



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



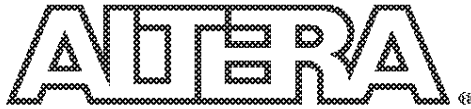
Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





FLEX 10KE

Embedded Programmable Logic Family

November 1998, ver. 1.01 Data Sheet

Features...

Preliminary Information

- Embedded programmable logic device (PLD) family, providing system integration in a single device
 - Enhanced embedded array for implementing megafunctions such as efficient memory and specialized logic functions
 - Dual-port capability up to 16-bit width per embedded array block (EAB)
 - Logic array for general logic functions
- Replaces FLEX® 10KB family
- High density
 - 30,000 to 250,000 typical gates (see Tables 1 and 2)
 - Up to 98,304 RAM bits; 4,096 bits per EAB, all of which can be used without reducing logic capacity
- System-level features
 - MultiVolt™ I/O pins can drive or be driven by 2.5-V, 3.3-V, or 5.0-V devices
 - Low power consumption
 - Fully compliant with the peripheral component interconnect Special Interest Group's (PCI-SIG) *PCI Local Bus Specification, Revision 2.1* for 3.3-V operation
 - Built-in Joint Test Action Group (JTAG) boundary-scan test (BST) circuitry compliant with IEEE Std. 1149.1-1990, available without consuming device logic

Feature	EPF10K30E	EPF10K50E	EPF10K100β	EPF10K100E
Typical gates (logic and RAM), <i>Note (1)</i>	30,000	50,000	100,000	100,000
Usable gates	22,000 to 119,000	36,000 to 199,000	62,000 to 158,000	62,000 to 257,000
Logic elements (LEs)	1,728	2,880	4,992	4,992
EABs	6	10	12	12
Total RAM bits	24,576	40,960	24,576	49,152
Maximum user I/O pins	220	254	191	338

Note:

(1) For designs that require IEEE Std. 1149.1 JTAG boundary-scan testing, the built-in JTAG circuitry contributes up to 31,250 additional gates.

Table 2 shows FLEX 10KE features for EPF10K130E, EPF10K200E, and EPF10K250E devices.

Feature	EPF10K130E	EPF10K200E	EPF10K250E
Typical gates (logic and RAM), <i>Note (1)</i>	130,000	200,000	250,000
Usable gates	82,000 to 342,000	123,000 to 513,000	149,000 to 474,000
Logic elements (LEs)	6,656	9,984	12,160
EABs	16	24	20
Total RAM bits	65,536	98,304	81,920
Maximum user I/O pins	413	470	470

Note:

- (1) For designs that require IEEE Std. 1149.1 JTAG boundary-scan testing, the built-in JTAG circuitry contributes up to 31,250 additional gates.

...and More Features

- Fabricated on advanced processes and operate with a 2.5-V internal supply voltage
- I/O circuits may be powered by 2.5-V or 3.3-V supply voltages
- In-circuit reconfigurability (ICR) via external Configuration EPROM, intelligent controller, or JTAG port
- ClockLock™ and ClockBoost™ options for reduced clock delay/skew and clock multiplication
- Built-in low-skew clock distribution trees
- 100% functional testing of all devices; test vectors or scan chains are not required
- Pull-up on I/O pins before and during configuration
- Supports hot-socketing operation
- ⊗ Flexible interconnect
 - FastTrack Interconnect™ continuous routing structure for fast, predictable interconnect delays
 - Dedicated carry chain that implements arithmetic functions such as fast adders, counters, and comparators (automatically used by software tools and megafunctions)
 - Dedicated cascade chain that implements high-speed, high-fan-in logic functions (automatically used by software tools and megafunctions)
 - Tri-state emulation that implements internal tri-state buses
 - Up to six global clock signals and four global clear signals
- ⊗ Powerful I/O pins
 - Individual tri-state output enable control for each pin
 - Open-drain option on each I/O pin
 - Programmable output slew-rate control to reduce switching noise
 - Clamp to V_{CC} user-selectable pin-by-pin
- ⊗ Peripheral register for fast setup and clock-to-output delay

- ❖ Software design support and automatic place-and-route provided by the Altera MAX+PLUS® II development system for 486- and Pentium-based PCs, Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations
- ❖ Flexible package options
 - Available in a variety of packages with 144 to 672 pins, including the innovative FineLine BGA™ packages (see Tables 3 and 4)
 - SameFrame™ pin-compatibility (with other FLEX 10KE devices) across device densities and pin counts
- ❖ Additional design entry and simulation support provided by electronic design interchange format (EDIF) 2.0.0 and 3.0.0 netlist files, library of parameterized modules (LPM), DesignWare components, Verilog HDL, VHDL, and other interfaces to popular electronic design automation (EDA) tools from manufacturers such as Cadence, Exemplar Logic, Mentor Graphics, OrCAD, Synopsys, Synplicity, VeriBest, and Viewlogic

Device	144-Pin TQFP	208-Pin PQFP	240-Pin PQFP	256-Pin FineLine BGA	356-Pin BGA	484-Pin FineLine BGA	599-Pin PGA	600-Pin BGA	672-Pin FineLine BGA
EPF10K30E	102	147		176		220			
EPF10K50E	102	147	189	191		254			
EPF10K100B		147	189	191					
EPF10K100E		147	189	191	274	338			
EPF10K130E			186			369			413
EPF10K200E							470	470	470
EPF10K250E							470	470	470

Notes:

- (1) Contact Altera Customer Marketing for up-to-date information on package availability.
- (2) FLEX 10KE device package types include thin quad flat pack (TQFP), plastic quad flat pack (PQFP), pin-grid array (PGA), and ball-grid array (BGA).
- (3) Devices in the same package are pin-compatible, although some devices have more I/O pins than others. When planning device migration, use the I/O pins that are common to all devices. MAX+PLUS II software versions 9.1 and higher provide features to help use only the common pins.

Device	144-Pin TQFP	208-Pin PQFP	240-Pin PQFP	256-Pin FineLine BGA	356-Pin BGA	484-Pin FineLine BGA	599-Pin PGA	600-Pin BGA	672-Pin FineLine BGA
Pitch (mm)	0.50	0.50	0.50	1.0	1.27	1.0	–	1.27	1.0
Area (mm ²)	484	936	1,197	289	1,225	529	3,904	2,025	729

General Description

Altera FLEX 10KE devices are an enhancement of the FLEX 10K device family. Based on reconfigurable CMOS SRAM elements, the Flexible Logic Element Matrix (FLEX) architecture incorporates all features necessary to implement common gate array megafunctions. With up to 250,000 gates, the FLEX 10KE family provides the density, speed, and features to integrate entire systems, including multiple 32-bit buses, into a single device.

FLEX 10KE devices are configurable, and they are 100% tested prior to shipment. As a result, the designer is not required to generate test vectors for fault coverage purposes. Instead, the designer can focus on simulation and design verification. In addition, the designer does not need to manage inventories of different gate array designs; FLEX 10KE devices can be configured on the board for the specific functionality required.

Table 5 shows FLEX 10KE performance for some common designs. All performance values shown were obtained with Synopsys DesignWare or LPM functions. Special design techniques are not required to implement the applications; the designer simply infers or instantiates a function in a Verilog HDL, VHDL, Altera Hardware Description Language (AHDL), or schematic design file.

Application	Resources Used		Performance			Units
	LEs	EABs	-1 Speed Grade	-2 Speed Grade	-3 Speed Grade	
16-bit loadable counter	16	0	196	182	143	MHz
16-bit accumulator	16	0	196	182	143	MHz
16-to-1 multiplexer, <i>Note (1)</i>	10	0	15	17	18	ns
16-bit multiplier with 3 stage pipeline, <i>Note (2)</i>	10	0	91	86	70	MHz
256 × 16 RAM read cycle speed, <i>Note (2)</i>	0	1	181	142	125	MHz
256 × 16 RAM write cycle speed, <i>Note (2)</i>	0	1	135	117	106	MHz

Notes:

- (1) This application uses combinatorial inputs and outputs.
- (2) This application uses registered inputs and outputs.

Table 6 shows FLEX 10KE performance for more complex designs. These designs are available as Altera MegaCore™ functions.

Application	LEs Used	Performance			Units
		-1 Speed Grade	-2 Speed Grade	-3 Speed Grade	
16-bit, 8-tap parallel finite impulse response (FIR) filter	592	138	128	103	MSPS
8-bit, 512-point fast Fourier transform (FFT) function	1,845	60	72	83	μs
		76	66	56	MHz
a16450 universal asynchronous receiver/transmitter (UART)	479	42	39	34	MHz

The FLEX 10KE architecture is similar to that of embedded gate arrays, which is the fastest-growing segment of the gate array market. As with standard gate arrays, embedded gate arrays implement general logic in a conventional “sea-of-gates” architecture. Additionally, embedded gate arrays have dedicated die areas for implementing large, specialized functions. By embedding functions in silicon, embedded gate arrays provide reduced die area and increased speed compared to standard gate arrays. While embedded megafunctions typically cannot be customized, FLEX 10KE devices are programmable, providing the designer with full control over embedded megafunctions and general logic while facilitating iterative design changes during debugging.

Each FLEX 10KE device contains an embedded array and a logic array. The embedded array is used to implement a variety of memory functions or complex logic functions, such as digital signal processing (DSP), microcontroller, wide-data-path manipulation, and data-transformation functions. The logic array performs the same function as the sea-of-gates in the gate array: it is used to implement general logic, such as counters, adders, state machines, and multiplexers. The combination of embedded and logic arrays provides the high performance and high density of embedded gate arrays, enabling designers to implement an entire system on a single device.

FLEX 10KE devices are configured at system power-up with data stored in an Altera serial Configuration EPROM device or provided by a system controller. Altera offers the EPC1, EPC2, and EPC1441 Configuration EPROMs, which configure FLEX 10KE devices via a serial data stream. Configuration data can also be downloaded from system RAM or from the Altera BitBlaster™ serial download cable, ByteBlaster™ parallel port download cable, or ByteBlasterMV™ parallel port download cable. After a FLEX 10KE device has been configured, it can be reconfigured in-circuit by resetting the device and loading new data. Because reconfiguration requires less than 330 ms, real-time changes can be made during system operation.

FLEX 10KE devices contain an optimized interface that permits microprocessors to configure FLEX 10KE devices serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat a FLEX 10KE device as memory and configure the device by writing to a virtual memory location, making it very easy for the designer to reconfigure the device.



For more information, go to the following documents:

- *Configuration EPROMs for FLEX Devices Data Sheet*
- *BitBlaster Serial Download Cable Data Sheet*
- *ByteBlaster Parallel Port Download Cable Data Sheet*
- *ByteBlasterMV Parallel Port Download Cable Data Sheet*

FLEX 10KE devices are supported by the Altera MAX+PLUS II development system, a single, integrated package that offers schematic, text—including AHDL—and waveform design entry; compilation and logic synthesis; full simulation and worst-case timing analysis; and device configuration. The MAX+PLUS II software provides EDIF 2 0 0 and 3 0 0, LPM, VHDL, Verilog HDL, and other interfaces for additional design entry and simulation support from other industry-standard PC- and UNIX workstation-based EDA tools.

The MAX+PLUS II software works easily with common gate array EDA tools for synthesis and simulation. For example, the MAX+PLUS II software can generate Verilog HDL files for simulation with tools such as Cadence Verilog-XL. Additionally, the MAX+PLUS II software contains EDA libraries that use device-specific features such as carry chains, which are used for fast counter and arithmetic functions. For instance, the Synopsys Design Compiler library supplied with the MAX+PLUS II development system includes DesignWare functions that are optimized for the FLEX 10KE architecture.

The MAX+PLUS II software runs on 486- and Pentium-based PCs, Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations.



Go to the *MAX+PLUS II Programmable Logic Development System & Software Data Sheet* in the **1998 Data Book** for more information.

Functional Description

Each FLEX 10KE device contains an enhanced embedded array to implement memory and specialized logic functions, and a logic array to implement general logic.

The embedded array consists of a series of EABs. When implementing memory functions, each EAB provides 4,096 bits, which can be used to create RAM, ROM, dual-port RAM, or first-in first-out (FIFO) functions. When implementing logic, each EAB can contribute 100 to 600 gates towards complex logic functions, such as multipliers, microcontrollers, state machines, and DSP functions. EABs can be used independently, or multiple EABs can be combined to implement larger functions.

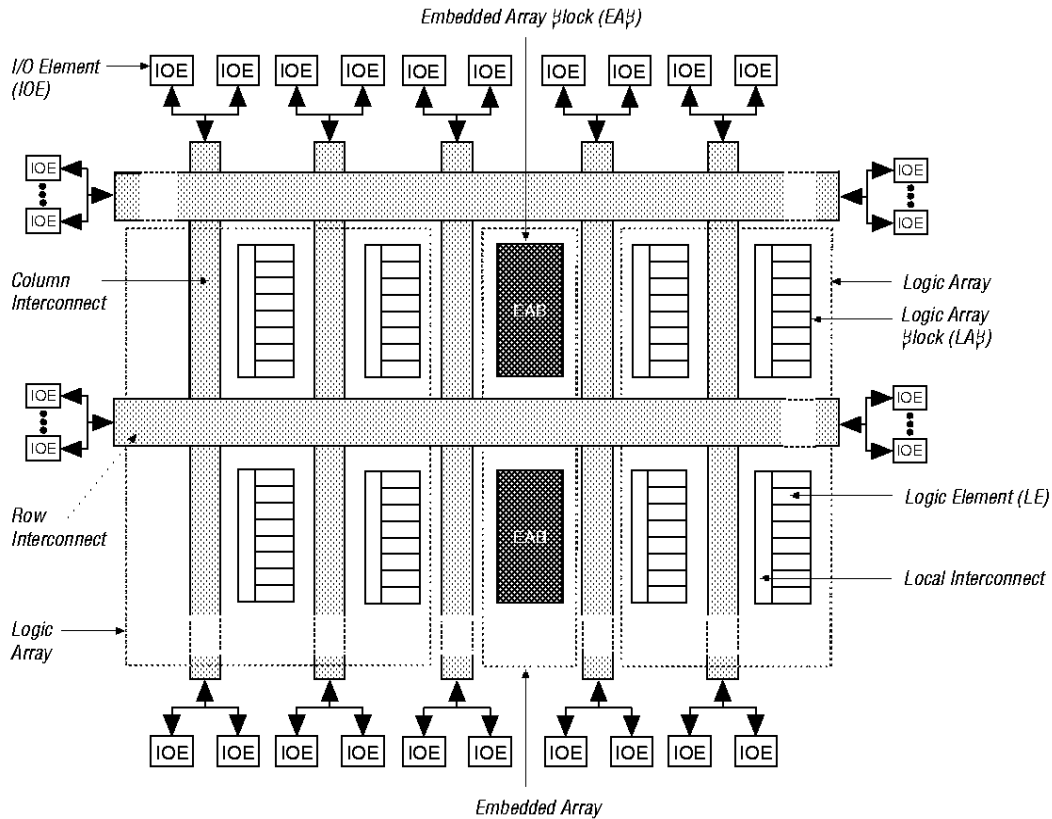
The logic array consists of logic array blocks (LABs). Each LAB contains eight LEs and a local interconnect. An LE consists of a 4-input look-up table (LUT), a programmable flipflop, and dedicated signal paths for carry and cascade functions. The eight LEs can be used to create medium-sized blocks of logic—such as 8-bit counters, address decoders, or state machines—or combined across LABs to create larger logic blocks. Each LAB represents about 96 usable gates of logic.

Signal interconnections within FLEX 10KE devices (as well as to and from device pins) are provided by the FastTrack Interconnect routing structure, a series of fast, continuous row and column channels that run the entire length and width of the device.

Each I/O pin is fed by an I/O element (IOE) located at the end of each row and column of the FastTrack Interconnect routing structure. Each IOE contains a bidirectional I/O buffer and a flipflop that can be used as either an output or input register to feed input, output, or bidirectional signals. When used with a dedicated clock pin, these registers provide exceptional performance. As inputs, they provide setup times as low as 2.9 ns and hold times of 0 ns; as outputs, these registers provide clock-to-output times as low as 4.3 ns. IOEs provide a variety of features, such as JTAG BST support, slew-rate control, tri-state buffers, and open-drain outputs.

Figure 1 shows a block diagram of the FLEX 10KE architecture. Each group of LEs is combined into an LAB; LABs are arranged into rows and columns. Each row also contains a single EAB. The LABs and EABs are interconnected by the FastTrack Interconnect routing structure. IOEs are located at the end of each row and column of the FastTrack Interconnect routing structure.

Figure 1. FLEX 10KE Device Block Diagram



FLEX 10KE devices provide six dedicated inputs that drive the control inputs of the flipflops to ensure the efficient distribution of high-speed, low-skew (less than 1.5 ns) control signals. These signals use dedicated routing channels that provide shorter delays and lower skews than the FastTrack Interconnect routing structure. Four of the dedicated inputs drive four global signals. These four global signals can also be driven by internal logic, providing an ideal solution for a clock divider or an internally generated asynchronous clear signal that clears many registers in the device.

Embedded Array Block

The EAB is a flexible block of RAM with registers on the input and output ports, and is used to implement common gate array megafunctions. Because it is large and flexible, the EAB is suitable for functions such as multipliers, vector scalars, and error correction circuits. These functions can be combined in applications such as digital filters and microcontrollers.

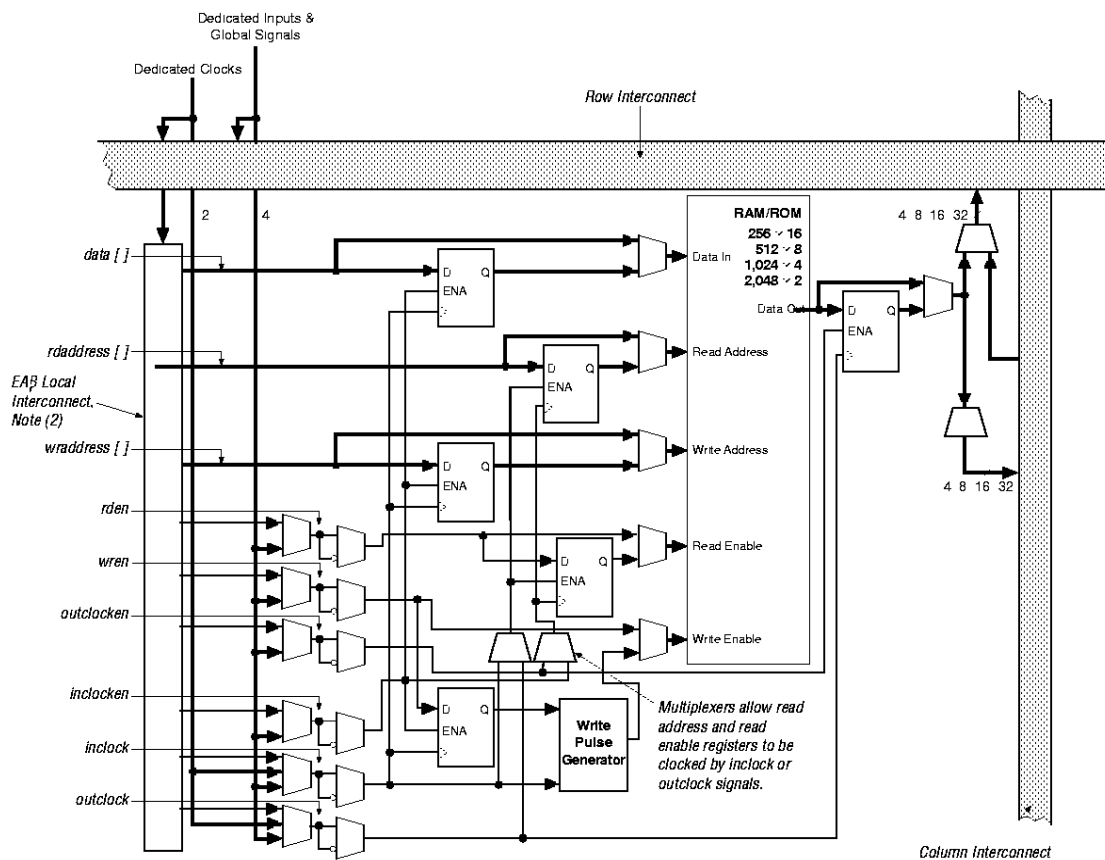
Logic functions are implemented by programming the EAB with a read-only pattern during configuration, thereby creating a large LUT. With LUTs, combinatorial functions are implemented by looking up the results, rather than by computing them. This implementation of combinatorial functions can be faster than using algorithms implemented in general logic, a performance advantage that is further enhanced by the fast access times of EABs. The large capacity of EABs enables designers to implement complex functions in one logic level without the routing delays associated with linked LEs or field-programmable gate array (FPGA) RAM blocks. For example, a single EAB can implement a 5×4 multiplier or any function with nine inputs and nine outputs. Parameterized functions such as LPM functions can take advantage of the EAB automatically.

The FLEX 10KE EAB provides advantages over FPGAs which implement on-board RAM as arrays of small, distributed RAM blocks. These small FPGA RAM blocks must be connected together to make usable-sized RAM blocks. The RAM blocks are connected together using multiplexers implemented with more logic blocks. These extra multiplexers cause extra delay, which slows down the RAM block. FPGA RAM blocks are also prone to routing problems because small blocks of RAM must be connected together to make larger blocks. In contrast, EABs can be used to implement large, dedicated blocks of RAM that eliminate these timing and routing concerns.

The FLEX 10KE enhanced EAB adds dual-port capability to the existing EAB structure. The dual-port structure is ideal for FIFO buffers with one or two clocks. The FLEX 10KE EAB can also support up to 16-bit-wide RAM blocks and is backward-compatible with any design containing FLEX 10K EABs. The FLEX 10KE EAB can act in dual-port or single-port mode. When in dual-port mode, separate clocks may be used for EAB read and write sections, which allows the EAB to be written and read at different rates. It also has separate synchronous clock enable signals for the EAB read and write sections, which allow independent control of these sections.

Alternatively, one clock and clock enable can be used to control the input registers of the EAB, while a different clock and clock enable control the output registers (see Figure 2).

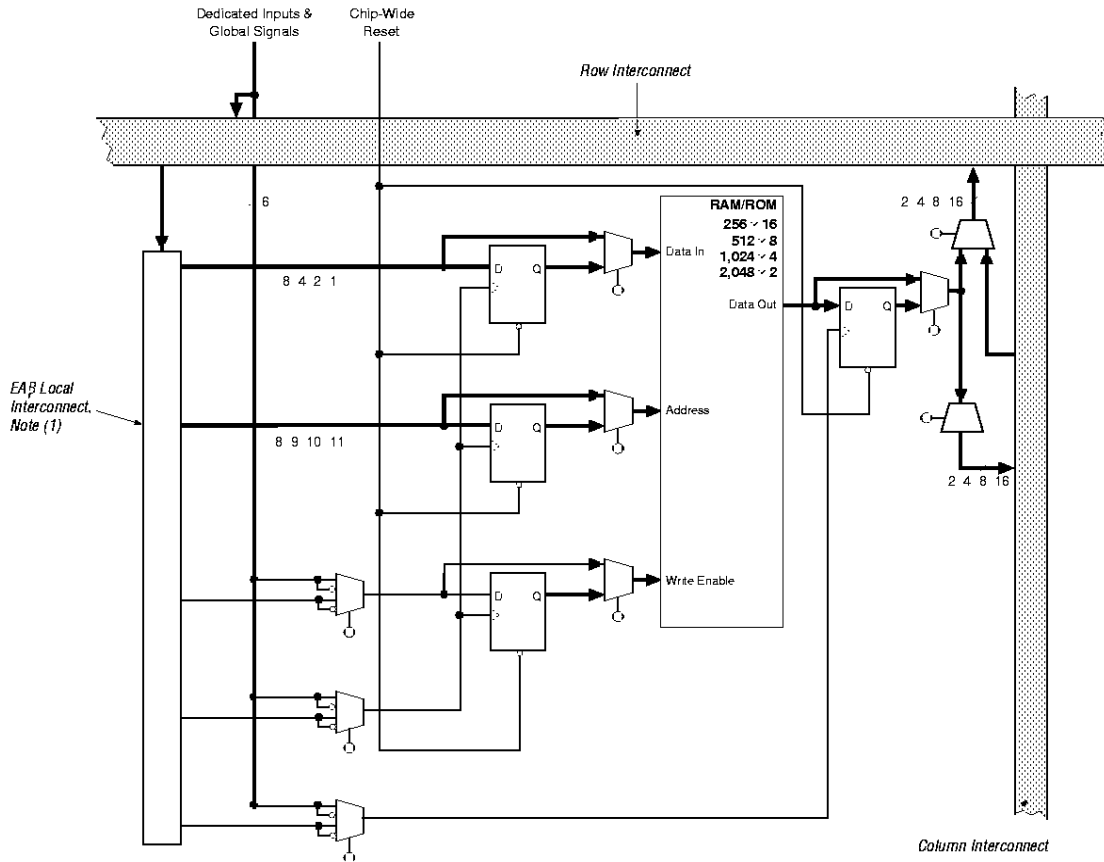
Figure 2. FLEX 10KE Device in Dual-Port RAM Mode Notes (1), (3)



Notes:

- (1) All registers can be asynchronously cleared by EAB local interconnect signals, global signals, or the chip-wide reset.
- (2) EPF10K30E and EPF10K50E devices have 88 EAB local interconnect channels. EPF10K100E, EPF10K130E, EPF10K200E, and EPF10K250E devices have 104 EAB local interconnect channels.
- (3) The EPF10K100B device does not offer dual-port RAM mode.

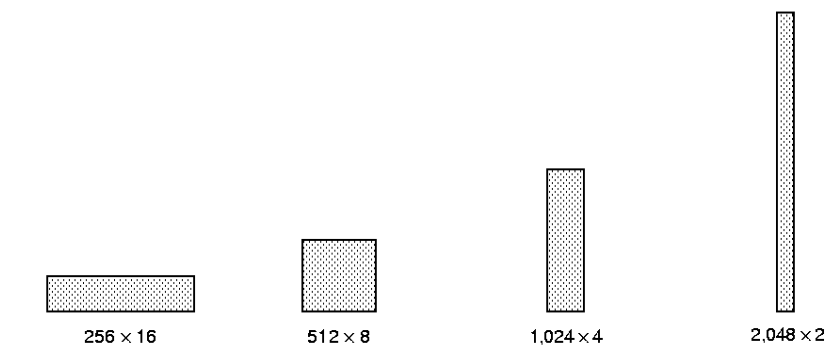
The FLEX 10KE EAB can be used in a single-port mode, which is useful for backward-compatibility with FLEX 10K designs (see Figure 3).

Figure 3. FLEX 10KE Device in Single-Port EAB**Note:**

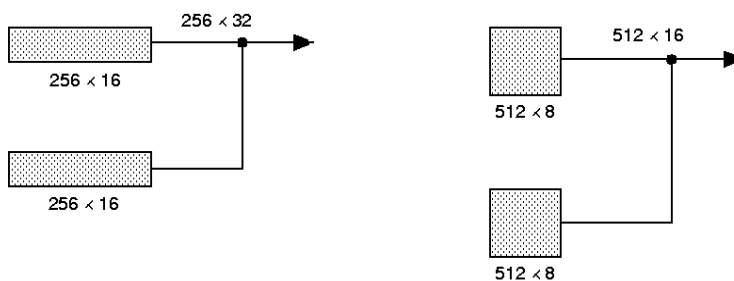
- (1) EPF10K30E and EPF10K50E devices have 88 EAB local interconnect channels. EPF10K100E, EPF10K100B, EPF10K130E, EPF10K200E, and EPF10K250E devices have 104 EAB local interconnect channels.

EABs can be used to 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 EAB's synchronous RAM generates its own WE signal and is self-timed with respect to the input or write clock. A circuit using the EAB's self-timed RAM must only meet the setup and hold time specifications of the global clock.

When used as RAM, each EAB can be configured in any of the following sizes: 256×16 , 512×8 , $1,024 \times 4$, or $2,048 \times 2$ (see Figure 4).

Figure 4. FLEX 10KE EAB Memory Configurations

Larger blocks of RAM are created by combining multiple EABs. For example, two 256×16 RAM blocks can be combined to form a 256×32 block; two 512×8 RAM blocks can be combined to form a 512×16 block (see Figure 5).

Figure 5. Examples of Combining FLEX 10KE EABs

If necessary, all EABs in a device can be cascaded to form a single RAM block. EABs can be cascaded to form RAM blocks of up to 2,048 words without impacting timing. The Altera MAX+PLUS II software automatically combines EABs to meet a designer's RAM specifications.

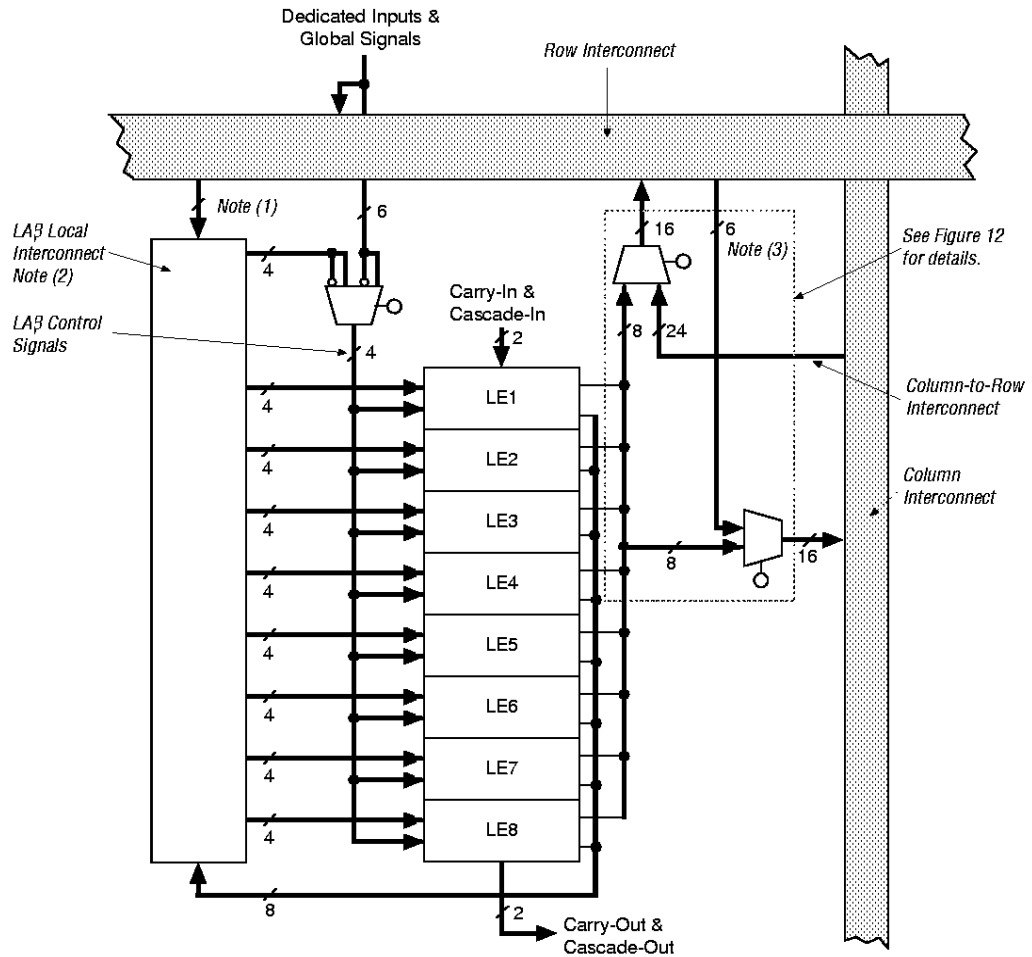
EABs provide flexible options for driving and controlling clock signals. Different clocks and clock enables can be used for reading and writing to the EAB. Registers can be independently inserted on the data input, EAB output, write address, WE signals, read address, and RE signals. The global signals and the EAB local interconnect can drive WE , RE , and clock enable signals. The global signals, dedicated clock pins, and EAB local interconnect can drive the EAB clock signals. Because the LEs drive the EAB local interconnect, the LEs can control WE , RE , clear, clock, and clock enable signals.

An EAB is fed by a row interconnect and can drive out to row and column interconnects. Each EAB output can drive up to two row channels and up to two column channels; the unused row channel can be driven by other LEs. This feature increases the routing resources available for EAB outputs (see Figures 2 and 3). The column interconnect, which is adjacent to the EAB, has twice as many channels as other columns in the device.

Logic Array Block

A LAB consists of eight LEs, their associated carry and cascade chains, LAB control signals, and the LAB local interconnect. The LAB provides the coarse-grained structure to the FLEX 10KE architecture, facilitating efficient routing with optimum device utilization and high performance (see Figure 6).

Figure 6. FLEX 10KE LAB



Notes:

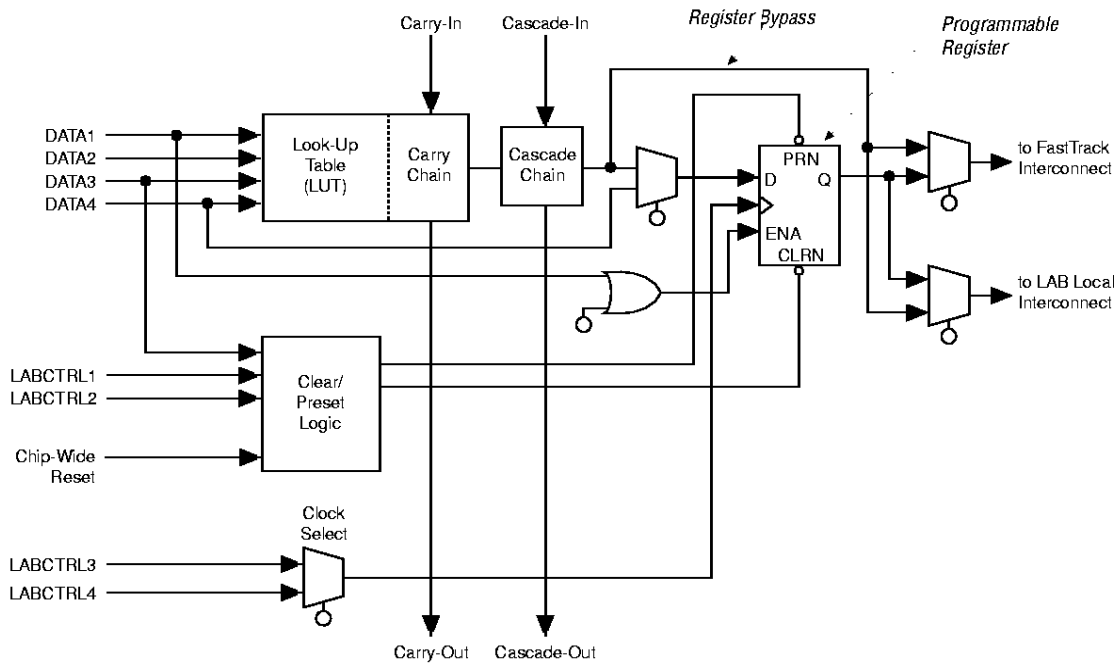
- (1) EPF10K30E and EPF10K50E devices have 22 inputs to the LAB local interconnect channel from the row; EPF10K100E, EPF10K100B, EPF10K130E, EPF10K200E, and EPF10K250E devices have 26.
- (2) EPF10K30E and EPF10K50E devices have 30 LAB local interconnect channels; EPF10K100E, EPF10K100B, EPF10K130E, EPF10K200E, and EPF10K250E devices have 34.
- (3) In EPF10K100B devices, four row channels can drive column channels at each intersection.

Each LAB provides four control signals with programmable inversion that can be used in all eight LEs. Two of these signals can be used as clocks; the other two can be used for clear/preset control. The LAB clocks can be driven by the dedicated clock input pins, global signals, I/O signals, or internal signals via the LAB local interconnect. The LAB preset and clear control signals can be driven by the global signals, I/O signals, or internal signals via the LAB local interconnect. The global control signals are typically used for global clock, clear, or preset signals because they provide asynchronous control with very low skew across the device. If logic is required on a control signal, it can be generated in one or more LEs in any LAB and driven into the local interconnect of the target LAB. In addition, the global control signals can be generated from LE outputs.

Logic Element

The LE, the smallest unit of logic in the FLEX 10KE architecture, has a compact size that provides efficient logic utilization. Each LE contains a four-input LUT, which is a function generator that can quickly compute any function of four variables. In addition, each LE contains a programmable flipflop with a synchronous clock enable, a carry chain, and a cascade chain. Each LE drives both the local and the FastTrack Interconnect routing structure (see Figure 7).

Figure 7. FLEX 10KE Logic Element



The programmable flipflop in the LE can be configured for D, T, JK, or SR operation. The clock, clear, and preset control signals on the flipflop can be driven by global signals, general-purpose I/O pins, or any internal logic. For combinatorial functions, the flipflop is bypassed and the output of the LUT drives the output of the LE.

The LE has two outputs that drive the interconnect; one drives the local interconnect and the other drives either the row or column FastTrack Interconnect routing structure. The two outputs can be controlled independently; for example, the LUT can drive one output while the register drives the other output. This feature, called register packing, can improve LE utilization because the register and the LUT can be used for unrelated functions.

The FLEX 10KE architecture provides two types of dedicated high-speed data paths that connect adjacent LEs without using local interconnect paths: carry chains and cascade chains. The carry chain supports high-speed counters and adders; the cascade chain implements wide-input functions with minimum delay. Carry and cascade chains connect all LEs in a LAB and all LABs in the same row. Intensive use of carry and cascade chains can reduce routing flexibility. Therefore, the use of these chains should be limited to speed-critical portions of a design.

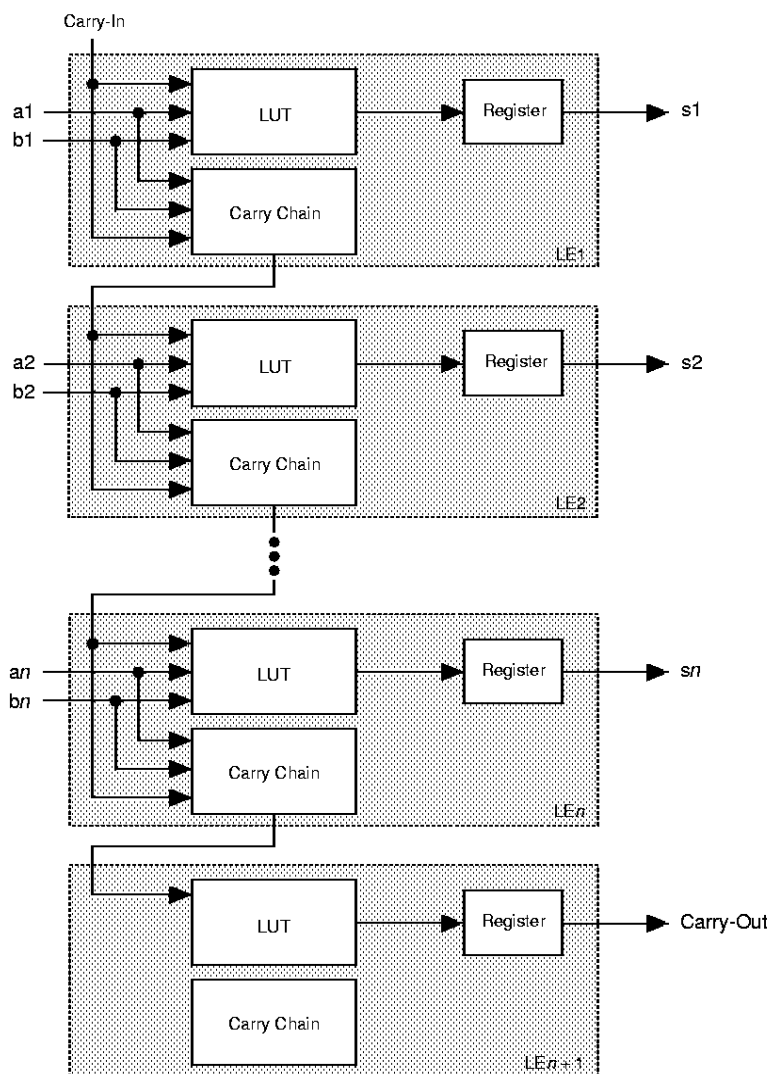
Carry Chain

The carry chain provides a very fast (as low as 0.2 ns) carry-forward function between LEs. The carry-in signal from a lower-order bit drives forward into the higher-order bit via the carry chain, and feeds into both the LUT and the next portion of the carry chain. This feature allows the FLEX 10KE architecture to implement high-speed counters, adders, and comparators of arbitrary width efficiently. Carry chain logic can be created automatically by the MAX+PLUS II Compiler during design processing, or manually by the designer during design entry. Parameterized functions such as LPM and DesignWare functions automatically take advantage of carry chains.

Carry chains longer than eight LEs are automatically implemented by linking LABs together. For enhanced fitting, a long carry chain skips alternate LABs in a row. A carry chain longer than one LAB skips either from even-numbered LAB to even-numbered LAB, or from odd-numbered LAB to odd-numbered LAB. For example, the last LE of the first LAB in a row carries to the first LE of the third LAB in the row. The carry chain does not cross the EAB at the middle of the row. For instance, in the EPF10K50E device, the carry chain stops at the eighteenth LAB and a new one begins at the nineteenth LAB.

Figure 8 shows how an n -bit full adder can be implemented in $n + 1$ LEs with the carry chain. One portion of the LUT generates the sum of two bits using the input signals and the carry-in signal; the sum is routed to the output of the LE. The register can be bypassed for simple adders or used for an accumulator function. Another portion of the LUT and the carry chain logic generates the carry-out signal, which is routed directly to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it can be used as a general-purpose signal.

Figure 8. FLEX 10KE Carry Chain Operation (N-Bit Full Adder)

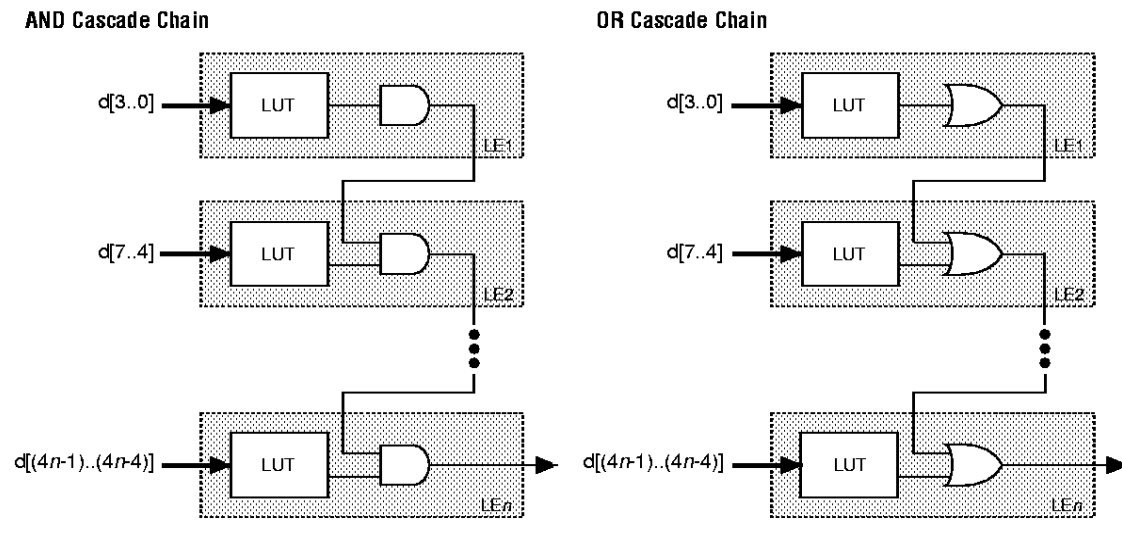


Cascade Chain

With the cascade chain, the FLEX 10KE architecture can implement functions that have a very wide fan-in. Adjacent LUTs can be used to compute portions of the function in parallel; the cascade chain serially connects the intermediate values. The cascade chain can use a logical AND or logical OR (via De Morgan's inversion) to connect the outputs of adjacent LEs. With a delay as low as 0.6 ns per LE, each additional LE provides four more inputs to the effective width of a function. Cascade chain logic can be created automatically by the MAX+PLUS II Compiler during design processing, or manually by the designer during design entry.

Cascade chains longer than eight bits are automatically implemented by linking several LABs together. For easier routing, a long cascade chain skips every other LAB in a row. A cascade chain longer than one LAB skips either from even-numbered LAB to even-numbered LAB, or from odd-numbered LAB to odd-numbered LAB (e.g., the last LE of the first LAB in a row cascades to the first LE of the third LAB). The cascade chain does not cross the center of the row (e.g., in the EPF10K50E device, the cascade chain stops at the eighteenth LAB and a new one begins at the nineteenth LAB). This break is due to the EAB's placement in the middle of the row.

Figure 9 shows how the cascade function can connect adjacent LEs to form functions with a wide fan-in. These examples show functions of $4n$ variables implemented with n LEs. The LE delay is 1.3 ns; the cascade chain delay is 0.6 ns. With the cascade chain, approximately 3.1 ns are needed to decode a 16-bit address.

Figure 9. FLEX 10KE Cascade Chain Operation

LE Operating Modes

The FLEX 10KE LE can operate in the following four modes:

- ⊠ Normal mode
- ⊠ Arithmetic mode
- ⊠ Up/down counter mode
- ⊠ Clearable counter mode

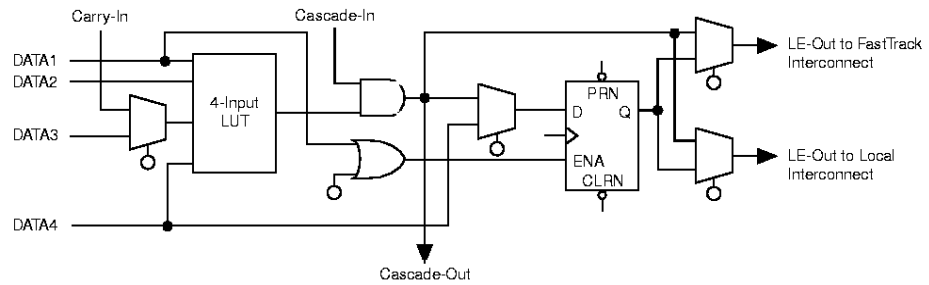
Each of these modes uses LE resources differently. In each mode, seven available inputs to the LE—the four data inputs from the LAB local interconnect, the feedback from the programmable register, and the carry-in and cascade-in from the previous LE—are directed to different destinations to implement the desired logic function. Three inputs to the LE provide clock, clear, and preset control for the register. The MAX+PLUS II software, in conjunction with parameterized functions such as LPM and DesignWare functions, automatically chooses the appropriate mode for common functions such as counters, adders, and multipliers. If required, the designer can also create special-purpose functions to use an LE operating mode for optimal performance.

The architecture provides a synchronous clock enable to the register in all four modes. The MAX+PLUS II software can set DATA1 to enable the register synchronously, providing easy implementation of fully synchronous designs.

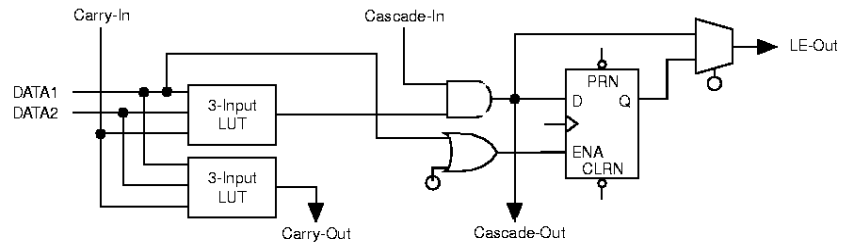
Figure 10 shows the LE operating modes.

Figure 10. FLEX 10KE LE Operating Modes

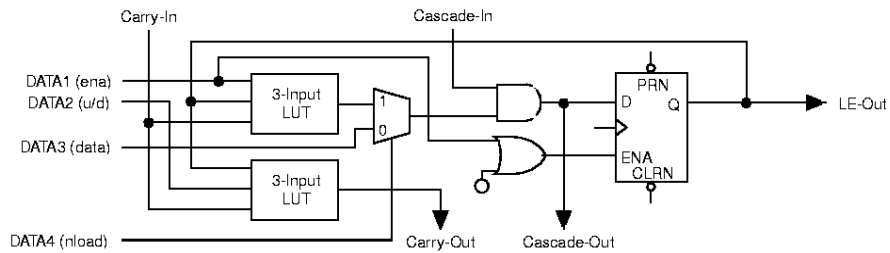
Normal Mode



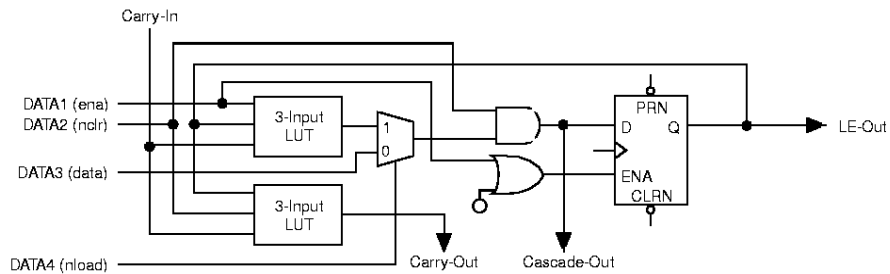
Arithmetic Mode



Up/Down Counter Mode



Clearable Counter Mode



Normal Mode

The normal mode is suitable for general logic applications and wide decoding functions that can take advantage of a cascade chain. In normal mode, four data inputs from the LAB local interconnect and the carry-in are inputs to a 4-input LUT. The MAX+PLUS II Compiler automatically selects the carry-in or the DATA3 signal as one of the inputs to the LUT. The LUT output can be combined with the cascade-in signal to form a cascade chain through the cascade-out signal. Either the register or the LUT can be used to drive both the local interconnect and the FastTrack Interconnect routing structure at the same time.

The LUT and the register in the LE can be used independently; this feature is known as register packing. To support register packing, the LE has two outputs; one drives the local interconnect, and the other drives the FastTrack Interconnect routing structure. The DATA4 signal can drive the register directly, allowing the LUT to compute a function that is independent of the registered signal; a 3-input function can be computed in the LUT, and a fourth independent signal can be registered. Alternatively, a 4-input function can be generated, and one of the inputs to this function can be used to drive the register. The register in a packed LE can still use the clock enable, clear, and preset signals in the LE. In a packed LE, the register can drive the FastTrack Interconnect routing structure while the LUT drives the local interconnect, or vice versa.

Arithmetic Mode

The arithmetic mode offers two 3-input LUTs that are ideal for implementing adders, accumulators, and comparators. One LUT computes a 3-input function; the other generates a carry output. As shown in Figure 10 on page 20, the first LUT uses the carry-in signal and two data inputs from the LAB local interconnect to generate a combinatorial or registered output. For example, in an adder, this output is the sum of three signals: a, b, and carry-in. The second LUT uses the same three signals to generate a carry-out signal, thereby creating a carry chain. The arithmetic mode also supports simultaneous use of the cascade chain.

Up/Down Counter Mode

The up/down counter mode offers counter enable, clock enable, synchronous up/down control, and data loading options. These control signals are generated by the data inputs from the LAB local interconnect, the carry-in signal, and output feedback from the programmable register. Two 3-input LUTs are used: one generates the counter data, and the other generates the fast carry bit. A 2-to-1 multiplexer provides synchronous loading. Data can also be loaded asynchronously with the clear and preset register control signals, without using the LUT resources.

Clearable Counter Mode

The clearable counter mode is similar to the up/down counter mode, but supports a synchronous clear instead of the up/down control. The clear function is substituted for the cascade-in signal in the up/down counter mode. Two 3-input LUTs are used: one generates the counter data, and the other generates the fast carry bit. Synchronous loading is provided by a 2-to-1 multiplexer. The output of this multiplexer is ANDed with a synchronous clear signal.

Internal Tri-State Emulation

Internal tri-state emulation provides internal tri-stating without the limitations of a physical tri-state bus. In a physical tri-state bus, the tri-state buffers' output enable (OE) signals select which signal drives the bus. However, if multiple OE signals are active, contending signals can be driven onto the bus. Conversely, if no OE signals are active, the bus will float. Internal tri-state emulation resolves contending tri-state buffers to a low value and floating buses to a high value, thereby eliminating these problems. The MAX+PLUS II software automatically implements tri-state bus functionality with a multiplexer.

Clear & Preset Logic Control

Logic for the programmable register's clear and preset functions is controlled by the DATA3, LABCTRL1, and LABCTRL2 inputs to the LE. The clear and preset control structure of the LE asynchronously loads signals into a register. Either LABCTRL1 or LABCTRL2 can control the asynchronous clear. Alternatively, the register can be set up so that LABCTRL1 implements an asynchronous load. The data to be loaded is driven to DATA3; when LABCTRL1 is asserted, DATA3 is loaded into the register.

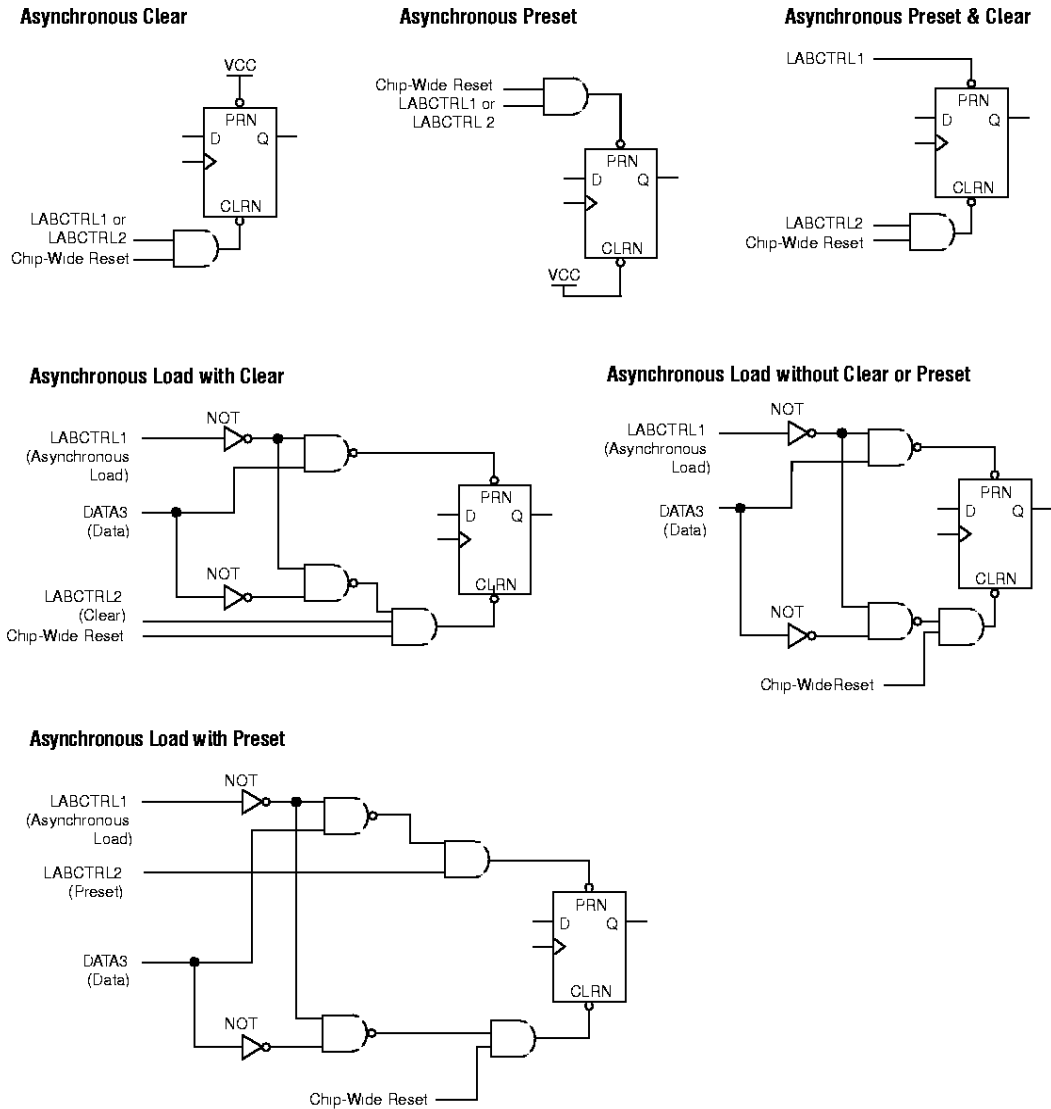
During compilation, the MAX+PLUS II Compiler automatically selects the best control signal implementation. Because the clear and preset functions are active-low, the Compiler automatically assigns a logic high to an unused clear or preset.

The clear and preset logic is implemented in one of the following six modes chosen during design entry:

- ⊠ Asynchronous clear
- ⊠ Asynchronous preset
- ⊠ Asynchronous clear and preset
- ⊠ Asynchronous load with clear
- ⊠ Asynchronous load with preset
- ⊠ Asynchronous load without clear or preset

In addition to the six clear and preset modes, FLEX 10KE devices provide a chip-wide reset pin that can reset all registers in the device. Use of this feature is set during design entry. In any of the clear and preset modes, the chip-wide reset overrides all other signals. Registers with asynchronous presets may be preset when the chip-wide reset is asserted. Inversion can be used to implement the asynchronous preset. Figure 11 shows examples of how to enter a design for the desired functionality.

Figure 11. FLEX 10KE LE Clear & Preset Modes



Asynchronous Clear

The flipflop can be cleared by either LABCTRL1 or LABCTRL2. In this mode, the preset signal is tied to VCC to deactivate it.

Asynchronous Preset

An asynchronous preset is implemented as an asynchronous load, or with an asynchronous clear. If DATA3 is tied to VCC, asserting LABCTRL1 asynchronously loads a one into the register. Alternatively, the MAX+PLUS II software can provide preset control by using the clear and inverting the input and output of the register. Inversion control is available for the inputs to both LEs and IOEs. Therefore, if a register is preset by only one of the two LABCTRL signals, the DATA3 input is not needed and can be used for one of the LE operating modes.

Asynchronous Preset & Clear

When implementing asynchronous clear and preset, LABCTRL1 controls the preset and LABCTRL2 controls the clear. DATA3 is tied to VCC, so that asserting LABCTRL1 asynchronously loads a one into the register, effectively presetting the register. Asserting LABCTRL2 clears the register.

Asynchronous Load with Clear

When implementing an asynchronous load in conjunction with the clear, LABCTRL1 implements the asynchronous load of DATA3 by controlling the register preset and clear. LABCTRL2 implements the clear by controlling the register clear; LABCTRL2 does not have to feed the preset circuits.

Asynchronous Load with Preset

When implementing an asynchronous load in conjunction with preset, the MAX+PLUS II software provides preset control by using the clear and inverting the input and output of the register. Asserting LABCTRL2 presets the register, while asserting LABCTRL1 loads the register. The MAX+PLUS II software inverts the signal that drives DATA3 to account for the inversion of the register's output.

Asynchronous Load without Preset or Clear

When implementing an asynchronous load without preset or clear, LABCTRL1 implements the asynchronous load of DATA3 by controlling the register preset and clear.

FastTrack Interconnect Routing Structure

In the FLEX 10KE architecture, connections between LEs, EABs, and device I/O pins are provided by the FastTrack Interconnect routing structure, which is a series of continuous horizontal and vertical routing channels that traverse the device. This global routing structure provides predictable performance, even in complex designs. In contrast, the segmented routing in FPGAs requires switch matrices to connect a variable number of routing paths, increasing the delays between logic resources and reducing performance.

The FastTrack Interconnect routing structure consists of row and column interconnect channels that span the entire device. Each row of LABs is served by a dedicated row interconnect. The row interconnect can drive I/O pins and feed other LABs in the row. The column interconnect routes signals between rows and can drive I/O pins.

Row channels drive into the LAB or EAB local interconnect. The row signal is buffered at every LAB or EAB to reduce the effect of fan-out on delay. A row channel can be driven by an LE or by one of three column channels. These four signals feed dual 4-to-1 multiplexers that connect to two specific row channels. These multiplexers, which are connected to each LE, allow column channels to drive row channels even when all eight LEs in a LAB drive the row interconnect.

Each column of LABs or EABs is served by a dedicated column interconnect. The column interconnect that serves the EABs has twice as many channels as other column interconnects. The column interconnect can then drive I/O pins or another row's interconnect to route the signals to other LABs or EABs in the device. A signal from the column interconnect, which can be either the output of a LE or an input from an I/O pin, must be routed to the row interconnect before it can enter a LAB or EAB. Each row channel that is driven by an IOE or EAB can drive one specific column channel.

Access to row and column channels can be switched between LEs in adjacent pairs of LABs. For example, a LE in one LAB can drive the row and column channels normally driven by a particular LE in the adjacent LAB in the same row, and vice versa. This routing flexibility enables routing resources to be used more efficiently (see Figure 12).