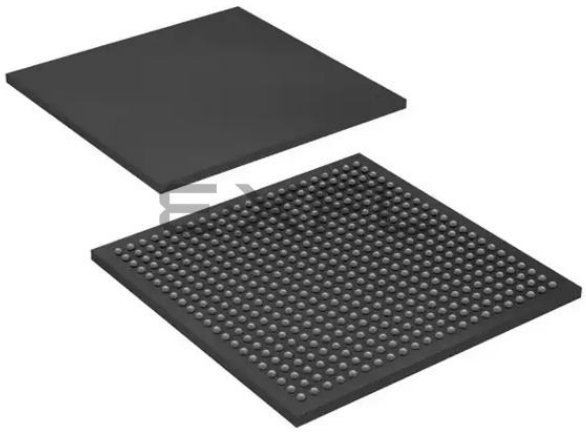


Welcome to E-XFL.COM

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.

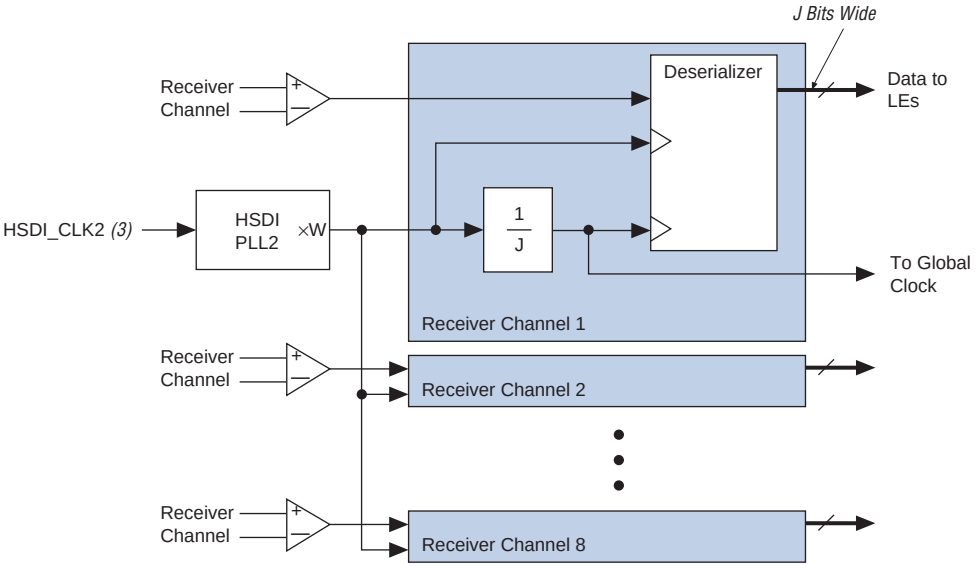
Applications of Embedded - FPGAs

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

Details	
Product Status	Obsolete
Number of LABs/CLBs	480
Number of Logic Elements/Cells	4800
Total RAM Bits	49152
Number of I/O	303
Number of Gates	120000
Voltage - Supply	1.71V ~ 1.89V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BBGA, FCBGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep1m120f484c7n

In source-synchronous mode, source synchronous interfacing is supported at up to 840 Mbps. Serial channels are transmitted and received along with a low speed clock. The receiving device then multiplies the clock by a factor of 1 to 12, 14, 16, 18, or 20. The serialization / deserialization rate can be any number from 4, 7, 8, 9 to 12, 14, 16, 18, or 20 and does not have to equal the clock multiplication value. For example, an 840-Mbps LVDS channel can be received along with a 84-MHz clock. The 84-MHz clock is multiplied by 10 to drive the serial shift register, but the register can be clocked out in parallel at 7-, 8-, 9- to 12-, 14-, 16-, 18-, or 20-bits wide at 42 to 120 MHz. See Figures 2 and 3.

Figure 2. Receiver Diagram for Source Synchronous Mode *Notes (1), (2)*



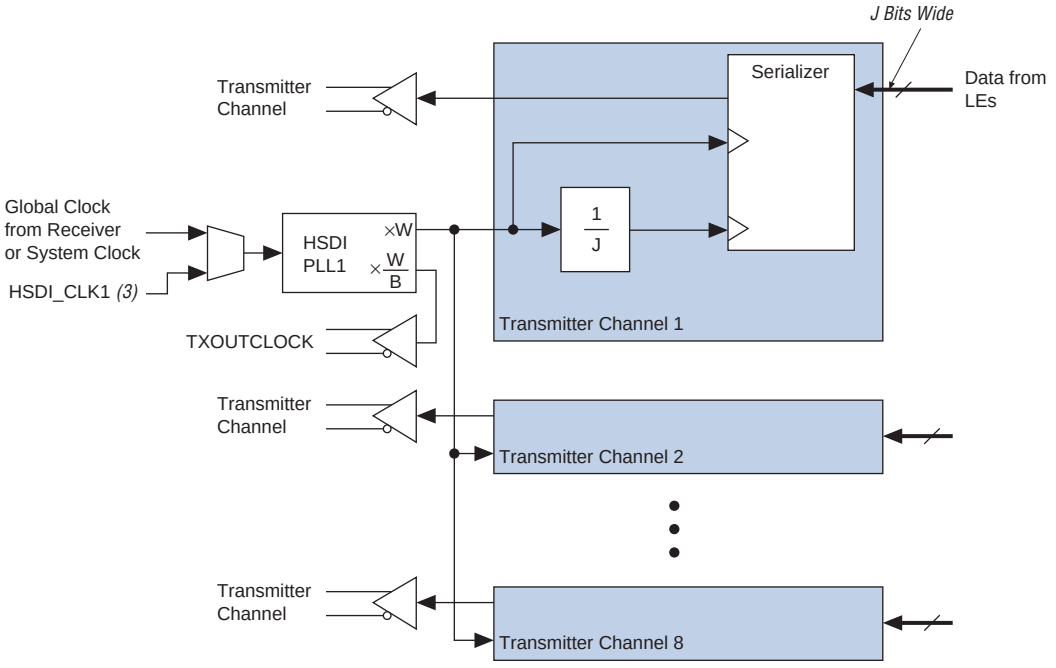
Notes to Figure 2:

(1) EP1M350 devices have 18 individual receiver channels. EP1M120 devices have 8 individual receiver channels.

(2) *W* = 1 to 12, 14, 16, 18, or 20
J = 4, 7, 8, 9 to 12, 14, 16, 18, or 20
W does not have to equal *J*.

(3) This clock pin drives an HSDI PLL only. It does not drive to the core.

Figure 3. Transmitter Diagram for Source Synchronous Mode *Notes (1), (2)*



Notes to Figure 3:

(1) EP1M350 devices have 18 individual transmitter channels. EP1M120 devices have 8 individual transmitter channels.

(2) *W* = 1 to 12, 14, 16, 18, or 20
B = 1 to 12, 14, 16, 18, or 20
J = 4, 7, 8, 9 to 12, 14, 16, 18, or 20
W, *B*, and *J* do not have to be equal.

(3) This clock pin drives an HSDI PLL only. It does not drive to the logic array.

The Mercury device's source-synchronous mode also supports the RapidIO interface protocol at up to 500 Mbps using the LVDS I/O standard.

For more information on source synchronous interfacing see *AN 159: Using HSDI in Source-Synchronous Mode in Mercury Devices*.

Mercury device HSDI performance is finalized for certain speed grades. Also, the industrial-grade CDR specification is the same as the -6 speed grade for commercial-grade CDR specification. See Table 8.

Table 8. CDR & Source-Synchronous Data Rates

Device	Speed Grade	Number of Channels	Maximum CDR Data Rate (Gbps)	Maximum Source-Synchronous Data Rate (Mbps)
EP1M120	-5	8	1.25	840
	-6 *(1)*	8	1.25	840
	-7	8	1.0	840
EP1M350	-5	18	1.25	840
	-6 *(1)*	8 *(2)*	1.25	840
		10 *(2)*	1.0	840
	-7	18	1.0	840

Notes to Table 8:

(1) The -6 speed grade specifications apply for both commercial and industrial devices.

(2) EP1M350 devices can support any 8 channels at 1.25 Gbps. The other 10 channels must run at 1.0 Gbps or less.

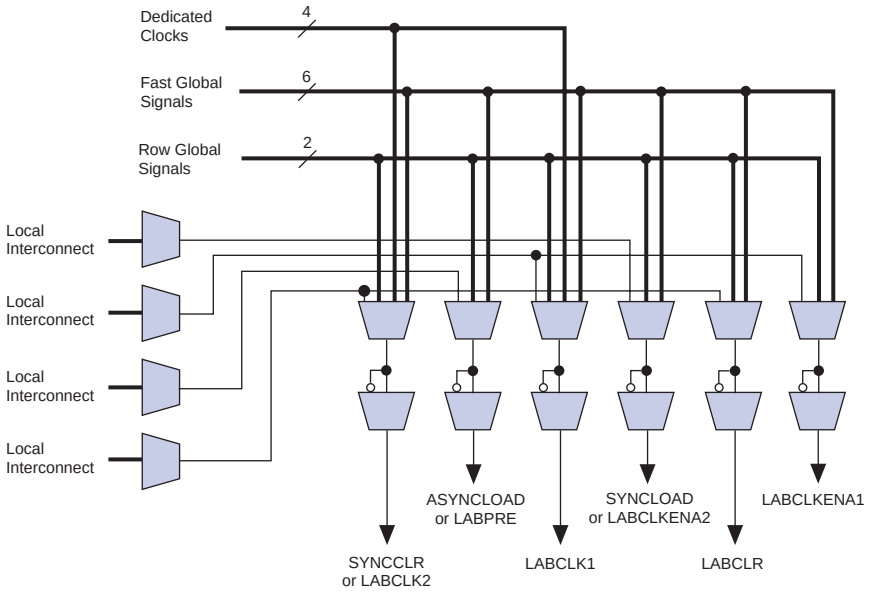
Logic & Interconnect

Mercury device logic is implemented in LEs. LE resources are used differently according to specific operating modes and the type of logic function being implemented. LEs are grouped into LABs in a row-based architecture. The multi-level FastTrack Interconnect structure provides the routing connection between LEs, ESBs, and IOEs.

Logic Array Block

Each LAB consists of 10 LEs, LE carry chains, multiplier circuitry, LAB control signals, local interconnect, and FastLUT connection lines. The local interconnect transfers signals between LEs within the same or adjacent LABs. FastLUT connections transfer the output of one LE to the adjacent LE for ultra-fast sequential LE connections within the same LAB. The Quartus II Compiler places associated logic within a LAB or adjacent LABs, allowing the use of fast local and FastLUT connections for high performance. Figure 5 shows the Mercury LAB structure.

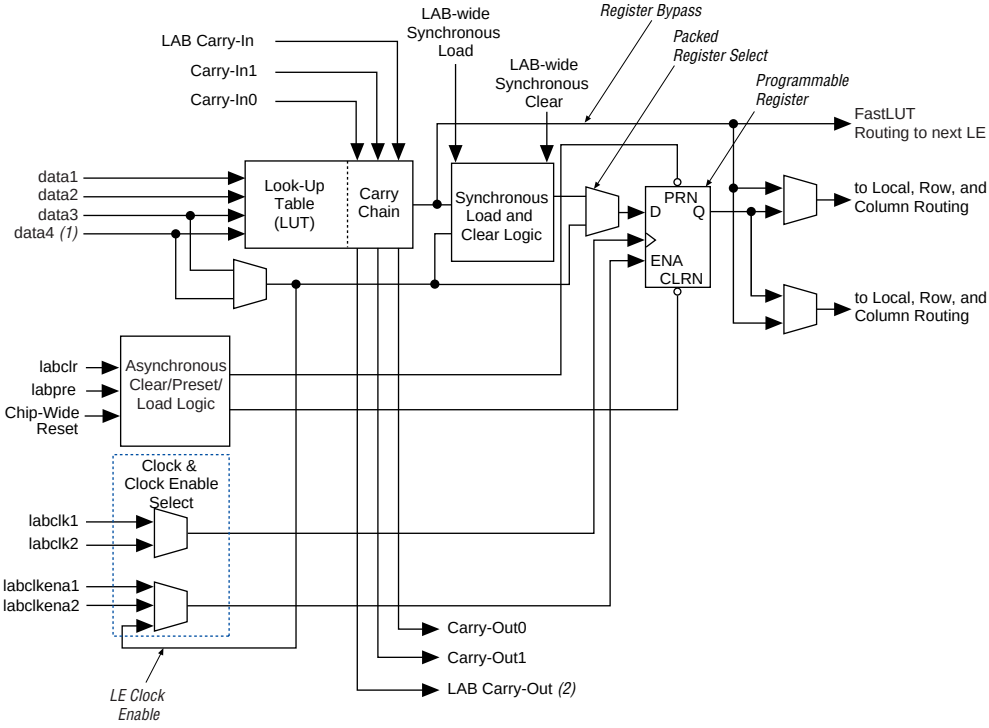
Figure 6. LAB-Wide Control Signals



Logic Element

The LE, the smallest unit of logic in the Mercury architecture, is compact and provides efficient logic usage. Each LE contains a four-input LUT, which is a function generator that can quickly implement any function of four variables. In addition, each LE contains a programmable register and carry chain with carry select look ahead capability. Each LE drives all interconnect types: local interconnect, row and priority row interconnect, column and priority column interconnect, leap lines, and RapidLAB interconnect. Each LE also has the ability to drive its combinatorial output directly to the next LE in the LAB using FastLUT connections. See Figure 7.

Figure 7. Mercury LE



Notes to Figure 7:

(1) FastLUT interconnect uses the `data4` input.

(2) LAB carry-out can only be generated by LE 4 and / or LE 10.

Each LE's programmable register can be configured for D, T, JK, or SR operation. The register's clock, clock enable, and clear control signals can be driven by global signals, general-purpose I / O pins, or any internal logic. For combinatorial functions, the register is bypassed and the output of the LUT drives directly to the outputs of the LE.

Each LE has four data inputs that can drive the internal LUT. One of these inputs has a shorter delay than the others, improving overall LE performance. This input is chosen automatically by the Quartus II software as appropriate.

The Quartus II Compiler creates carry chains longer than ten LEs by linking LABs together automatically. For enhanced fitting, a long carry chain skips intermediate LABs in a row structure. A carry chain longer than one LAB skips either from an even-numbered LAB to the next even-numbered LAB, or from an odd-numbered LAB to the next odd-numbered LAB. For example, the last LE of the first LAB in a LAB row carries to the first LE of the third LAB in the same LAB row.

Multiplier Mode

Multiplier mode is used for implementing high-speed multipliers up to 16×16 in size. The LUT implements the partial product formation and summation in a single stage for a $N \times M$-bit multiply operation. A single LE can implement the summation of $A_N B_{M+1} + A_{N+1} B_M$ for the multiplier and multiplicand inputs. To increase the speed of the multiplication, LAB wide signals are used to control the partial product sum generation. These multiplier LAB-wide signals use the LABCLKENA1 and PRESET/ASYNCLOAD resources. The multiplier mode takes advantage of the CSLA circuitry for optimized sum and carry generation in the partial product sum. There is a special CSLA circuitry mode used for the multiplier where the carry chain runs vertically between LABs in the same column. The Quartus II Compiler automatically uses this special mode for dedicated multiplier implementation only. The summation of the multiplier and multiplicand bits is driven out along with the carry-out0 and carry-out1 bits. The combinatorial or registered versions of the sum can be driven out, allowing the multiplier to be pipelined.

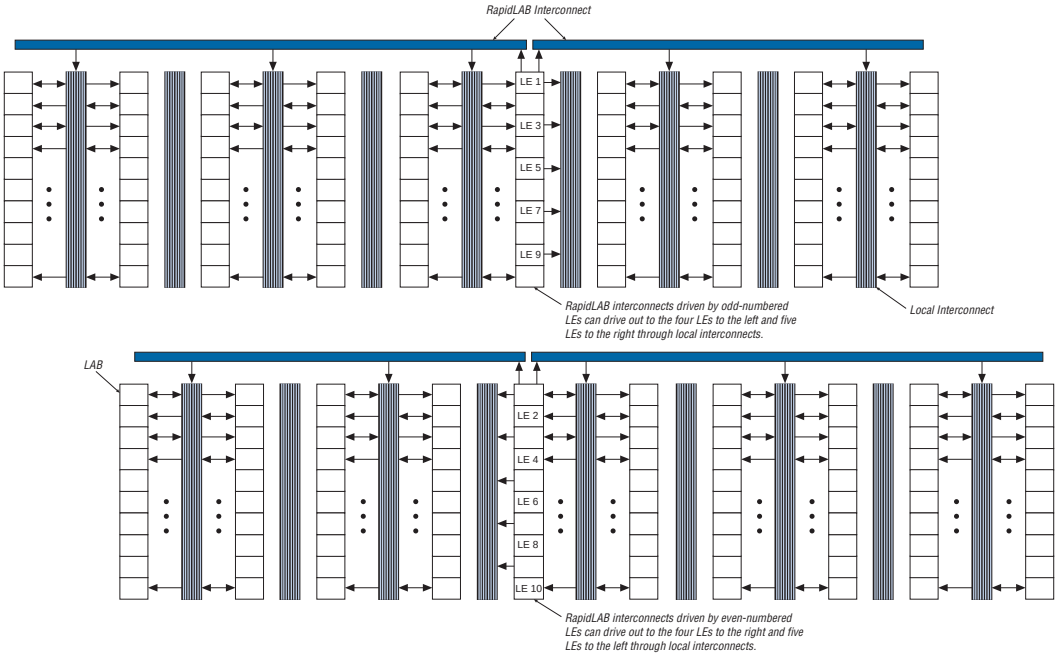
The RapidLAB interconnect has dedicated fast connections to the LE inputs in multiplier mode, further increasing the speed of the multiplier. These dedicated connections allow RapidLAB lines to avoid delay incurred by driving onto local interconnects and then into the LE.

The Quartus II software implements parameterized functions that use the multiplier mode automatically when multiply operators are used.

Figure 11 shows a Mercury device LE in multiplier mode.

The RapidLAB interconnect provides a specialized high-speed structure to allow a central LAB to drive other LABs within a 10-LAB-wide region. The RapidLAB lines drive alternating local LAB interconnect regions, allowing communication to all LABs in the 10-LAB-wide region. Even numbered LEs in a LAB directly drive a RapidLAB line that drives one set of alternating local interconnect regions, while odd-numbered LEs drive a RapidLAB line that drives the opposite set of alternating local interconnect regions. Figure 14 shows RapidLAB interconnect connections. This 10-LAB wide region of the RapidLAB interconnect is repeated for every LAB in the row. The region covered by the RapidLAB interconnect is smaller than 10 for source LABs that are four or five LABs in from either edge of the LAB row. The RapidLAB row interconnect is used for LAB-to-LAB routing; it is only used by I/O bands or ESBs indirectly through other interconnects. The RapidLAB interconnect drives an LE directly when that LE is in multiplier mode.

Figure 14. RapidLAB Interconnect Connections

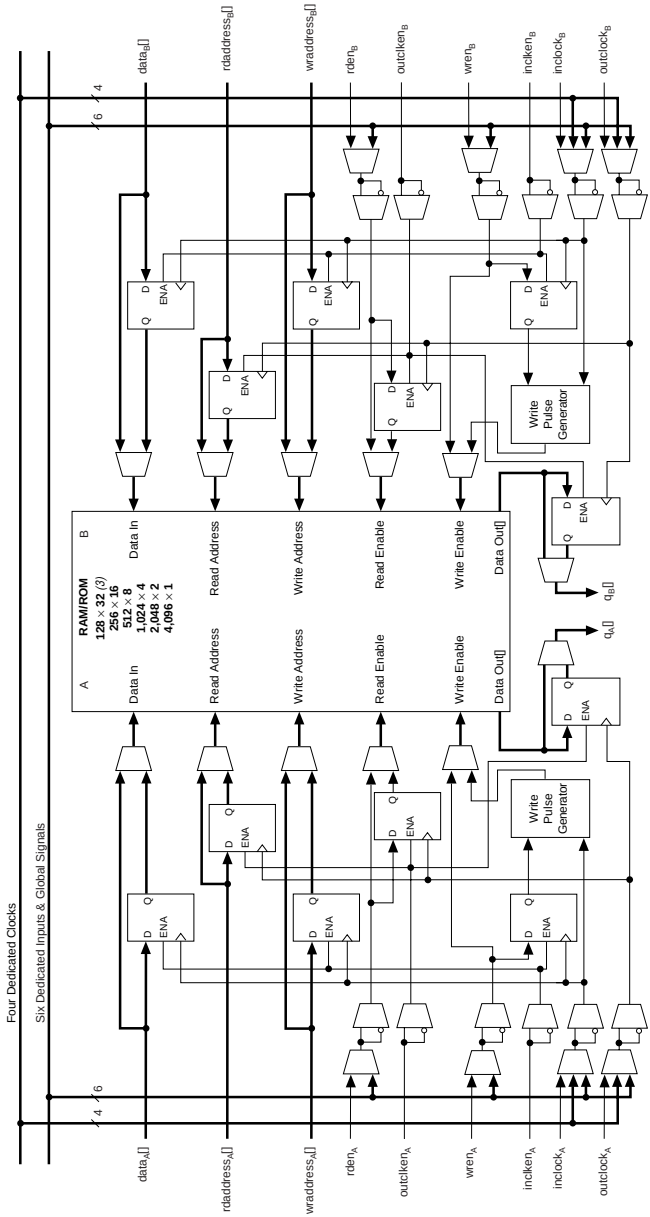


The column interconnect vertically routes signals to and from LABs, ESBs, and I/O bands. Each column of LABs is served by a dedicated column interconnect. These column resources include:

- Column interconnect traversing the entire device from top to bottom
- Priority column interconnect for high speed access across the device vertically
- Leap line interconnect for vertical routing between adjacent LAB rows and between adjacent ESP rows and LAB rows.

Leap lines are driven directly by LEs for fast access to adjacent row interconnects. LABs can drive a leap line to the row above and/or below (including ESB rows). The even-numbered LEs in a LAB drive leap lines down, while odd-numbered LEs drive leap lines up. This allows a single LAB to access row and RapidLAB interconnects within a three-row region. Figure 15 shows the leap line interconnect.

Figure 19. ESB in Read/Write Clock Mode *Notes (1), (2)*



Notes to Figure 19:

(1) Only half of the ESB, either A or B, is used for dual-port configuration.
(2) All registers can be asynchronously cleared by ESB local interconnect signals, global signals, or the chip-wide reset.
(3) This configuration is supported for dual-port configuration.

If the same data is written into multiple locations in the memory, a CAM block can be used in multiple-match or fast multiple-match modes. The ESB outputs the matched data's locations as an encoded or unencoded address. In multiple-match mode, it takes two clock cycles to write into a CAM block. For reading, there are 16 outputs from each ESB at each clock cycle. Therefore, it takes two clock cycles to represent the 32 words from a single ESB port. In this mode, encoded and unencoded outputs are available. To implement the encoded version, the Quartus II software adds a priority encoder with LEs. Fast multiple-match is identical to the multiple-match mode, however, it only takes one clock cycle to read from a CAM block and generate valid outputs. To do this, the entire ESB is used to represent 16 outputs. In fast multiple-match mode, the ESB can implement a maximum CAM block size of 16 words.

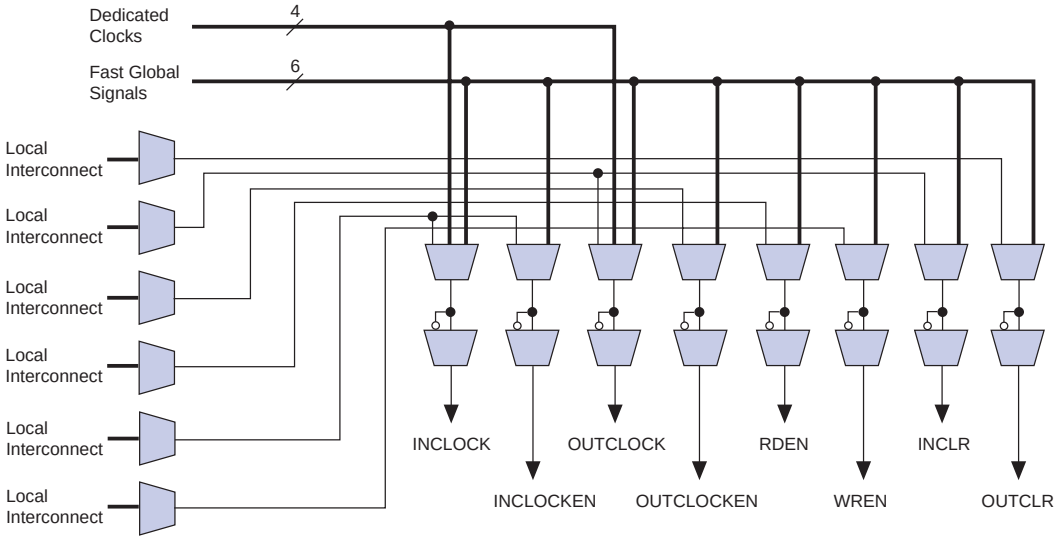
A CAM block can be pre-loaded with data during configuration, or it can be written during system operation. In most cases, two clock cycles are required to write each word into CAM. When don't-care bits are used, a third clock cycle is required.

For more information on CAM, see *Application Note 119 (Implementing High-Speed Search Applications with APEX CAM)*.

Driving into ESBs

ESBs provide flexible options for driving control signals. Different clocks can be used for the ESB inputs and outputs. Registers can be inserted independently on the data input, data output, read address, write address, WREN, and RDEN signals on each port of the ESB. The fast global signals and ESB local interconnect can drive the WREN and RDEN signals. The fast global signals, dedicated clock pins, and ESB local interconnect can drive the ESB clock signals. The ESB local interconnect is driven by the ESB row interconnects which, in turn, are driven by all types of column interconnects, including high-speed leap lines. Because the LEs drive the column interconnect to the ESB local interconnect, the LEs can control the WREN and RDEN signals and the ESB clock, clock enable, and asynchronous clear signals. Figure 25 shows the ESB control signal generation logic.

Figure 25. ESB Control Signal Generation



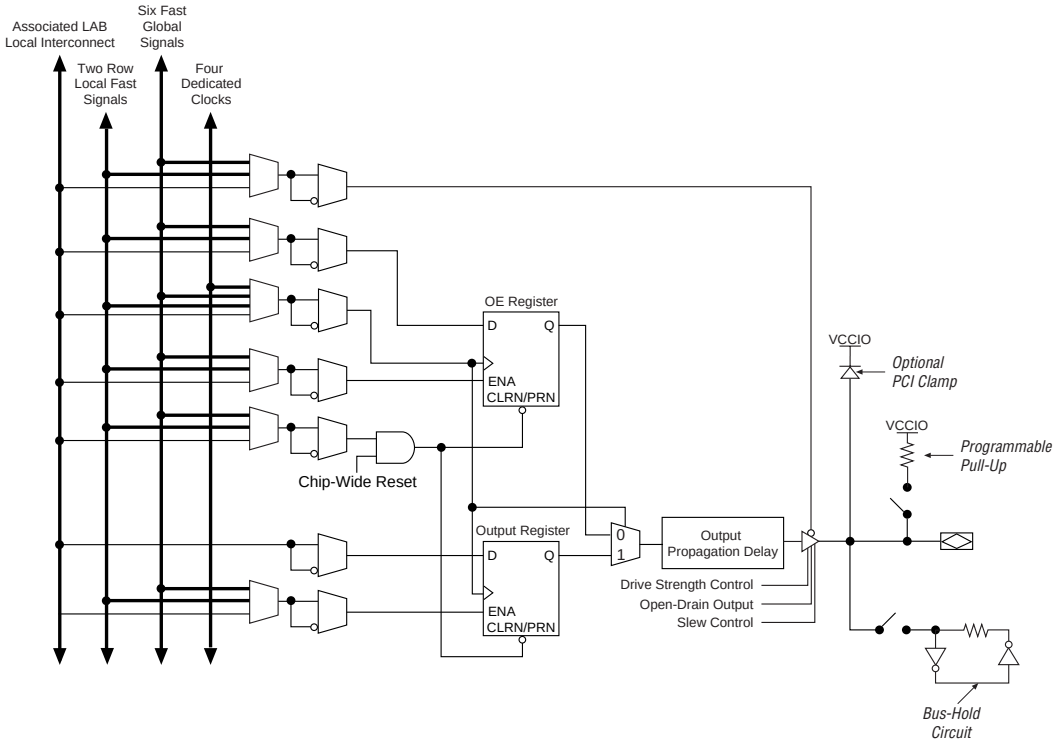
The ESB can drive row interconnects within its own ESB row and can directly drive all the column interconnects: column, priority column, and leap lines.

Implementing Logic in ROM

In addition to implementing RAM functions, the ESB can implement logic functions when it is programmed with a read-only pattern during configuration, creating a large LUT. With LUTs, combinatorial functions are implemented by looking up the results, rather than by computing them. This implementation of combinatorial functions can be faster than using algorithms implemented in general logic, a performance advantage further enhanced by the fast access times of ESBs. The large capacity of ESBs enables designers to implement complex functions in one logic level without the routing delays associated with linked LEs or distributed RAM blocks. Parameterized functions such as LPM functions can take advantage of the ESB automatically. Further, the Quartus II software can implement portions of a design with ESBs where appropriate.

When using the IOE for double data rate outputs, the output register and OE register are automatically configured to clock two data paths from LEs on rising clock edges. These register outputs are multiplexed by the clock to drive the output pin at a ×2 rate. The output register clocks the first bit out on the clock high time, while the OE register clocks the second bit out on the clock low time. Figure 28 shows the IOE configured for DDR output.

Figure 28. IOE Configured for DDR Output



Bidirectional DDR on an I/O pin is possible by using the IOE for DDR output and using LEs to acquire the double data rate input. Bidirectional DDR I/O pins support double data rate synchronous DRAM (DDR SDRAM) at 166 MHz (334 Mbps), which transfer data on a double data rate bidirectional bus. QDR SRAMs are also supported with DDR I/O pins on separate read and write ports.

Bus Hold

Each Mercury device I/O pin provides an optional bus-hold feature. When this feature is enabled for an I/O pin, the bus-hold circuitry weakly holds the signal at its last driven state. By holding the last driven state of the pin until the next input signal is present, the bus-hold feature eliminates the need to add external pull-up or pull-down resistors 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. This feature can be selected 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. The bus-hold feature should also be disabled if open-drain outputs are used with the GTL+ I/O standard.

The bus-hold circuitry weakly pulls the signal level to the last driven state through a resistor with a nominal resistance (R_{BH}) of approximately 8 kΩ. Table 42 gives specific sustaining current that will be driven through this resistor and overdrive current that will 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 Mercury device I/O pin provides an optional programmable pull-up resistor during user mode. When this feature is enabled for an I/O pin, the pull-up resistor (50 kΩ) weakly holds the output to the V_{CCIO} level of the bank that the output pin resides in.

I/O Row Bands

The I/O row bands are one of the advanced features of the Mercury architecture. All IOEs are grouped in I/O row bands across the device. The number of I/O row bands depends on the Mercury device size. The I/O row bands are designed for flip-chip technology, allowing I/O pins to be distributed across the entire chip, not only in the periphery. This array driver technology allows higher I/O pin density (I/O pins per device area) than peripheral I/O pins.

Table 16. Mercury 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	Allows the external circuitry and board-level interconnections 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.
ICR Instructions	These instructions are used when configuring a Mercury device via the JTAG port with a ByteBlasterMV™ download cable, or using a Jam STAPL or Jam Byte-Code file via an embedded processor.
SignalTap Instructions	These instructions monitor internal device operation with the SignalTap embedded logic analyzer.

The Mercury device instruction register length is 10 bits. The Mercury device USERCODE register length is 32 bits. Tables 17 and 18 show the boundary-scan register length and device IDCODE information for Mercury devices.

Table 17. Mercury Boundary-Scan Register Length

Device	Boundary-Scan Register Length (Bits)
EP1M120	1,125
EP1M350	1,695

Table 18. 32-Bit Mercury Device IDCODE

Device	IDCODE (32 Bits) *(1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	1 (1 Bit) *(2)*
EP1M120	0000	0011 0000 0000 0000	000 0110 1110	1
EP1M350	0000	0011 0000 0000 0001	000 0110 1110	1

Notes to Table 18:

(1) The most significant bit (MSB) is on the left.

(2) The IDCODE's least significant bit (LSB) is always 1.

For more information, see the following documents:

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

Generic Testing

Each Mercury device is functionally tested. Complete testing of each configurable static random access memory (SRAM) bit and all logic functionality ensures 100% yield. AC test measurements for Mercury devices are made under conditions equivalent to those shown in Figure 33. Multiple test patterns can be used to configure devices during all stages of the production flow.

Table 30. 3.3-V PCI Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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
I_I	Input pin leakage current	0 < V_{IN} < V_{CCIO}	–10		10	µA
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 31. PCI-X Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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
I_{IL}	Input leakage current	0 < V_{IN} < V_{CCIO}	–10		10	µA
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
L_{PIN}	Pin inductance				15	nH

Table 32. GTL+ I/O Specifications *Note (10)*

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} = 34 mA			0.65	V

Table 39. 1.5-V HSTL Class I Specifications *Note (10)*

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –8 mA			0.4	V

Table 40. 1.5-V HSTL Class II Specifications *Note (10)*

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA			0.4	V

Table 41. CTT I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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
I_I	Input pin leakage current	0 < V_{IN} < V_{CCIO}			±10	µA
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 ð V_{OUT} ð V_{CCIO}			±10	µA

Notes to Tables 20 – 43.:

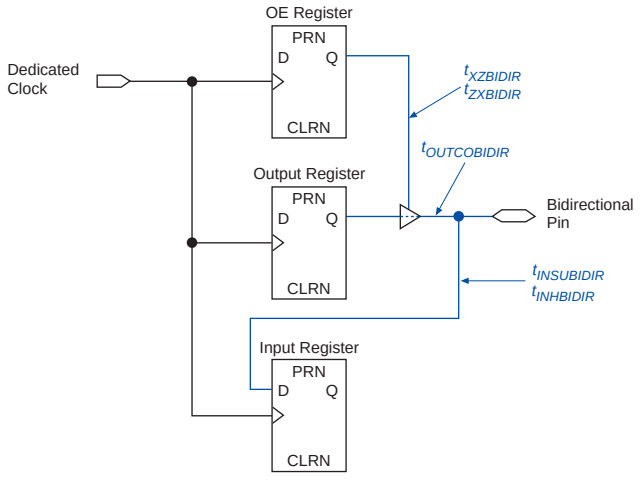
(1) See the *Operating Requirements for Altera Devices Data Sheet*.
(2) Minimum DC input is –0.5 V. During transitions, the inputs may undershoot to –0.5 V or overshoot to 4.1 V for input currents less than 100 mA and periods shorter than 20 ns.
(3) Maximum V_{CC} rise time is 100 ms, and V_{CC} must rise monotonically.
(4) V_{CCIO} maximum and minimum conditions for LVPECL, LVDS, RapidIO, and 3.3-V PCML are shown in parentheses.
(5) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT} and V_{CCIO} are powered.
(6) Typical values are for T_A = 25° C, V_{CCINT} = 1.8 V, and V_{CCIO} = 1.8 V, 2.5 V, and 3.3 V.
(7) These values are specified under the Mercury Device Recommended Operating Conditions shown in Table 3 on page 3.
(8) Input pins are grounded. In the test design, internal logic does not toggle. The test design does not use PLL or HSDI circuitry. All ESBs are in power-down mode.
(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 11 on page 53.
(11) For more information on termination, see *AN 134: Using Programmable I/O Standards in Mercury Devices* or *AN 159: Using HSDI in Source-Synchronous Mode in Mercury Devices*.
(12) V_{REF} specifies the center point of the switching range.
(13) Capacitance is sample-tested only. Capacitance is measured using time-domain reflections (TDR). Measurement accuracy is within ±5%.

Timing Model

The high-performance multi-level FastTrack Interconnect routing resources ensure predictable performance, accurate simulation, and accurate timing analysis. The predictable performance of Mercury devices offer an advantage over FPGAs, which use a segmented connection scheme and therefore have unpredictable performance.

Figure 36 shows the timing model for bidirectional IOE pin timing. All registers are within the IOE.

Figure 36. Synchronous Bidirectional Pin External Timing Model



Tables 44 and 45 describe the Mercury device's external timing parameters.

Table 44. Mercury External Timing Parameters *Notes (1), (2)*

Symbol	Parameter	Conditions
t_{INSU}	Setup time with global clock at IOE register	
t_{INH}	Hold time with global clock at IOE register	
t_{OUTCO}	Clock-to-output delay with global clock at IOE register	C1 = 35 pF
$t_{INSUPLL}$	Setup time with PLL clock at IOE input register	
t_{INHPLL}	Hold time with PLL clock at IOE input register	
$t_{OUTCOPLL}$	Clock-to-output delay with PLL clock at IOE output register	C1 = 35 pF

Table 45. Mercury External Bidirectional Timing Parameters *Notes (1), (2)*

Symbol	Parameter	Conditions
$t_{INSUBIDIR}$	Setup time for bidirectional pins with gobal clock at IOE input register	
$t_{INHBIDIR}$	Hold time for bidirectional pins with global clock at IOE input register	
$t_{OUTCOBIDIR}$	Clock-to-output delay for bidirectional pins with global clock at IOE output register	C1 = 35 pF
$t_{XZBIDIR}$	Synchronous IOE output enable register to output buffer disable delay	C1 = 35 pF
$t_{ZXBIDIR}$	Synchronous IOE output enable register output buffer enable delay	C1 = 35 pF
$t_{INSUBIDIRPLL}$	Setup time for bidirectional pins with PLL clock at IOE input register	
$t_{INHBIDIRPLL}$	Hold time for bidirectional pins with PLL clock at IOE input register	
$t_{OUTCOBIDIRPLL}$	Clock-to-output delay for bidirectional pins with PLL clock at IOE output register	C1 = 35 pF
$t_{XZBIDIRPLL}$	Synchronous IOE output enable register to output buffer disable delay with PLL	C1 = 35 pF
$t_{ZXBIDIRPLL}$	Synchronous IOE output enable register output buffer enable delay with PLL	C1 = 35 pF

Notes to Tables 44 and 45:

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

(2) All timing parameters are either to and / or from pins, including global clock pins.

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

Copyright © 2003 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and / or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, mask work rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

I.S. EN ISO 9001