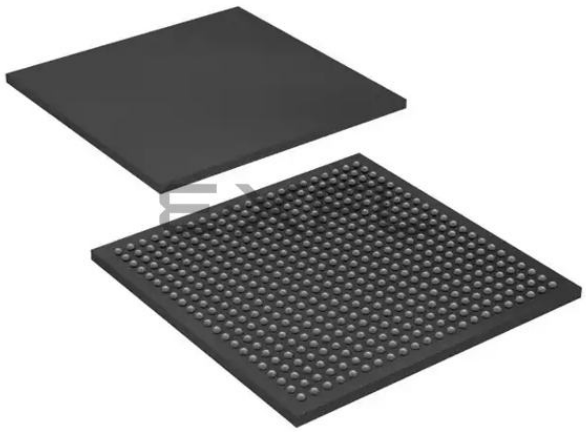


Welcome to E-XFL.COM

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/ep1m120f484c6n

Figure 4. Receiver & Transmitter Diagrams for CDR Mode *Notes (1), (2)*

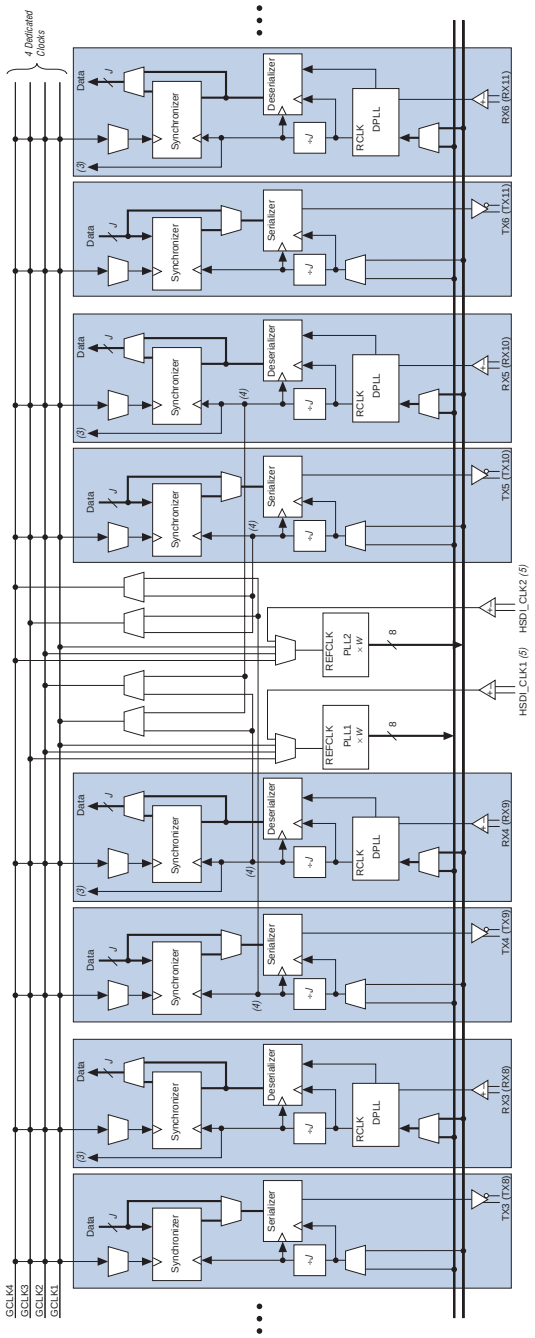
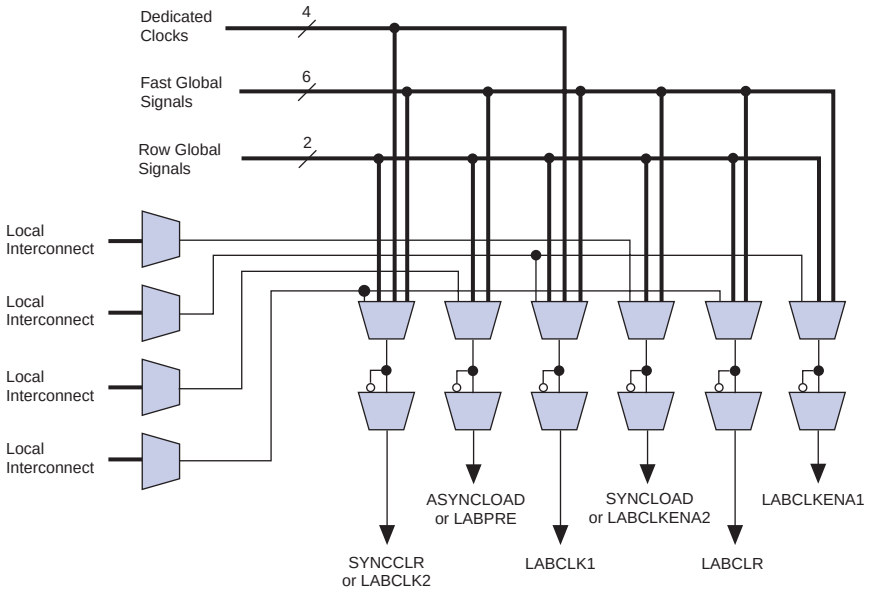


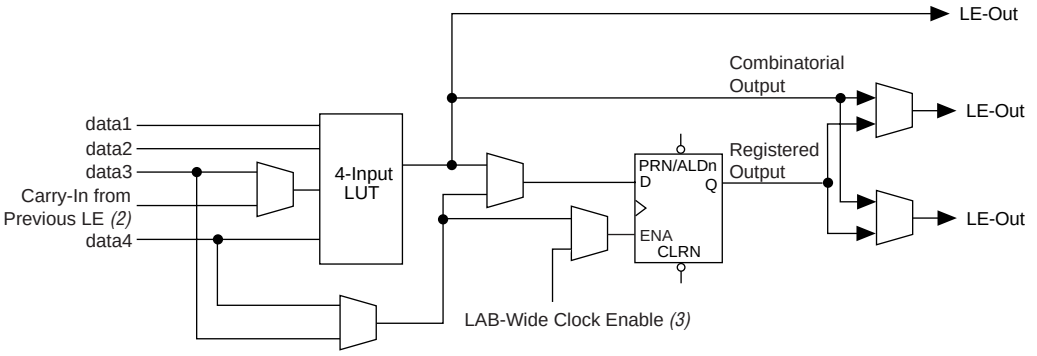
Figure 6. LAB-Wide Control Signals



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)*



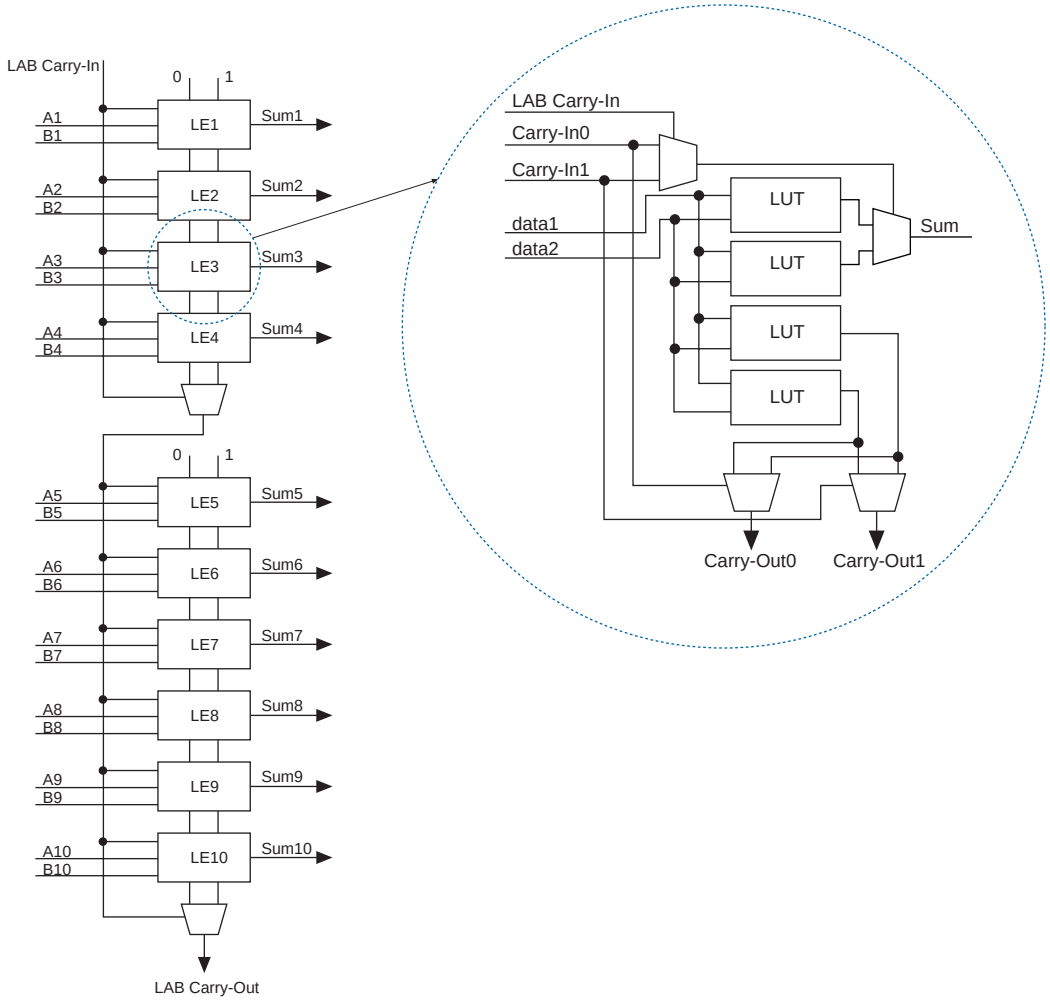
Notes to Figure 8:

(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.

Figure 10. CSLA Details



The Quartus II Compiler can create CSLA logic automatically during design processing. Alternatively, the designer can create CSLA logic manually during design entry. Parameterized functions such as library of parameterized modules (LPM) and DesignWare functions automatically take advantage of carry chains for the appropriate functions.

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.

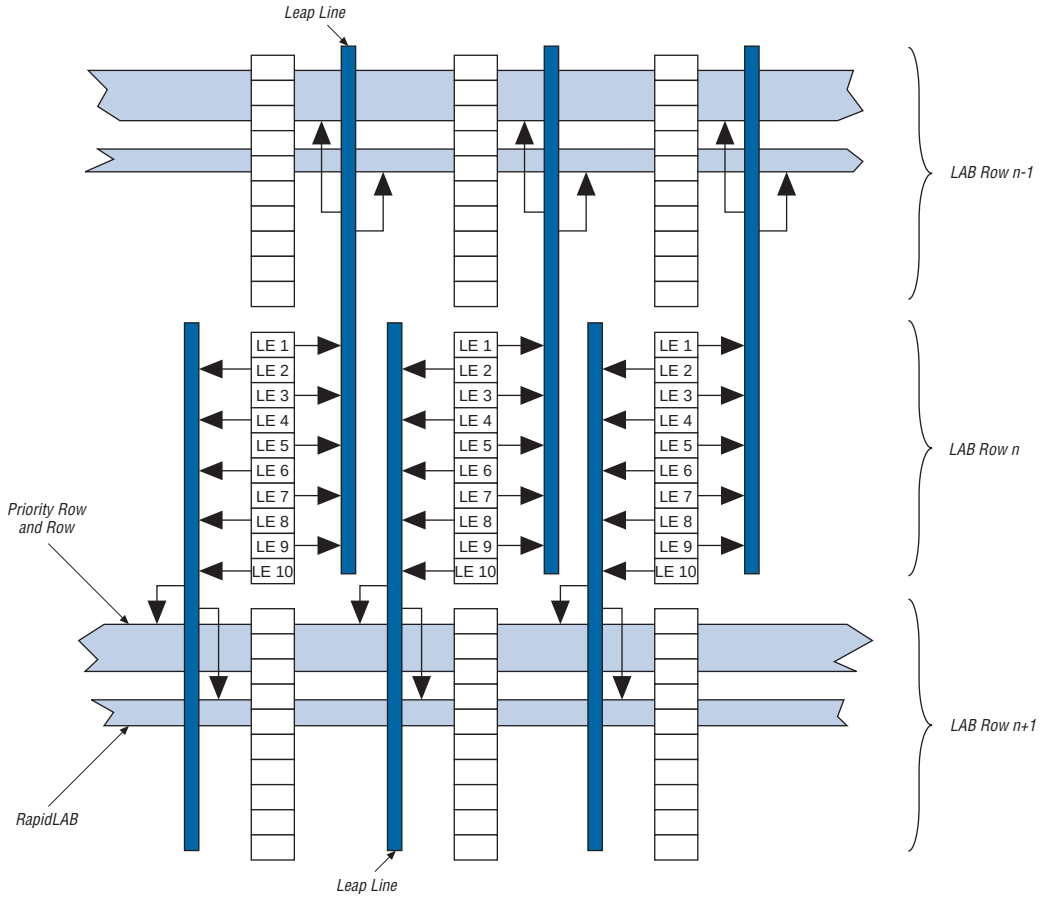
The RapidLAB interconnect has dedicated fast connections to the LE inputs in multiplier mode, further increasing the speed of the multiplier. These dedicated connections allow RapidLAB lines to avoid delay incurred by driving onto local interconnects and then into the LE.

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 15. Leap Line Interconnect



FastLUT Interconnect

Mercury devices include an enhanced interconnect structure within LABs for faster routing of LE output to LE input connections. The FastLUT connection allows the combinatorial output of an LE to directly drive the fast input of the LE directly below it, bypassing the local interconnect. This resource can be used as a high speed connection for wide fan-in functions from LE 1 to LE 10 in the same LAB. Figure 16 shows a FastLUT interconnect.

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.

ESBs can implement synchronous RAM, which is easier to use than asynchronous RAM. A circuit using asynchronous RAM must generate the RAM write enable (WE) signal while ensuring that its data and address signals meet setup and hold time specifications relative to the WE signal. In contrast, the ESB's synchronous RAM generates its own WE signal and is self-timed with respect to the global clock. Circuits using the ESB's self-timed RAM must only meet the setup and hold time specifications relative to the global clock.

ESBs are grouped together in rows at the top and bottom of the device for fast horizontal communication. The ESB row interconnect can be driven by any ESB in the row. The row interconnect drives the ESB local interconnect, which in turn drives the ESB ports. ESB outputs drive the ESB local interconnect, which can drive row interconnect as well as all types of column interconnect, including leap lines. The leap lines allow fast access between ESBs and the adjacent LAB row.

When implementing memory, each ESB can be configured in any of the following sizes for quad port and true dual-port memory modes: 256 × 16; 512 × 8; 1,024 × 4; 2,048 × 2; or 4,096 × 1. For dual-port and single-port modes, the ESB can be configured for 128 × 32 in addition to the list above. For variable port width RAMs, any port width ratio combination must be 1, 2, 4, 8, or 16. For example, a RAM with data ports of width 1 and 16 or 2 and 32 will work, but not 1 and 32.

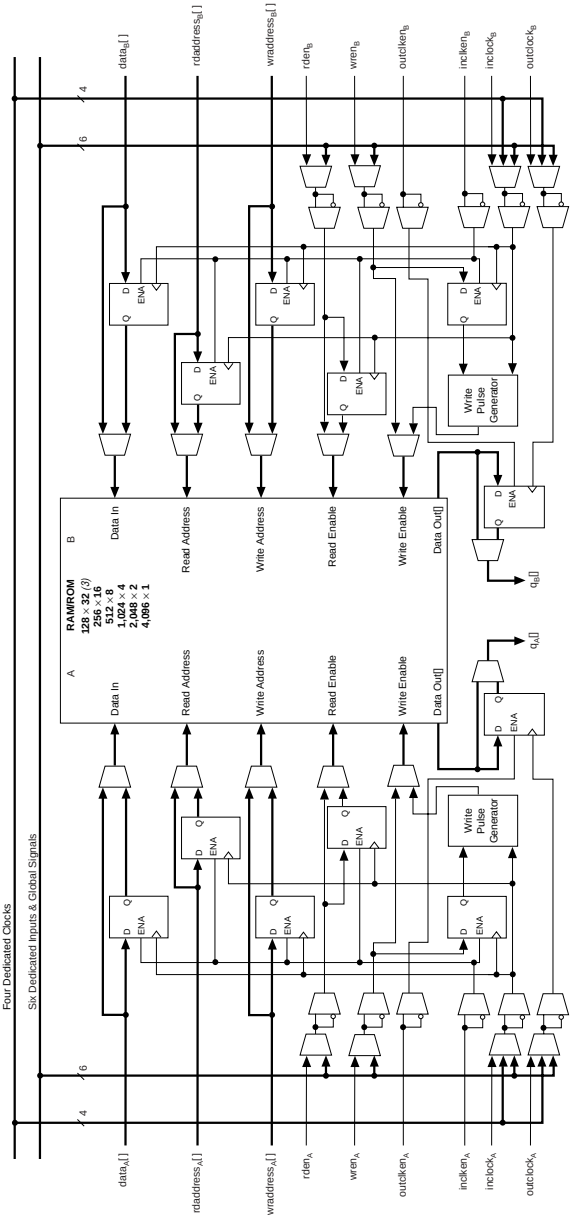
The ESB can also be split in half and used for two independent 2,048-bit single-port or dual-port RAM blocks. For example, one half of the ESB can be used as a 128 × 16 memory single-port memory while the other half can be used for a 1,024 × 2 dual-port memory. This effectively doubles the number of RAMs a Mercury device can implement for its given number of ESBs. The Quartus II software automatically merges two logical memory functions in a design into an ESB; the designer does not need to merge the functions manually.

By combining multiple ESBs, the Quartus II software implements larger memory blocks automatically. For example, two 256 × 16 RAM blocks can be combined to form a 256 × 32 RAM block, and two 512 × 8 RAM blocks can be combined to form a 512 × 16 RAM block. Memory performance does not degrade for memory blocks up to 4,096 words deep. Each ESB can implement a 4,096-word-deep memory; the ESBs are used in parallel, eliminating the need for any external control logic and its associated delays. To create a high-speed memory block more than 4,096 words deep, the Quartus II software will automatically combine ESBs with LE control logic.

Input/Output Clock Mode

An ESB using input/output clock mode can also use up to four clocks. On each of the two ports, A or B, one clock controls all registers for inputs into the ESB: data input, WE, RE, read address, and write address. The other clock controls the ESB data output registers. Each ESB port, A or B, also supports independent read clock enable, write clock enable, and asynchronous clear signals. Input/output clock mode is commonly used for applications where the reads and writes occur at the same system frequency, but require different clock enable signals for the input and output registers. Figure 20 shows the ESB in input/output clock mode.

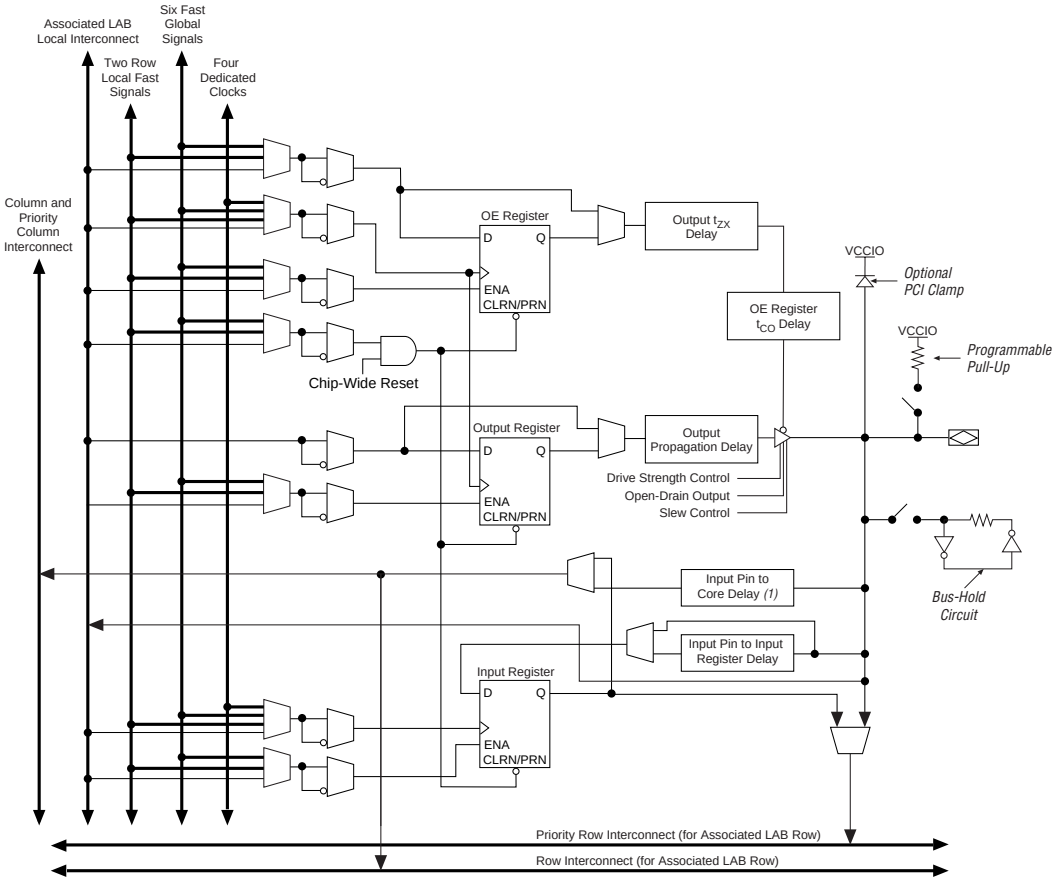
Figure 20. ESB in Input/Output Clock Mode *Notes (1), (2)*



Notes to Figure 20:

(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.

Figure 26. Mercury IOE



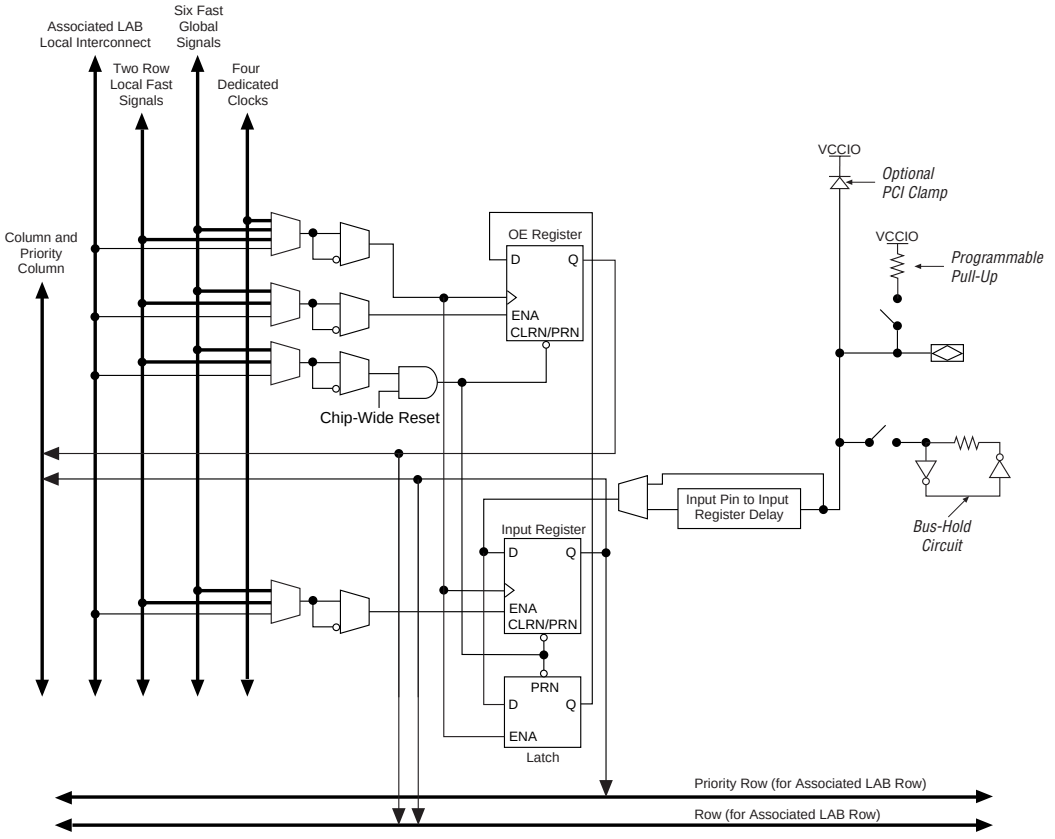
Note to Figure 26:
(1) This programmable delay has four settings: off and three levels of delay.

Double Data Rate I/O

Mercury device's have three register IOEs to support the DDRIO feature, which makes double data rate interfaces possible by clocking data on both positive and negative clock edges. The IOE in Mercury devices supports double data rate input and double data rate output modes.

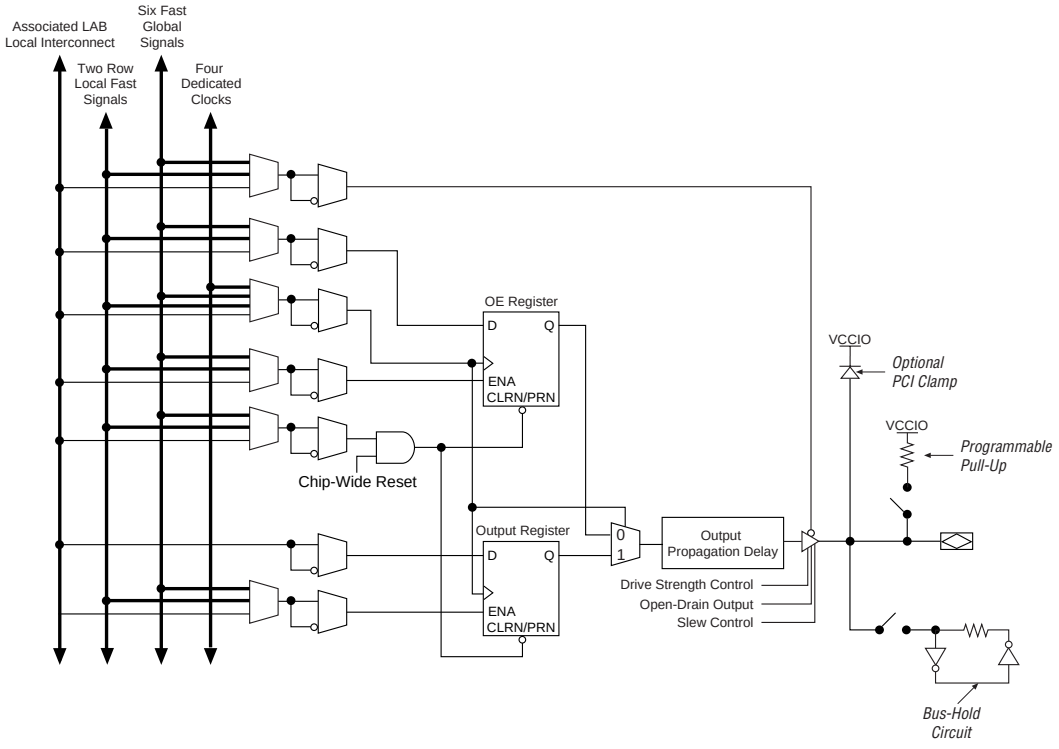
In Mercury device IOEs, the OE register is a multi-purpose register available as a second input or output register. When using the IOE for double data rate inputs, the input register and OE register are automatically configured as input registers to clock input double rate data on alternating edges. An input latch is also used within the IOE for DDR input acquisition. The latch holds the data that is present during the clock high times, driving it to the OE register. This allows the OE register and input register to clock both bits of data into LEs, synchronous to the same clock edge (either rising or falling). Figure 27 shows an IOE configured for DDR input.

Figure 27. IOE Configured for DDR Input



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.

Dedicated Fast Lines & I/O Pins

Mercury devices incorporate dedicated bidirectional pins for signals with high internal fanout, such as PCI control signals. These pins are called dedicated fast I/O pins (FAST1, FAST2, FAST3, FAST4, FAST5, and FAST6) and can drive the six global fast lines throughout the device, ideal for fast clock, clock enable, clear, preset, or high fanout logic signal distribution. The dedicated fast I/O pins have the same IOE as a regular I/O pin. The dedicated fast lines can also be driven by a LE local interconnect to generate internal global signals.

In addition to the device global fast lines, each LAB row has two dedicated fast lines local to the row. This is ideal for high fanout control signals for a section of a design that may fit into a single LAB row. Each I/O band (with the exception of the top I/O band) has two dedicated row-global fast I/O pins to drive the row-global fast resources for the associated LAB. The dedicated local fast I/O pins have the same IOE as a regular I/O pin. The LE local interconnect can drive dedicated row-global fast lines to generate internal global signals specific to a row. There are no pin connections for buried LAB rows; LE local interconnects drive the row-global signals in those rows.

I/O Standard Support

Mercury device IOEs support the following I/O standards:

- LVTTL
- LVCMOS
- 1.8-V
- 2.5-V
- 3.3-V PCI
- 3.3-V PCI-X
- 3.3-V AGP (1×, 2×)
- LVDS
- LVPECL
- 3.3-V PCML
- GTL+
- HSTL class I and II
- SSTL-3 class I and II
- SSTL-2 class I and II
- CTT

Lock Signals

The Mercury device general purpose PLL circuits support individual LOCK signals. The LOCK signal drives high when the PLL has locked onto the input clock. Lock remains high as long as the input remains within specification. It will go low if the input is out of specification. A LOCK pin is optional for each PLL used in the Mercury devices; when not used, they are I/O pins. This signal is not available internally; if it is used in the core, it must be fed back in with an input pin.

SignalTap Embedded Logic Analyzer

Mercury devices include device enhancements to support the SignalTap embedded logic analyzer. By including this circuitry, the Mercury device provides the ability to monitor design operation over a period of time through the IEEE Std. 1149.1 JTAG circuitry; a designer can analyze internal logic at speed without bringing internal signals to the I/O pins. This feature is particularly important for advanced packages such as FineLine BGA packages, because it can be difficult to add a connection to a pin during the debugging process after a board is designed and manufactured.

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All Mercury devices provide JTAG BST circuitry that complies with the IEEE Std. 1149.1-1990 specification. JTAG boundary-scan testing can be performed before or after configuration, but not during configuration. Mercury devices can also use the JTAG port for configuration with the Quartus II software or with hardware using either Jam Standard Test and Programming Language (STAPL) Files (**.jam**) or Jam STAPL Byte-Code Files (**.jbc**). Mercury devices also use the JTAG port to monitor the logic operation of the device with the SignalTap embedded logic analyzer. Mercury devices support the JTAG instructions shown in Table 16.

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

Table 26. 1.8-V I/O Specifications *Note (10)*					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Units**
V_{CCIO}	Output supply voltage		1.71	1.89	V
V_{IH}	High-level input voltage		$0.65 \times V_{CCIO}$	4.1	V
V_{IL}	Low-level input voltage		–0.5	$0.35 \times V_{CCIO}$	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} = –2 mA	V_{CCIO} – 0.45		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA		0.45	V

Figures 34 and 35 show receiver input and transmitter output waveforms, respectively, for all differential I/O standards (LVPECL, 3.3-V PCML, LVDS, and HyperTransport technology).

Figure 34. Receiver Input Waveforms for Differential I/O Standards

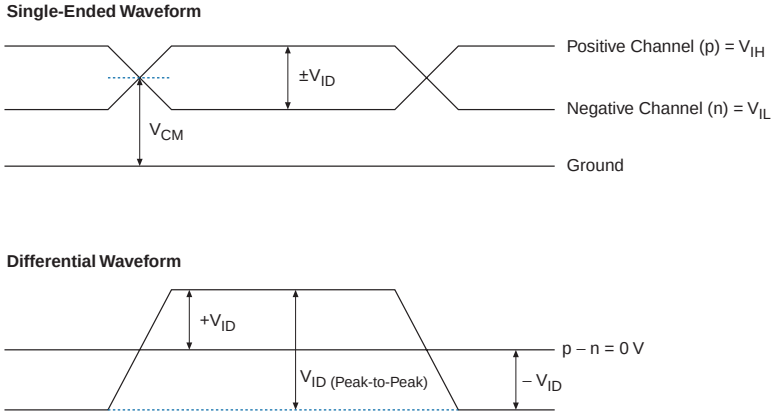


Table 39. 1.5-V HSTL Class I Specifications *Note (10)*

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

Table 48. EP1M120 External Timing Parameters *Note (1)*

Symbol	-7A Speed Grade		-8A Speed Grade		Unit
	Min	Max	Min	Max	
t_{INSU}	0.74		0.79		ns
t_{INH}	0.00		0.00		ns
t_{OUTCO}	2.00	3.50	2.00	4.10	ns
$t_{INSUPLL}$	0.62		0.75		ns
t_{INHPLL}	0.00		0.00		ns
$t_{OUTCOPLL}$	0.50	2.15	0.50	2.43	ns

Table 49. EP1M120 External Bidirectional Timing Parameters *Note (1)*

Symbol	-7A Speed Grade		-8A Speed Grade		Unit
	Min	Max	Min	Max	
$t_{INSUBIDIR}$	0.74		0.79		ns
$t_{INHBIDIR}$	0.00		0.00		ns
$t_{OUTCOBIDIR}$	2.00	3.50	2.00	4.10	ns
$t_{XZBIDIR}$		3.75		4.30	ns
$t_{ZXBIDIR}$ *(2)*		3.75		4.30	ns
$t_{ZXBIDIR}$ *(3)*		4.00		4.58	ns
$t_{INSUBIDIRPLL}$	0.62		0.75		ns
$t_{INHBIDIRPLL}$	0.00		0.00		ns
$t_{OUTCOBIDIRPLL}$	0.50	2.15	0.50	2.43	ns
$t_{XZBIDIRPLL}$		2.39		2.67	ns
$t_{ZXBIDIRPLL}$ *(2)*		2.39		2.67	ns
$t_{ZXBIDIRPLL}$ *(3)*		2.64		2.95	ns

Table 50. EP1M350 External Timing Parameters *Note (1)*

Symbol	-5 Speed Grade		-6 Speed Grade		-7 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	0.60		0.57		0.71		ns
t_{INH}	0.00		0.00		0.00		ns
t_{OUTCO}	2.00	3.95	2.00	3.97	2.00	4.75	ns
$t_{INSUPLL}$	0.69		0.70		0.82		ns
t_{INHPLL}	0.00		0.00		0.00		ns
$t_{OUTCOPLL}$	0.50	2.23	0.50	2.23	0.50	2.69	ns