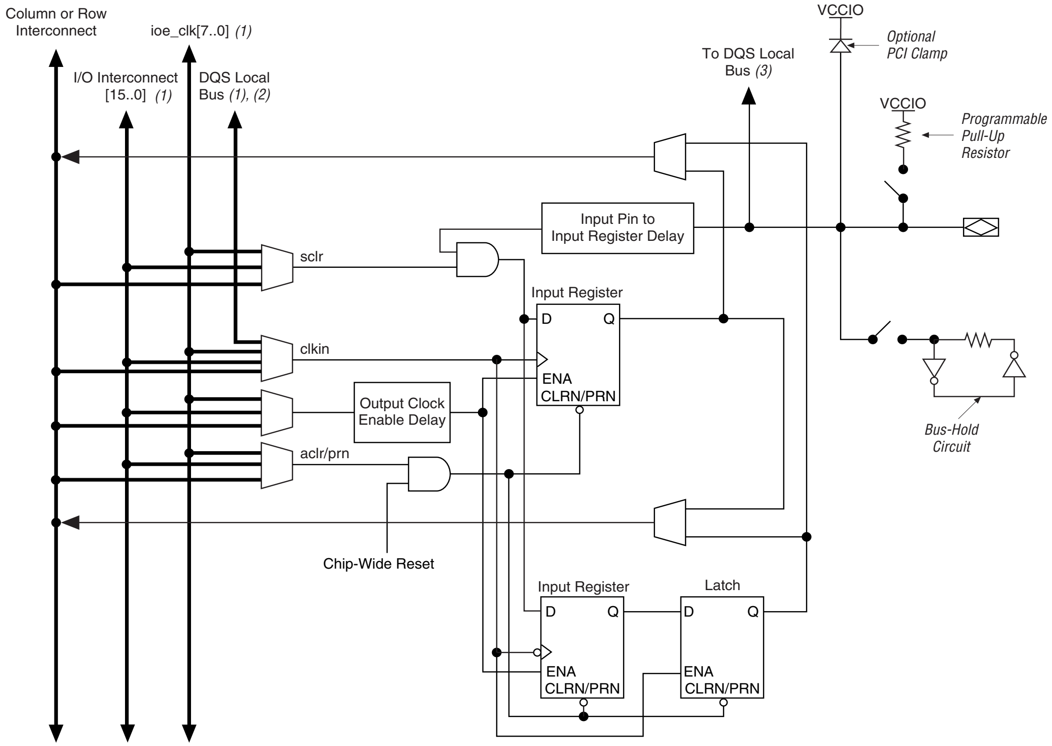
Figure 2–38. Two-Multipliers Adder Mode Implementing Complex Multiply



Four-Multipliers Adder Mode

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

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



Notes to Figure 2–65:

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

(2) This signal connection is only allowed on dedicated DQ function pins.

(3) This signal is for dedicated DQS function pins only.

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

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

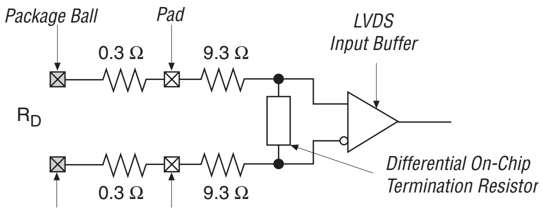


Table 2–35 defines the specification for internal termination resistance for commercial devices.

Table 2–35. Differential On-Chip Termination

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

Notes to Table 2–35:

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

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

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

MultiVolt I/O Interface

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

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

Table 3–1. Stratix JTAG Instructions

JTAG Instruction	Instruction Code	Description
SAMPLE/PRELOAD	00 0000 0101	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 II embedded logic analyzer.
EXTEST *(1)*	00 0000 0000	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	11 1111 1111	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	00 0000 0111	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	00 0000 0110	Selects the `IDCODE` register and places it between `TDI` and `TDO`, allowing the `IDCODE` to be serially shifted out of `TDO`.
`HIGHZ` *(1)*	00 0000 1011	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)*	00 0000 1010	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™, ByteBlasterMV™, or ByteBlaster™ II download cable, or when using a Jam File or Jam Byte-Code File via an embedded processor or JRunner.
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the `nCONFIG` pin low to trigger reconfiguration even though the physical pin is unaffected.
CONFIG_IO	00 0000 1101	Allows configuration of I/O standards through the JTAG chain for JTAG testing. Can be executed before, after, or during configuration. Stops configuration if executed during configuration. Once issued, the `CONFIG_IO` instruction will hold `nSTATUS` low to reset the configuration device. `nSTATUS` is held low until the device is reconfigured.
SignalTap II instructions		Monitors internal device operation with the SignalTap II embedded logic analyzer.

Note to Table 3–1:

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

Internal Timing Parameters

Internal timing parameters are specified on a speed grade basis independent of device density. Tables 4–37 through 4–42 describe the Stratix device internal timing microparameters for LEs, IOEs, TriMatrix™ memory structures, DSP blocks, and MultiTrack interconnects.

Table 4–37. 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}	Register minimum clock high or low time. The maximum core clock frequency can be calculated by $1/(2 \times t_{CLKHL})$.

Table 4–38. IOE Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{SU_R}	Row IOE input register setup time
t_{SU_C}	Column IOE input register setup time
t_H	IOE input and output register hold time after clock
t_{CO_R}	Row IOE input and output register clock-to-output delay
t_{CO_C}	Column IOE input and output register clock-to-output delay
$t_{PIN2COMBOUT\_R}$	Row input pin to IOE combinatorial output
$t_{PIN2COMBOUT\_C}$	Column input pin to IOE combinatorial output
$t_{COMBIN2PIN\_R}$	Row IOE data input to combinatorial output pin
$t_{COMBIN2PIN\_C}$	Column IOE data input to combinatorial output pin
t_{CLR}	Minimum clear pulse width
t_{PRE}	Minimum preset pulse width
t_{CLKHL}	Register minimum clock high or low time. The maximum I/O clock frequency can be calculated by $1/(2 \times t_{CLKHL})$. Performance may also be affected by I/O timing, use of PLL, and I/O programmable settings.

Table 4–42. M-RAM Block Internal Timing Microparameter Descriptions (Part 2 of 2)

Symbol	Parameter
$t_{MRAMDATABH}$	B port hold time after clock
$t_{MRAMADDRBSU}$	B port address setup time before clock
$t_{MRAMADDRBH}$	B port address hold time after clock
$t_{MRAMDATACO1}$	Clock-to-output delay when using output registers
$t_{MRAMDATACO2}$	Clock-to-output delay without output registers
$t_{MRAMCLKHL}$	Register minimum clock high or low time. This is a limit on the min time for the clock on the registers in these blocks. The actual performance is dependent upon the internal point-to-point delays in the blocks and may give slower performance as shown in Table 4–36 on page 4–20 and as reported by the timing analyzer in the Quartus II software.
$t_{MRAMCLR}$	Minimum clear pulse width.

Table 4–75. EP1S30 External I/O Timing on Column Pins Using Global Clock Networks (Part 2 of 2)

Parameter	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	Min	Max	
t_{XZ}	2.754	5.406	2.754	5.848	2.754	6.412	2.754	7.159	ns
t_{ZX}	2.754	5.406	2.754	5.848	2.754	6.412	2.754	7.159	ns
$t_{INSUPLL}$	1.265		1.236		1.403		1.756		ns
t_{INHPLL}	0.000		0.000		0.000		0.000		ns
$t_{OUTCOPLL}$	1.068	2.302	1.068	2.483	1.068	2.510	1.068	2.423	ns
t_{XZPLL}	1.008	2.176	1.008	2.351	1.008	2.386	1.008	2.308	ns
t_{ZXPLL}	1.008	2.176	1.008	2.351	1.008	2.386	1.008	2.308	ns

Table 4–76. EP1S30 External I/O Timing on Row Pins Using Fast Regional Clock Networks

Parameters	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.616		2.808		3.223		3.797		ns
t_{INH}	0.000		0.000		0.000		0.000		ns
t_{OUTCO}	2.542	5.114	2.542	5.502	2.542	5.965	2.542	6.581	ns
t_{XZ}	2.569	5.168	2.569	5.558	2.569	6.033	2.569	6.663	ns
t_{ZX}	2.569	5.168	2.569	5.558	2.569	6.033	2.569	6.663	ns

Table 4–121. Stratix Maximum Output Clock Rate (Using I/O Pins) for PLL[1, 2, 3, 4] Pins in Flip-Chip Packages

I/O Standard	-5 Speed Grade	-6 Speed Grade	-7 Speed Grade	-8 Speed Grade	Unit
LVTTL	400	350	300	300	MHz
2.5 V	400	350	300	300	MHz
1.8 V	400	350	300	300	MHz
1.5 V	350	300	300	300	MHz
LVCMOS	400	350	300	300	MHz
GTL	200	167	125	125	MHz
GTL+	200	167	125	125	MHz
SSTL-3 Class I	167	150	133	133	MHz
SSTL-3 Class II	167	150	133	133	MHz
SSTL-2 Class I	150	133	133	133	MHz
SSTL-2 Class II	150	133	133	133	MHz
SSTL-18 Class I	150	133	133	133	MHz
SSTL-18 Class II	150	133	133	133	MHz
1.5-V HSTL Class I	250	225	200	200	MHz
1.5-V HSTL Class II	225	225	200	200	MHz
1.8-V HSTL Class I	250	225	200	200	MHz
1.8-V HSTL Class II	225	225	200	200	MHz
3.3-V PCI	250	225	200	200	MHz
3.3-V PCI-X 1.0	225	225	200	200	MHz
Compact PCI	400	350	300	300	MHz
AGP 1×	400	350	300	300	MHz
AGP 2×	400	350	300	300	MHz
CTT	300	250	200	200	MHz
LVPECL *(2)*	717	717	500	500	MHz
PCML *(2)*	420	420	420	420	MHz
LVDS *(2)*	717	717	500	500	MHz
HyperTransport technology *(2)*	420	420	420	420	MHz

Table 4–126. High-Speed I/O Specifications for Wire-Bond Packages (Part 1 of 2)

Symbol	Conditions	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSCLK} (Clock frequency) (LVDS,LVPECL, HyperTransport technology) $f_{HSCLK} = f_{HSDR}$ / W	*W* = 4 to 30 (Serdes used)	10		156	10		115.5	10		115.5	MHz
	W = 2 (Serdes bypass)	50		231	50		231	50		231	MHz
	W = 2 (Serdes used)	150		312	150		231	150		231	MHz
	W = 1 (Serdes bypass)	100		311	100		270	100		270	MHz
	W = 1 (Serdes used)	300		624	300		462	300		462	MHz
f_{HSDR} Device operation, (LVDS,LVPECL, HyperTransport technology)	J = 10	300		624	300		462	300		462	Mbps
	J = 8	300		624	300		462	300		462	Mbps
	J = 7	300		624	300		462	300		462	Mbps
	J = 4	300		624	300		462	300		462	Mbps
	J = 2	100		462	100		462	100		462	Mbps
	J = 1 (LVDS and LVPECL only)	100		311	100		270	100		270	Mbps
f_{HSCLK} (Clock frequency) (PCML) $f_{HSCLK} = f_{HSDR}$ / W	*W* = 4 to 30 (Serdes used)	10		77.75							MHz
	W = 2 (Serdes bypass)	50		150	50		77.5	50		77.5	MHz
	W = 2 (Serdes used)	150		155.5							MHz
	W = 1 (Serdes bypass)	100		200	100		155	100		155	MHz
	W = 1 (Serdes used)	300		311							MHz
Device operation, f_{HSDR} (PCML)	J = 10	300		311							Mbps
	J = 8	300		311							Mbps
	J = 7	300		311							Mbps
	J = 4	300		311							Mbps
	J = 2	100		300	100		155	100		155	Mbps
	J = 1	100		200	100		155	100		155	Mbps
TCCS	All			400			400			400	ps