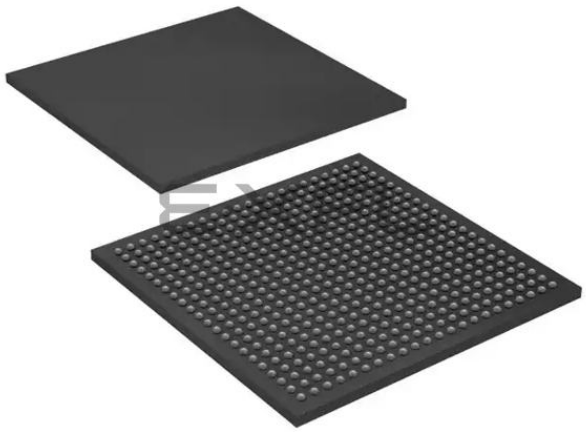


Welcome to E-XFL.COM

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	-40°C ~ 100°C (TJ)
Package / Case	484-BBGA, FCBGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep1m120f484i6n

Mercury device I/O pins are evenly distributed across the entire device area; other Altera device families have I/O pins placed on the device periphery. Mercury device I/O pin placement allows for higher I/O count at a given die size; pad size is no longer a limiting issue. Each I/O pin is fed by an IOE. IOEs are grouped in IOE row bands from the top to the bottom of the device. IOE row bands are separated by several LAB rows. LABs from the associated LAB row closest to the I/O row band drive IOEs through the local interconnect. This feature allows fast clock-to-output times when a pin is driven by any of the 10 LEs in the adjacent associated LAB. Each IOE contains a bidirectional buffer along with an input register, output register, output enable (OE) register, and input latch for DDR. When used with a global clock, these dedicated registers provide exceptional bidirectional I/O performance.

IOEs provide a variety of features, such as 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; output drive strength control; slew-rate control; tri-state buffers; bus-hold circuitry; programmable pull-up resisters; programmable input and output delays; and open-drain outputs. Mercury devices offer enhanced I/O support, including support for 1.8-V I/O, 2.5-V I/O, LVCMOS, LVTTL, HSTL, LVPECL, 3.3-V PCML, 3.3-V PCI, PCI-X, LVDS, GTL+, SSTL-2, SSTL-3, CTT, and 3.3-V AGP I/O standards. CDR (up to 1.25 Gbps) and source-synchronous (up to 840 Mbps) transfers are supported with HSDI circuitry for LVDS, LVPECL, and 3.3-V PCML I/O standards.

The ESB can implement a variety of memory functions, including CAM, quad-port RAM, true dual-port RAM, dual- and single-port RAM, ROM, and FIFO functions. ESBs are grouped into two rows: one at the top and one at the bottom of the device. Embedding the memory directly into the die improves performance and reduces die area compared to distributed-RAM implementations. Moreover, the abundance of cascadable ESBs, in conjunction with the ability for one ESB to implement two separate memory blocks, ensures that the Mercury device can implement multiple wide memory blocks for high-density designs. The ESB's high speed ensures the implemention of small memory blocks without any speed penalty. The abundance of ESBs ensures that designers can create as many different-sized memory blocks as the system requires. Figure 1 shows an overview of the Mercury device.

Mercury devices provide four dedicated clock input pins and six dedicated fast I/O pins that globally drive register control inputs, including clocks. These signals ensure efficient distribution of high-speed, low-skew control signals. The control signals use dedicated routing channels to provide short delays and low skew. The dedicated fast signals can also be driven by internal logic, providing an ideal solution for a clock divider or internally generated asynchronous control signal with high fan-out. The dedicated clock and fast I/O pins on Mercury devices can also feed logic. Dedicated clocks can also be used with the Mercury general purpose PLLs for clock management.

Each I/O row band also provides two additional I/O pins that can drive two row-global signals. Row-global signals can drive register control inputs for the LAB row associated with that particular I/O row band.

High-Speed Differential Interface

The top I/O or HSDI band in Mercury devices contains dedicated circuitry for supporting differential standards at speeds up to 1.25 Gbps. Mercury devices have dedicated differential buffers and circuitry to support LVDS, LVPECL, and 3.3-V PCML I/O standards. Two dedicated high-speed PLLs (separate from the general purpose PLLs) multiply reference clocks and drive high-speed differential serializer/deserializer channels. In addition, clock recovery units (CRUs) at each receiver channel enable CDR. EP1M120 devices support eight input channels, eight output channels, and two dedicated clock inputs for feeding the receiver and/or transmitter PLLs. EP1M350 devices support 18 input channels, 18 output channels, and two dedicated clock inputs.

Mercury devices have optional built-in 100-Ω termination resistors on HSDI differential receiver data pins and the `HSDI_CLK1` and `HSDI_CLK2` pins.

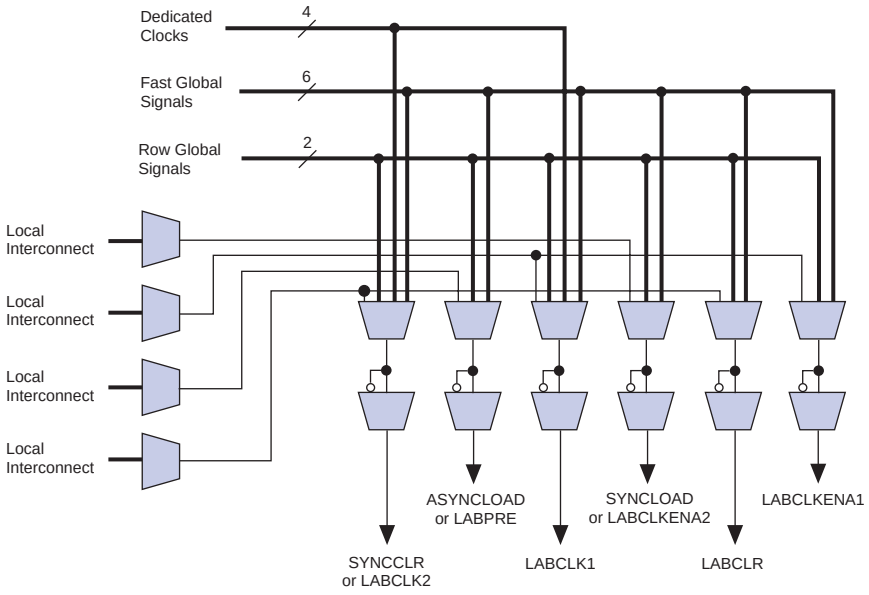
Designers can use the HSDI circuitry for the following applications:

- Gigabit Ethernet backplanes
- ATM, SONET
- RapidIO
- POS-PHY Level 4
- Fibre Channel
- SDTV

The HSDI band supports one of two possible modes:

- Source-synchronous mode
- Clock data recovery (CDR) mode

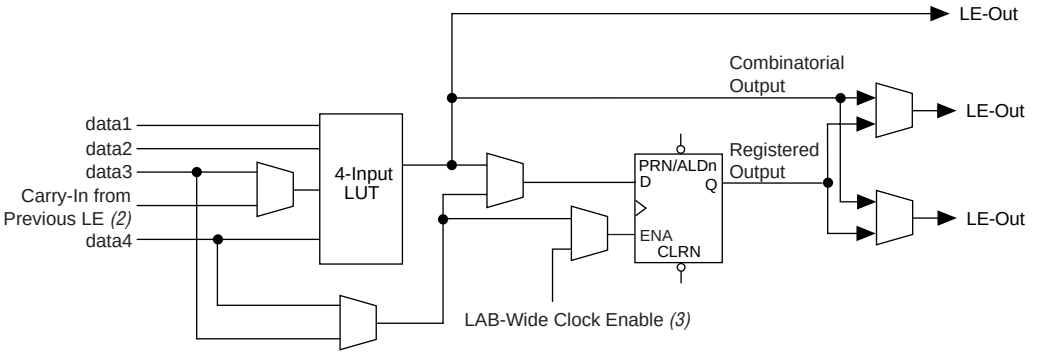
Figure 6. LAB-Wide Control Signals



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 8. Normal-Mode LE *Note (1)*



Notes to Figure 8:

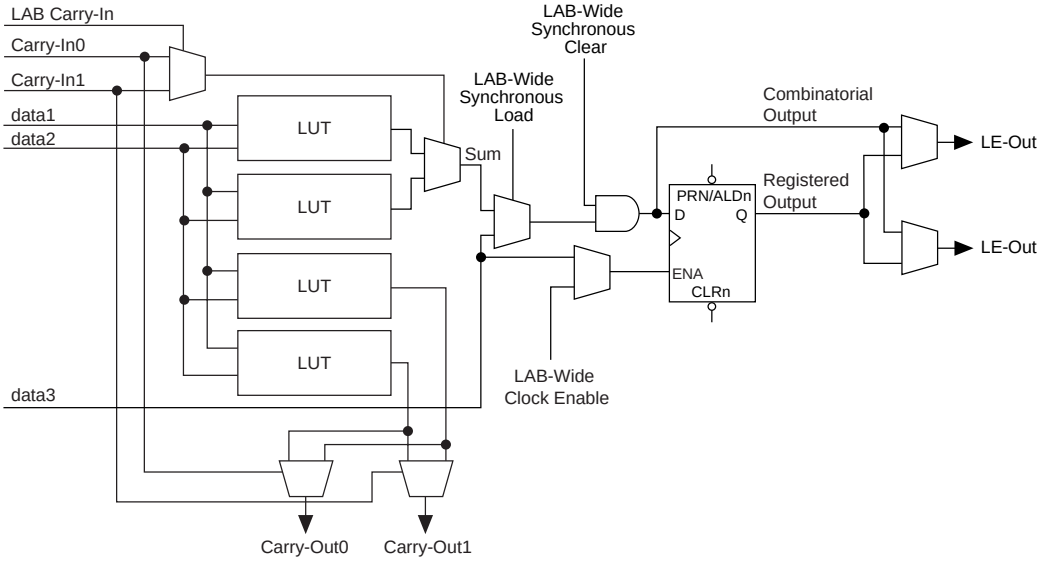
(1) LEs in normal mode support register packing.
(2) When using the carry-in in normal mode, the packed register feature is unavailable.
(3) There are two LAB-wide clock enables per LAB in addition to LE-specific clock enables.

Arithmetic Mode

The arithmetic mode is ideal for implementing adders, accumulators, and comparators. A LE in arithmetic mode contains four 2-input LUTs. The first two 2-input LUTs compute two summations based on a possible carry of `1` or `0`; the other two LUTs generate carry outputs for the two possible chains of the carry-select look-ahead (CSLA) circuitry. As shown in Figure 9, the LAB carry-in signal selects the appropriate carry-in chain (either `carry-in0` or `carry-in1`). The logic level of the chain selected in turn selects which parallel sum is generated as a combinatorial or registered output. For example, when implementing an adder, this output is the signal comprised of the sum `data1` + `data2` + *carry*, where *carry* is `0` or `1`. 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; `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. Figure 9 shows a Mercury LE in arithmetic mode.

The arithmetic mode also offers clock enable, counter enable, synchronous up / down control, synchronous clear, and synchronous load options. The counter enable and synchronous up / down control signals are generated from the data inputs of the LAB local interconnect. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. Consequently, if any of the LEs in a LAB use the counter mode, other LEs in that LAB must be used as part of the same counter or be used for a combinatorial function. The Quartus II software automatically places any registers that are not used by the counter into other LABs.

Figure 9. Arithmetic Mode LE



Carry-Select Look-Ahead Chain

The CSLA chain provides a very fast carry-forward function between LEs in arithmetic mode or multiplier mode. The CSLA chain uses the redundant carry calculation to increase the speed of carry functions. The LE can calculate sum and carry values 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 drive forward into the higher-order bit via the parallel carry chain and feed into both the LUT and the next portion of the CSLA chain. CSLA chains can begin in any LE within a LAB.

The CSLA chain's speed advantage results from the parallel pre-computation of carry chains. Instead of including every LUT in the critical path, only the propagation delays between LAB carry-in generation circuits (LE 4 and LE 10) make up the critical path. This feature allows the Mercury architecture to implement high-speed counters, adders, multipliers, parity functions, and comparators of arbitrary width.

Figure 10 shows the CSLA circuitry in a 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. A lab-wide carry-in bit selects which chain is used for the addition of given inputs. The actual carry-in signal for that selected chain, `carry-in0` or `carry-in1`, selects the carry-out to carry forward, which is routed to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it is driven to local, row, or column interconnects.

For a typical 16 × 16-bit binary tree multiplier, five stages are needed to determine the final product. The Mercury LE multiplier mode allows the partial product formation stage (Stage 1) and the first sum of stages (Stage 2) to be combined in a single stage, shown in Figure 13. This feature, combined with the direct connection between RapidLAB lines and LEs in multiplier mode, allows the fast dedicated implementation of multipliers.

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.

Table 9 summarizes how various elements of the Mercury architecture drive each other.

Table 9. Mercury Routing Scheme

Source	Destination										
	LE	Local Interconnect	IOE	ESB Row Interconnect	ESB	Row	Priority Row	RapidLAB Interconnect	Column	Priority Column	Leap Lines
LE	✓ (1)	✓				✓	✓	✓	✓	✓	✓
Local Interconnect	✓		✓								
IOE		✓ (2)				✓ (3)	✓ (3)		✓	✓	
ESB Row Interconnect					✓						
ESB				✓					✓	✓	✓
Row		✓									
Priority Row		✓									
RapidLAB Interconnect	✓ (4)	✓									
Column				✓		✓	✓		✓		
Priority Column				✓			✓	✓	✓	✓	
Leap Lines				✓		✓	✓	✓	✓		

Notes to Table 9:

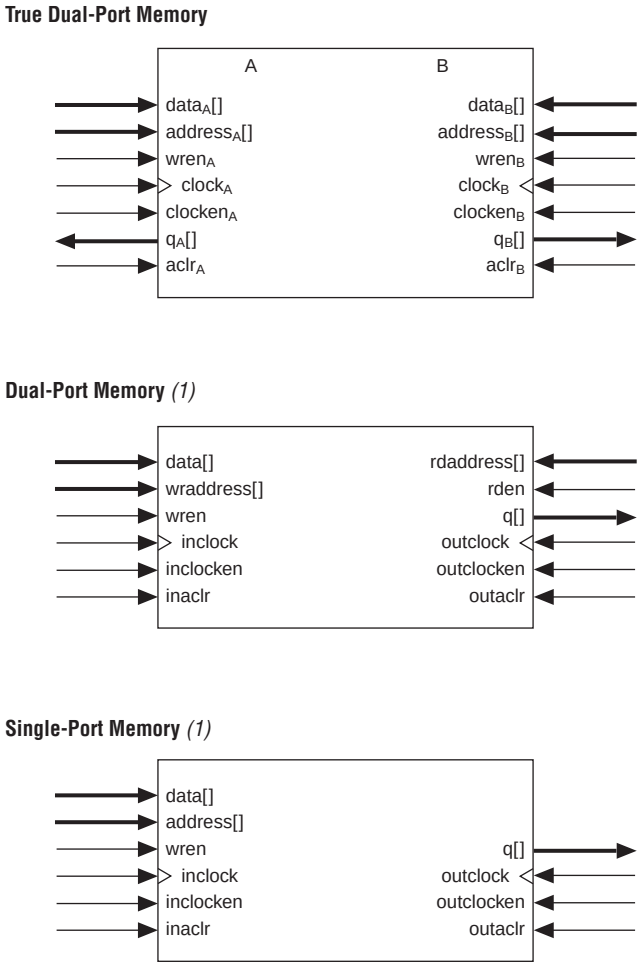
(1) This direct connection is possible through the FastLUT connection.
(2) IOEs can connect to the adjacent LAB's local interconnects in the associated LAB row.
(3) IOEs can connect to row and priority row interconnects in the associated LAB row.
(4) This connection is used for multiplier mode.

Embedded System Block

The ESB can implement various types of memory blocks, including quad-port, true dual-port, dual- and single-port RAM, ROM, FIFO, and CAM blocks.

The ESB includes input and output registers; the input registers synchronize reads and/or writes, and the output registers can pipeline designs to further increase system performance. The ESB offers a quad port mode, which supports up to four port operations, two reads and two writes simultaneously, with the ability for a different clock on each of the four ports. Figure 17 shows the ESB quad-port block diagram.

Figure 18. RAM Memory Port Configurations



Note to Figure 18:
(1) Two dual- or single-port memory blocks can be implemented in a single ESB.

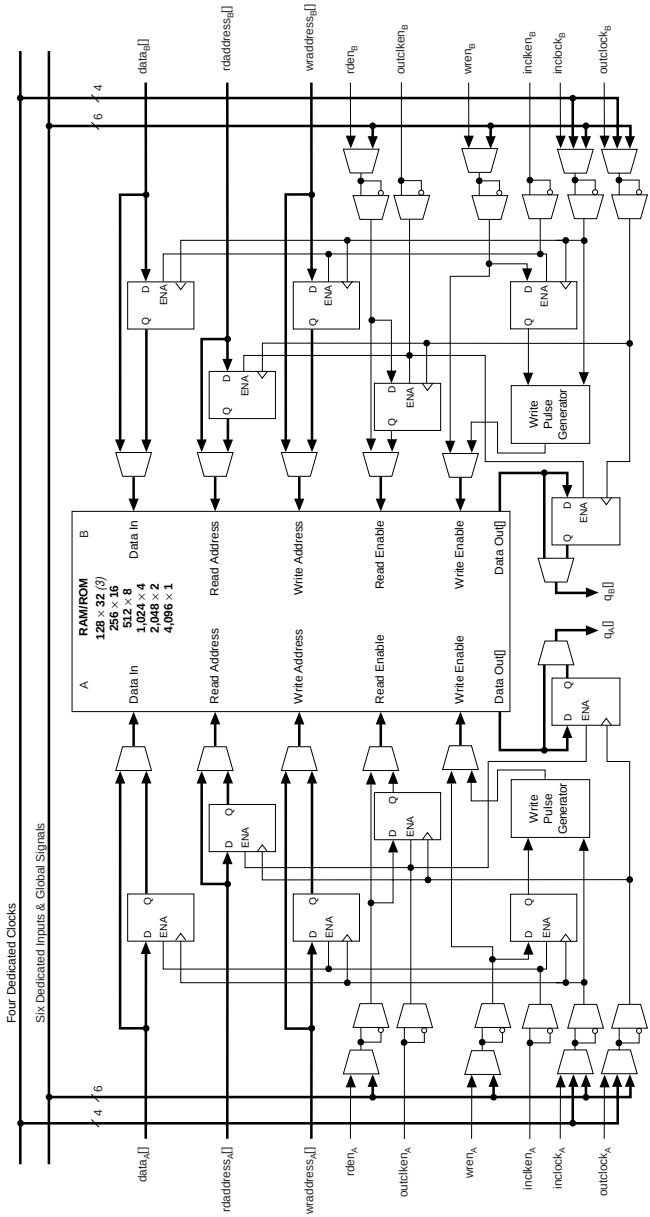
The ESB also allows variable width data ports for reading and writing to any of the RAM ports in any RAM configuration. For example, the ESB in quad port configuration can be written in ×1 mode at port A, read in ×16 from port A, written in ×4 mode at port B, and read in ×2 mode from port B.

The ESB implements two forms of clocking modes for quad-port and dual-port memory—read / write clock mode and input / output clock mode.

Read/Write Clock Mode

An ESB implementing quad-port memory in read / write clock mode can use up to four clocks. For port A, one clock controls all registers associated with writing: data input, WE, and write address. The other clock controls all registers associated with reading: read enable (RE), read address, and data output. Another set of clocks can be used for port B of the RAM, or the same clocks can be used. Each ESB port, A or B, also supports independent read clock enable, write clock enable, and asynchronous clear signals. Read / write clock mode is commonly used for applications where reads and writes occur at different system frequencies. Figure 19 shows the ESB in read / write clock mode.

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



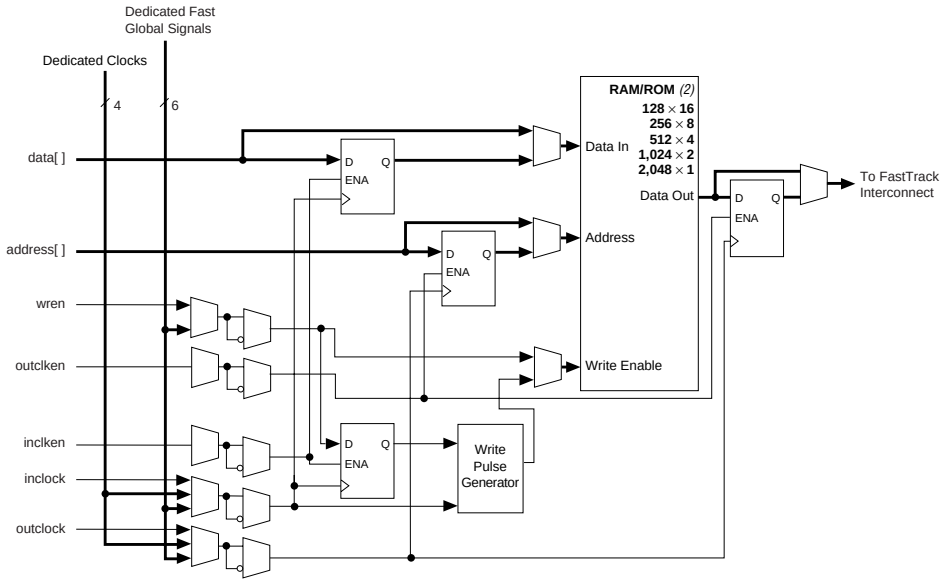
Notes to Figure 19:

(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.

Single-Port Mode

The Mercury device's ESB also supports a single-port mode, which is used when simultaneous reads and writes are not required. See Figure 21. A single ESB can support up to two single-port mode RAMs.

Figure 21. ESB in Single-Port Mode *Note (1)*



Notes to Figure 21:

(1) All registers can be asynchronously cleared by ESB local interconnect signals, global signals, or chip-wide reset.

(2) If there is only one single-port RAM block in an ESB, it can support the following configurations: 4,096 × 1; 2,048 × 2; 1,028 × 4; 512 × 8; 256 × 16; or 128 × 32.

Content-Addressable Memory

Mercury devices can implement CAM in ESBs. CAM can be thought of as the inverse of RAM. RAM stores data in a specific location; when the system submits an address, the RAM block provides the data. Conversely, when the system submits data to CAM, the CAM block provides the address where the data is found. For example, if the data FA12 is stored in address 14, the CAM outputs 14 when FA12 is driven into it.

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)*.

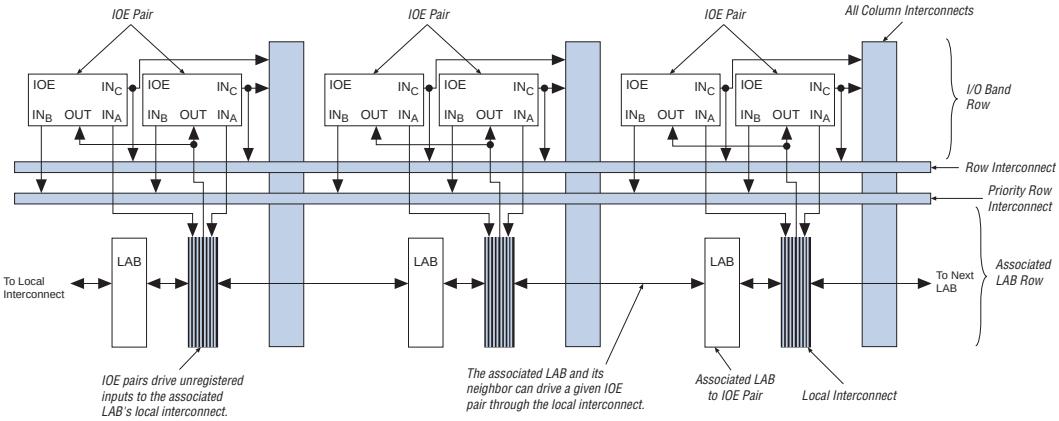
Driving into ESBs

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.

Each row of I/O pins has an associated LAB row for driving to and from the core of the Mercury device. For a given I/O band row, its associated LAB row is located below it with the exception of the bottom I/O band row. The bottom I/O band is located at the bottom periphery of the device, hence its associated LAB row is located above it. Figure 29 shows an example of an I/O band to associated LAB row interconnect in a Mercury device.

There is a maximum of two IOEs associated with each LAB in the associated LAB row. The local interconnect of the associated LAB drives the IOEs. Since local interconnect is shared with the LAB neighbor, any given LAB can directly drive up to four IOEs. The local interconnect drives the data and OE signals when the IOE is used as an output or bidirectional pin.

Figure 29. IOE Connection to Interconnects and Adjacent LAB *Note (1)*



Note to Figure 29:

(1) IN_A: unregistered input; IN_B: registered / unregistered input; IN_C: registered / unregistered input or OE register output in DDR mode.

The IOEs drive registered or combinatorial versions of input data into the device. The unregistered input data can be driven to the local interconnect (for fast input setup), row and priority row interconnect, and column and priority column interconnects. The registered data can also be driven to the same row and column resources. The OE register output can be fed back through column and row interconnects to implement DDR I/O pins.

For the HSDI I/O band, half of the dedicated banks support LVDS, 3.3-V PCML or LVPECL, and receiver inputs, while the other half support LVDS, PCML or LVPECL, and transmitter outputs. A single device can support 1.5-V, 1.8-V, 2.5-V, and 3.3-V interfaces; each bank can support a V_{CCIO} standard independently. Each bank can also use a separate V_{REF} level so that each bank can support any of the terminated standards (such as SSTL-3) independently. A bank can support a single V_{REF} level. Each bank contains a fixed VREF pin for voltage referenced standards. This pin can be used as a regular I/O if a V_{REF} standard is not used. Table 13 shows the number of I/O banks in each Mercury device.

Table 13. Number of I/O Banks per Device

Device	Regular I/O Banks	HSDI Band I/O Banks
EP1M120	8	4
EP1M350	12	4

Each bank can support multiple standards with the same V_{CCIO} for output pins. For EP1M120 devices, each bank can support one voltage-referenced I/O standard, but can support multiple I/O standards with the same V_{CCIO} and V_{REF} voltage levels. 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. Figure 30 shows the I/O bank layout for an EP1M120 device. For EP1M350 devices, each bank can support two voltage-reverenced I/O standards; each I/O bank is split into two voltage-referenced sub-banks. When using the two HSDI transmitter banks as regular I/O banks in a non-HSDI mode, those two banks require the same V_{CCIO} level. However, each HSDI transmitter bank supports its own V_{REF} level.

Table 23. LVTTL Specifications *Note (10)*

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
I_I	Input pin leakage current	V_{IN} = 0 V or V_{CCIO}	–10	10	µA
V_{OH}	High-level output voltage	I_{OH} = –4 mA	2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 4 mA		0.45	V

Table 24. LVCMOS Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Units
V_{CCIO}	Power supply voltage range		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
I_I	Input pin leakage current	V_{IN} = 0 V or V_{CCIO}	–10	10	µA
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 25. 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
I_I	Input pin leakage current	V_{IN} = 0 V or V_{CCIO}	10	10	µA
V_{OH}	High-level output voltage	I_{OH} = –0.1 mA	2.1		V
		I_{OH} = –1 mA	2.0		V
		I_{OH} = –2 mA	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 mA		0.7	V

Tables 46 through 51 show external timing parameters for Mercury devices.

Table 46. EP1M120 External Timing Parameters *Note (1)*

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	0.67		0.70		0.73		ns
t_{INH}	0.00		0.00		0.00		ns
t_{OUTCO}	2.00	3.30	2.00	3.32	2.00	3.49	ns
$t_{INSUPLL}$	0.59		0.64		0.62		ns
t_{INHPLL}	0.00		0.00		0.00		ns
$t_{OUTCOPLL}$	0.50	2.08	0.50	2.08	0.50	2.15	ns

Table 47. EP1M120 External Bidirectional Timing Parameters *Note (1)*

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	0.67		0.70		0.73		ns
$t_{INHBIDIR}$	0.00		0.00		0.00		ns
$t_{OUTCOBIDIR}$	2.00	3.30	2.00	3.32	2.00	3.49	ns
$t_{XZBIDIR}$		3.52		3.53		3.74	ns
$t_{ZXBIDIR}$ *(2)*		3.52		3.53		3.74	ns
$t_{ZXBIDIR}$ *(3)*		3.72		3.73		3.99	ns
$t_{INSUBIDIRPLL}$	0.59		0.64		0.62		ns
$t_{INHBIDIRPLL}$	0.00		0.00		0.00		ns
$t_{OUTCOBIDIRPLL}$	0.50	2.08	0.50	2.08	0.50	2.15	ns
$t_{XZBIDIRPLL}$		2.29		2.29		2.39	ns
$t_{ZXBIDIRPLL}$ *(2)*		2.29		2.29		2.39	ns
$t_{ZXBIDIRPLL}$ *(3)*		2.49		2.49		2.64	ns

Table 51. EP1M350 External Bidirectional Timing Parameters *Note (1)*

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	0.60		0.57		0.71		ns
$t_{INHBIDIR}$	0.00		0.00		0.00		ns
$t_{OUTCOBIDIR}$	2.00	3.95	2.00	3.97	2.00	4.75	ns
$t_{XZBIDIR}$		3.90		3.93		4.70	ns
$t_{ZXBIDIR}$ *(2)*		3.90		3.93		4.70	ns
$t_{ZXBIDIR}$ *(3)*		4.10		4.13		4.94	ns
$t_{INSUBIDIRPLL}$	0.69		0.70		0.82		ns
$t_{INHBIDIRPLL}$	0.00		0.00		0.00		ns
$t_{OUTCOBIDIRPLL}$	0.50	2.23	0.50	2.23	0.50	2.69	ns
$t_{XZBIDIRPLL}$		2.19		2.18		2.63	ns
$t_{ZXBIDIRPLL}$ *(2)*		2.19		2.18		2.63	ns
$t_{ZXBIDIRPLL}$ *(3)*		2.39		2.38		2.87	ns

Notes to Tables 46 – 51:

(1) Timing will vary by I/O pin placement. Therefore, use the Quartus II software to determine exact I/O timing for each pin.

(2) This parameter is measured with the **Increase t_{ZX} Delay to Output Pin** option set to **Off**.

(3) This parameter is measured with the **Increase t_{ZX} Delay to Output Pin** option set to **On**.

Power Consumption

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

Configuration & Operation

The Mercury architecture supports several configuration schemes. This section summarizes the device operating modes and available device configuration schemes.

Operating Modes

The Mercury 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.

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