



Welcome to [E-XFL.COM](https://www.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	-
Number of Logic Elements/Cells	-
Total RAM Bits	-
Number of I/O	-
Number of Gates	-
Voltage - Supply	-
Mounting Type	Through Hole
Operating Temperature	-
Package / Case	599-BCPGA
Supplier Device Package	599-PGA (62.5x62.5)
Purchase URL	https://www.e-xfl.com/product-detail/intel/epf10k250egc599-3

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.

Table 5. FLEX 10KE Performance						
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.

Table 6. FLEX 10KE Performance for Complex Designs					
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.

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

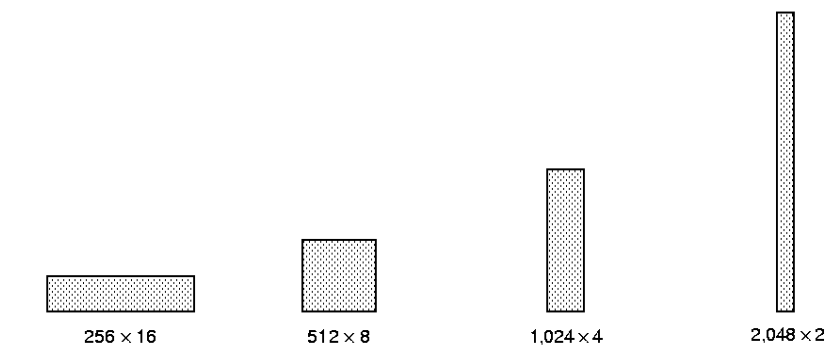
Each FLEX 10KE device contains an enhanced embedded array to implement memory and specialized logic functions, and a logic array to implement general logic.

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.

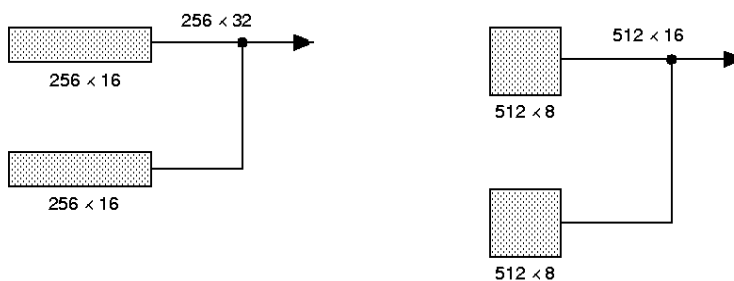
The logic array consists of logic array blocks (LABs). Each LAB contains eight LEs and a local interconnect. An LE consists of a 4-input look-up table (LUT), a programmable flipflop, and dedicated signal paths for carry and cascade functions. The eight LEs can be used to create medium-sized blocks of 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 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.

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.

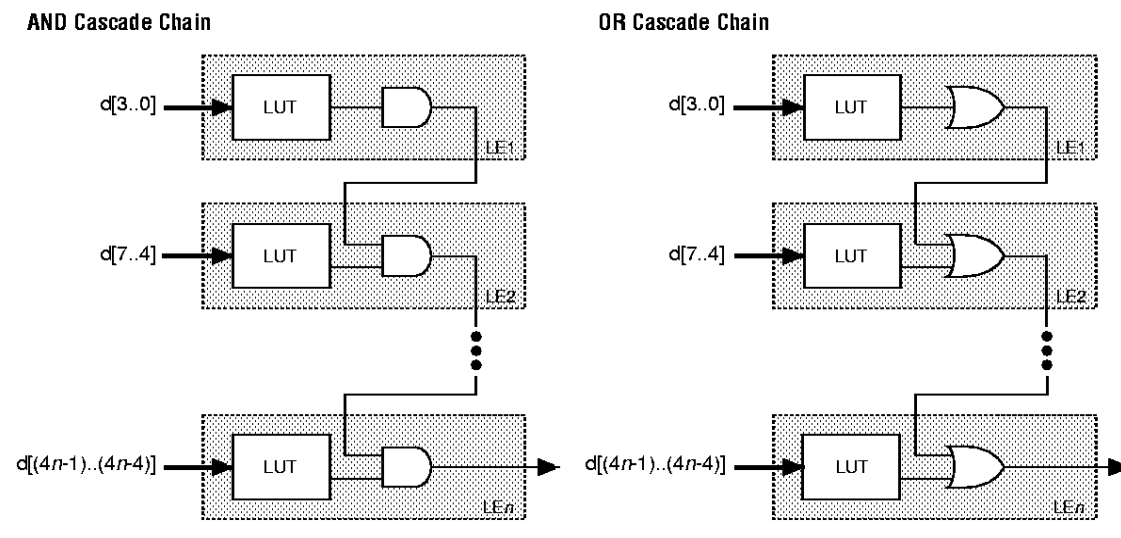
The LE has two outputs that drive the interconnect; one drives the local interconnect and the other drives either the row or column FastTrack Interconnect routing structure. The two outputs can be controlled independently; for example, the LUT can drive one output while the register drives the other 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 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.

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.

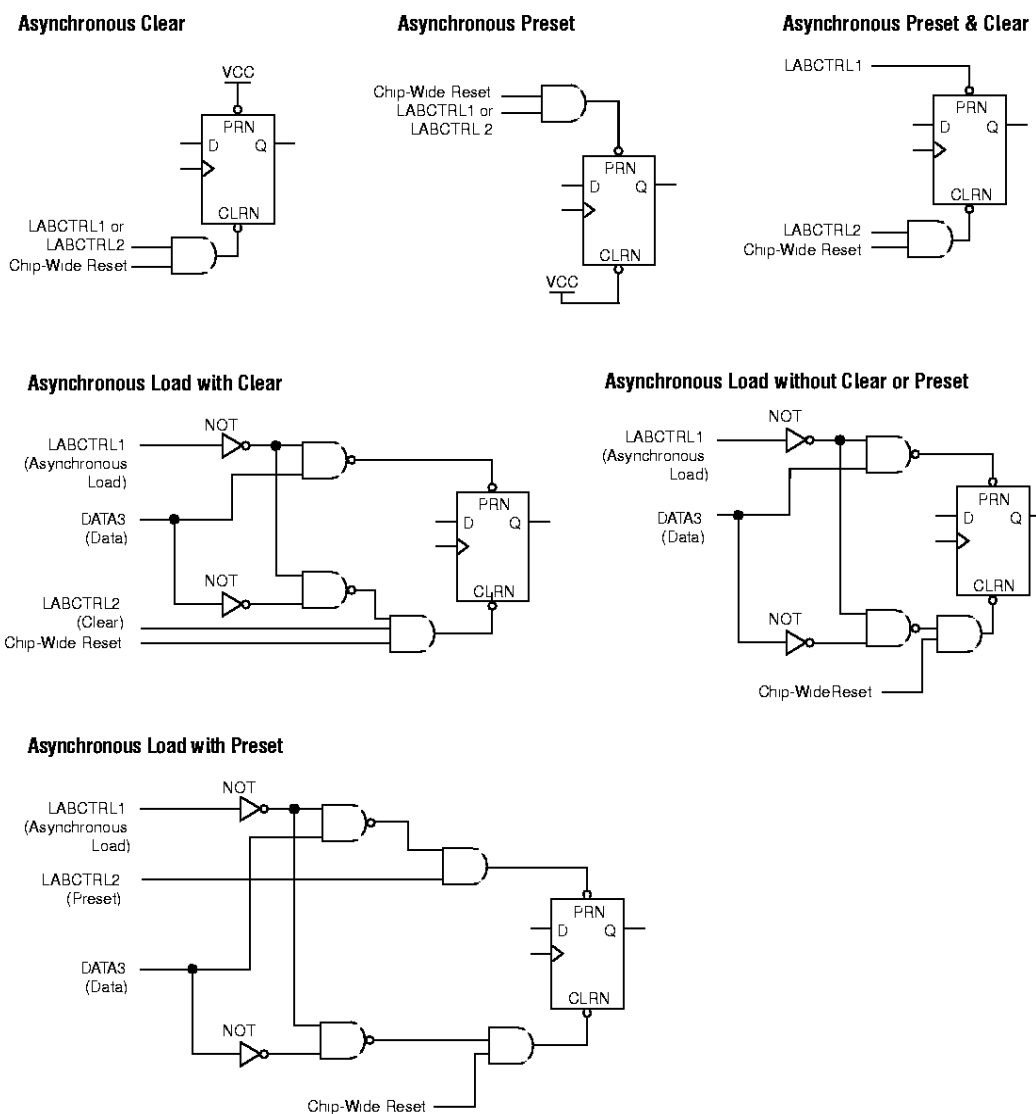
During compilation, the MAX+PLUS II Compiler automatically selects the best control signal implementation. Because the clear and preset functions are active-low, the Compiler automatically assigns a logic high to an unused clear or preset.

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

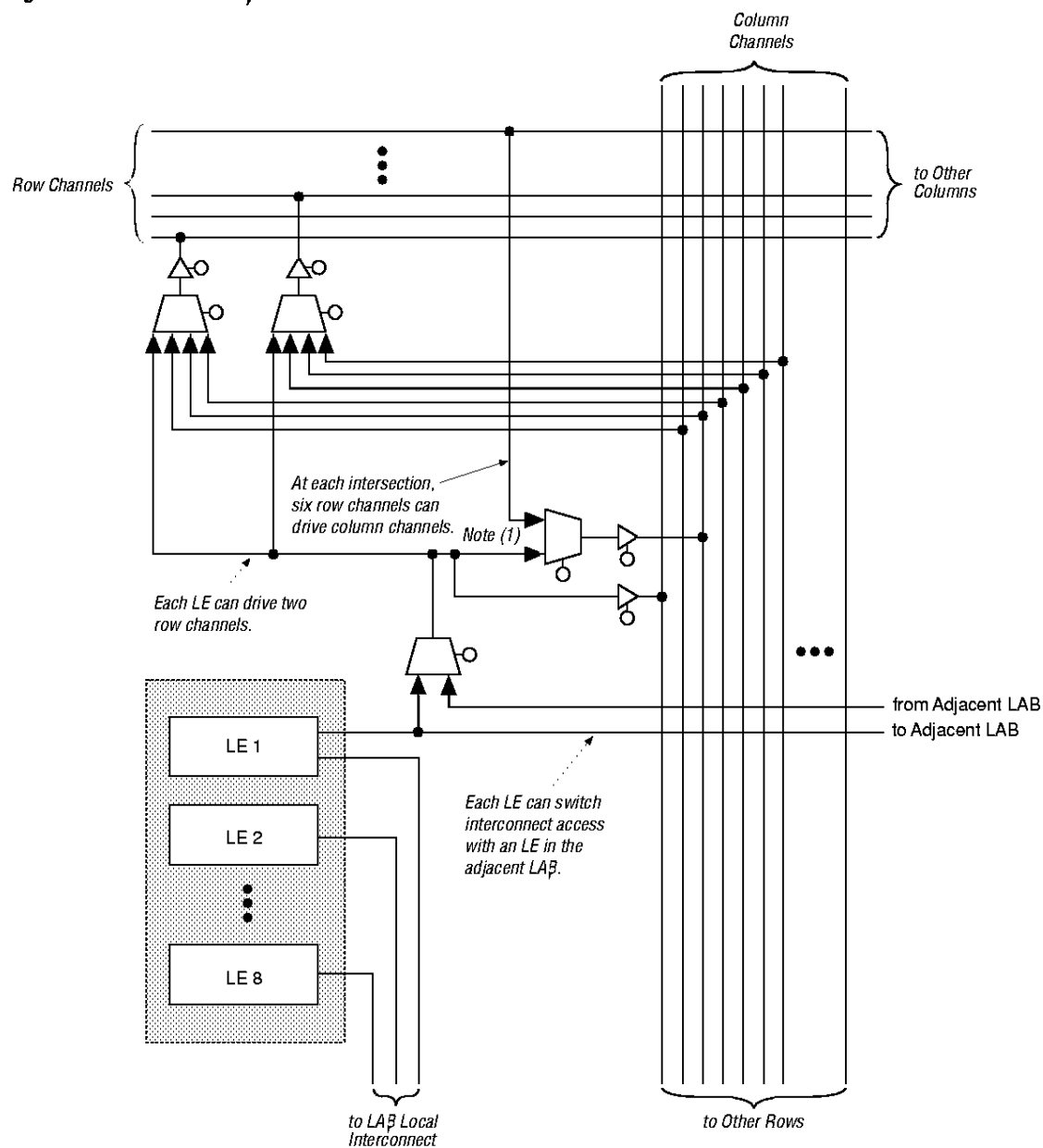
When implementing an asynchronous load in conjunction with the clear, LABCTRL1 implements the asynchronous load of DATA3 by controlling the register preset and clear. LABCTRL2 implements the clear by controlling the register clear; LABCTRL2 does not have to feed the preset circuits.

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.

Figure 12. FLEX 10KE LAB Connections to Row & Column Interconnect**Note:**

(1) In EPF10K100B devices, four row channels can drive column channels at each intersection.

For improved routing, the row interconnect consists of a combination of full-length and half-length channels. The full-length channels connect to all LABs in a row; the half-length channels connect to the LABs in half of the row. The EAB can be driven by the half-length channels in the left half of the row and by the full-length channels. The EAB drives out to the full-length channels. In addition to providing a predictable, row-wide interconnect, this architecture provides increased routing resources. Two neighboring LABs can be connected using a half-row channel, thereby saving the other half of the channel for the other half of the row.

Table 7 summarizes the FastTrack Interconnect routing structure resources available in each FLEX 10KE device.

Table 7. FLEX 10KE FastTrack Interconnect Resources				
Device	Rows	Channels per Row	Columns	Channels per Column
EPF10K30E	6	216	36	24
EPF10K50E	10	216	36	24
EPF10K100B EPF10K100E	12	312	52	24
EPF10K130E	16	312	52	32
EPF10K200E	24	312	52	48
EPF10K250E	20	456	76	40

In addition to general-purpose I/O pins, FLEX 10KE devices have six dedicated input pins that provide low-skew signal distribution across the device. These six inputs can be used for global clock, clear, preset, and peripheral output enable and clock enable control signals. These signals are available as control signals for all LABs and IOEs in the device. The dedicated inputs can also be used as general-purpose data inputs because they can feed the local interconnect of each LAB in the device.

Figure 13 shows the interconnection of adjacent LABs and EABs, with row, column, and local interconnects, as well as the associated cascade and carry chains. Each LAB is labeled according to its location: a letter represents the row and a number represents the column. For example, LAB B3 is in row B, column 3.

- Up to eight output enable signals
- Up to six clock enable signals
- Up to two clock signals
- Up to two clear signals

If more than six clock enable or eight output enable signals are required, each IOE on the device can be controlled by clock enable and output enable signals driven by specific LEs. In addition to the two clock signals available on the peripheral control bus, each IOE can use one of two dedicated clock pins. Each peripheral control signal can be driven by any of the dedicated input pins or the first LE of each LAB in a particular row. In addition, a LE in a different row can drive a column interconnect, which causes a row interconnect to drive the peripheral control signal. The chip-wide reset signal resets all IOE registers, overriding any other control signals.

When a dedicated clock pin drives IOE registers, it can be inverted for all IOEs in the device. All IOEs must use the same sense of the clock. For example, if any IOE uses the inverted clock, all IOEs must use the inverted clock and no IOE can use the non-inverted clock. However, LEs can still use the true or complement of the clock on a LAB-by-LAB basis.

The incoming signal may be inverted at the dedicated clock pin and will drive all IOEs. To use the true and complement of a clock to drive IOEs, drive it into both global clock pins. One global clock pin will supply the true, and the other will supply the complement.

When the true and complement of a dedicated input drives IOE clocks, then two signals on the peripheral control bus are consumed, one for each sense of the clock.

When dedicated inputs drive non-inverted and inverted peripheral clears, clock enables, and output enables, two signals on the peripheral control bus will be used.

Row-to-IOE Connections

When an IOE is used as an input signal, it can drive two separate row channels. The signal is accessible by all LEs within that row. When an IOE is used as an output, the signal is driven by a multiplexer that selects a signal from the row channels. Up to eight IOEs connect to each side of each row channel (see Figure 15).

Figure 15. FLEX 10KE Row-to-IOE Connections

The values for m and n are provided in Table 10.

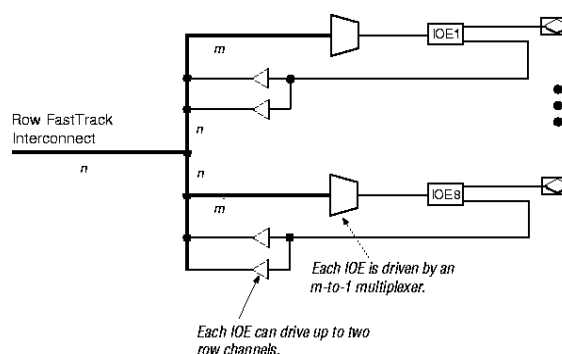


Table 10 lists the FLEX 10KE row-to-IOE interconnect resources.

Table 10. FLEX 10KE Row-to-IOE Interconnect Resources		
Device	Channels per Row (n)	Row Channels per Pin (m)
EPF10K30E	216	27
EPF10K50E	216	27
EPF10K100B EPF10K100E	312	39
EPF10K130E	312	39
EPF10K200E	312	39
EPF10K250E	456	57

Slew-Rate Control

The output buffer in each IOE has an adjustable output slew rate that can be configured for low-noise or high-speed performance. A slower slew rate reduces system noise and adds a maximum delay of approximately 4.3 ns. The fast slew rate should be used for speed-critical outputs in systems that are adequately protected against noise. Designers can specify the slew rate pin-by-pin or assign a default slew rate to all pins on a device-wide basis. The slow slew rate setting affects both the falling and rising edges of the output.

Open-Drain Output Option

FLEX 10KE devices provide an optional open-drain output (electrically equivalent to open-collector output) output for each I/O pin. This open-drain output enables the device to provide system-level control signals (e.g., interrupt and write enable signals) that can be asserted by any of several devices. It can also provide an additional wired-OR plane.

MultiVolt I/O Interface

The FLEX 10KE device architecture supports the MultiVolt I/O interface feature, which allows FLEX 10KE devices in all packages to interface with systems of differing supply voltages. These devices have one set of V_{CC} pins for internal operation and input buffers (V_{CCINT}), and another set for I/O output drivers (V_{CCIO}).

The V_{CCINT} pins must always be connected to a 2.5-V power supply. With a 2.5-V V_{CCINT} level, input voltages are compatible with 2.5-V, 3.3-V, and 5.0-V inputs. The V_{CCIO} pins can be connected to either a 2.5-V or 3.3-V power supply, depending on the output requirements. When the V_{CCIO} pins are connected to a 2.5-V power supply, the output levels are compatible with 2.5-V systems. When the V_{CCIO} pins are connected to a 3.3-V power supply, the output high is at 3.3 V and is therefore compatible with 3.3-V or 5.0-V systems. Devices operating with V_{CCIO} levels higher than 3.0-V achieve a faster timing delay of t_{OD2} instead of t_{OD1} .

Capacitance of FLEX 10KE Devices *Note (12)*

Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Input capacitance	$V_{IN} = 0\text{ V}$, $f = 1.0\text{ MHz}$		10	pF
C_{INCLK}	Input capacitance on dedicated clock pin	$V_{IN} = 0\text{ V}$, $f = 1.0\text{ MHz}$		12	pF

Notes to tables:

- (1) See the *Operating Requirements for Altera Devices Data Sheet* in the **1998 Data Book**.
- (2) Minimum DC input is -0.3 V . During transitions, the inputs may undershoot to -2.0 V or overshoot to 5.7 V for periods shorter than 20 ns under no-load conditions.
- (3) Numbers in parentheses are for industrial-temperature-range devices.
- (4) Maximum V_{CC} rise time is 100 ms , and V_{CC} must rise monotonically.
- (5) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT} and V_{CCIO} are powered.
- (6) Typical values are for $T_A = 25^\circ\text{C}$, $V_{CCINT} = 2.5\text{ V}$, and $V_{CCIO} = 2.5\text{ V}$ or 3.3 V .
- (7) These values are specified under "FLEX 10KE 2.5-V Device Recommended Operating Conditions" on page 32.
- (8) The FLEX 10KE input buffers are compatible with 2.5-V, 3.3-V LVTTTL and LVCMOS, 5.0-V TTL, and CMOS signals. Additionally, the input buffers are 3.3-V PCI compliant when V_{CCIO} and V_{CCINT} meet the relationship shown in Figure 20.
- (9) The I_{OH} parameter refers to high-level TTL, PCI, or CMOS output current.
- (10) The I_{OL} parameter refers to low-level TTL, PCI, or CMOS output current. This parameter applies to open-drain pins as well as output pins.
- (11) Pin pull-up resistance values will be lower if the pin is driven higher than V_{CCIO} by an external source.
- (12) Capacitance is sample-tested only.

Figure 20 shows the relationship between V_{CCIO} and V_{CCINT} for 3.3-V compliance.

Figure 20. Relationship between V_{CCIO} & V_{CCINT} for 3.3-V PCI Compliance

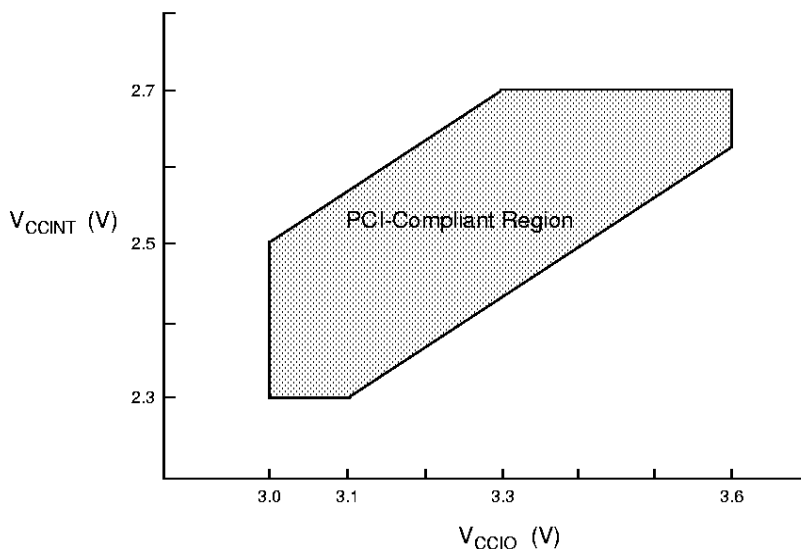
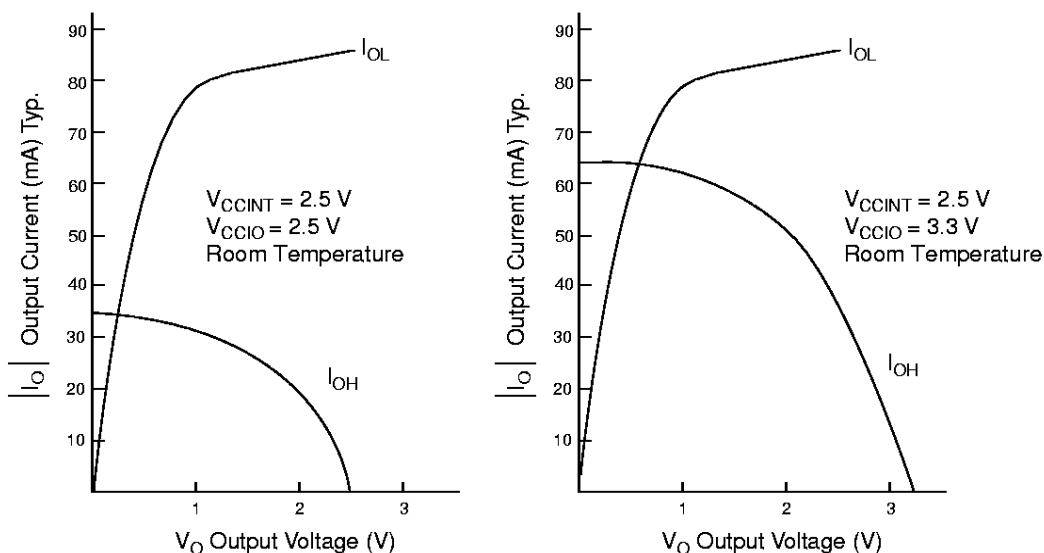


Figure 21 shows the typical output drive characteristics of FLEX 10KE devices with 3.3-V and 2.5-V V_{CCIO} . The output driver is compatible with the 3.3-V *PCI Local Bus Specification, Revision 2.1* (with 3.3-V V_{CCIO}).

Figure 21. Output Drive Characteristics of FLEX 10KE Devices



Timing Model

Table 15 shows the external timing parameters of FLEX 10KE devices. Detailed timing information for these devices will be released as it is available.

Table 15. FLEX 10KE External Timing Parameters									
Symbol	Parameter	Device	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
			Min	Max	Min	Max	Min	Max	
t_{DRR}	Register-to-register delay via 4 LEs, 3 row interconnects, and 4 local interconnects	EPF10K30E		8.5		10.0		13.5	ns
		EPF10K50E		8.5		10.0		13.5	ns
		EPF10K100B		11.0		12.0		14.5	ns
		EPF10K100E		10.0		12.0		16.0	ns
		EPF10K130E		10.0		12.0		16.0	ns
		EPF10K200E		10.0		12.0		16.0	ns
		EPF10K250E		11.0		13.5		17.0	ns

Table 20. FLEX 10KE FineLine β GA Device Pin-Outs (Part 3 of 4)*Note (1)*

Pin Name	256-Pin FineLine β GA EPF10K30E	256-Pin FineLine β GA EPF10K50E EPF10K100E EPF10K100 β	672-Pin FineLine β GA EPF10K130E	672-Pin FineLine β GA EPF10K200E EPF10K250E
No Connect (N.C.)	D1, E3, E16, G3, H1, H16, J1, K3, K14, K16, L2, L4, M14, M16, N15	—	A4, A5, A6, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20, A21, A22, A23, A24, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13, B16, B19, B20, B21, B22, B23, B24, C1, AE4, AE5, AE6, AE7, AE8, AE9, AE10, AE11, AE12, AE14, AE15, AE16, AE17, AF18, AE19, AE20, AE21, AE22, AE23, AF4, AF5, AF6, AF7, AF8, AF9, AF10, AF12, AF13, AF14, AF15, AF16, AF20, AF21, AF24, AF23	A3, A4, A5, A6, A8, A9, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20, A21, A22, A23, A24, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13, B14, B16, B19, B20, B21, B22, B23, B24, B26, C1, C25, C26, D1, D2, D25, D26, E1, E25, E26, F1, F25, G25, G26, H1, J1, J25, J26, K26, L2, L25, N2, P1, P2, R1, R26, T1, U1, U25, V1, V26, W1, Y26, AA1, AA2, AA25, AB2, AB25, AB26, AC1, AC2, AC25, AC26, AD2, AD26, AE1, AE3, AE4, AE5, AE6, AE7, AE8, AE9, AE10, AE11, AE12, AE14, AE15, AE16, AE17, AE19, AE20, AE21, AE22, AE23, AE24, AE26, AF3, AF4, AF5, AF6, AF7, AF8, AF9, AF10, AF11, AF12, AF13, AF14, AF15, AF16, AF17, AF18, AF20, AF21, AF23, AF24

Table 21. FLEX 10KE FineLine β GA Device Pin-Outs (Part 2 of 5) *Note (1)*

Pin Name	484-Pin FineLine β GA EPF10K30E	484-Pin FineLine β GA EPF10K50E	484-Pin FineLine β GA EPF10K100E	484-Pin FineLine β GA EPF10K130E
DATA5 (4)	D7	D7	D7	D7
DATA4 (4)	E7	E7	E7	E7
DATA3 (4)	F6	F6	F6	F6
DATA2 (4)	D5	D5	D5	D5
DATA1 (4)	E6	E6	E6	E6
DATA0 (2), (5)	D4	D4	D4	D4
TDI (2)	F5	F5	F5	F5
TDO (2)	F19	F19	F19	F19
TCK (2)	E18	E18	E18	E18
TMS (2)	U18	U18	U18	U18
TRST (2)	V19	V19	V19	V19
Dedicated Inputs	E12, H11, R12, V11	E12, H11, R12, V11	E12, H11, R12, V11	E12, H11, R12, V11
Dedicated Clock Pins	D12, P11	D12, P11	D12, P11	D12, P11
DEV_CLRn (3)	G11	G11	G11	G11
DEV_OE (3)	F12	F12	F12	F12
VCCINT (2.5 V)	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P12, P15, R14, V5, W21, Y8, AA12	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P12, P15, R14, V5, W21, Y8, AA12	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P12, P15, R14, V5, W21, Y8, AA12	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P12, P15, R14, V5, W21, Y8, AA12
VCCIO (2.5 or 3.3 V)	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13

Table 23 shows the FLEX 10KE device/package combinations that support SameFrame pin-outs. All FineLine BGA packages support SameFrame pin-outs providing the flexibility to migrate not only from device to device within the same package, but also from one package to another. The I/O count will vary, and MAX+PLUS II software versions 9.1 and higher provide features to help use only the common pins.

Table 23. FLEX 10KE SameFrame Pin-Out Support

Device	256-Pin FineLine BGA	484-Pin FineLine BGA	672-Pin FineLine BGA
EPF10K30E	✓	✓	
EPF10K50E	✓	✓	
EPF10K100B	✓		
EPF10K100E	✓	✓	
EPF10K130E		✓	✓
EPF10K200E			✓
EPF10K250E			✓

Revision History

The information contained in the *FLEX 10KE Embedded Programmable Logic Family Data Sheet* version 1.01 supersedes information published in previous versions.

The *FLEX 10KE Embedded Programmable Logic Family Data Sheet* version 1.01 contains the following change:

- ✎ Updated Table 15 (FLEX 10KE external timing parameters).