



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	88
Number of Logic Elements/Cells	880
Total RAM Bits	-
Number of I/O	171
Number of Gates	10000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	256-BGA
Supplier Device Package	256-FBGA (17x17)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/intel/epf6010afc256-3">https://www.e-xfl.com/product-detail/intel/epf6010afc256-3</a>

## ...and More Features

- Powerful I/O pins
  - Individual tri-state output enable control for each pin
  - Programmable output slew-rate control to reduce switching noise
  - Fast path from register to I/O pin for fast clock-to-output time
- 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 networks
  - Four low-skew global paths for clock, clear, preset, or logic signals
- Software design support and automatic place-and-route provided by Altera's development system for Windows-based PCs, Sun SPARCstations, and HP 9000 Series 700/800
- Flexible package options
  - Available in a variety of packages with 100 to 256 pins, including the innovative FineLine BGA™ packages (see [Table 2](#))
  - SameFrame™ pin-compatibility (with other FLEX® 6000 devices) across device densities and pin counts
  - Thin quad flat pack (TQFP), plastic quad flat pack (PQFP), and ball-grid array (BGA) packages (see [Table 2](#))
  - Footprint- and pin-compatibility with other FLEX 6000 devices in the same package
- Additional design entry and simulation support provided by EDIF 2.0.0 and 3.0.0 netlist files, the library of parameterized modules (LPM), Verilog HDL, VHDL, DesignWare components, and other interfaces to popular EDA tools from manufacturers such as Cadence, Exemplar Logic, Mentor Graphics, OrCAD, Synopsys, Synplicity, VeriBest, and Viewlogic

**Table 2. FLEX 6000 Package Options & I/O Pin Count**

Device	100-Pin TQFP	100-Pin FineLine BGA	144-Pin TQFP	208-Pin PQFP	240-Pin PQFP	256-Pin BGA	256-pin FineLine BGA
EPF6010A	71		102				
EPF6016			117	171	199	204	
EPF6016A	81	81	117	171			171
EPF6024A			117	171	199	218	219

## General Description

The Altera® FLEX 6000 programmable logic device (PLD) family provides a low-cost alternative to high-volume gate array designs. FLEX 6000 devices are based on the OptiFLEX architecture, which minimizes die size while maintaining high performance and routability. The devices have reconfigurable SRAM elements, which give designers the flexibility to quickly change their designs during prototyping and design testing. Designers can also change functionality during operation via in-circuit reconfiguration.

FLEX 6000 devices are reprogrammable, and they are 100% tested prior to shipment. As a result, designers are not required to generate test vectors for fault coverage purposes, allowing them to focus on simulation and design verification. In addition, the designer does not need to manage inventories of different gate array designs. FLEX 6000 devices are configured on the board for the specific functionality required.

Table 3 shows FLEX 6000 performance for some common designs. All performance values shown were obtained using 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 3. FLEX 6000 Device Performance for Common Designs**

Application	LEs Used	Performance			Units
		-1 Speed Grade	-2 Speed Grade	-3 Speed Grade	
16-bit loadable counter	16	172	153	133	MHz
16-bit accumulator	16	172	153	133	MHz
24-bit accumulator	24	136	123	108	MHz
16-to-1 multiplexer (pin-to-pin) (1)	10	12.1	13.4	16.6	ns
16 × 16 multiplier with a 4-stage pipeline	592	84	67	58	MHz

**Note:**

(1) This performance value is measured as a pin-to-pin delay.

Table 4 shows FLEX 6000 performance for more complex designs.

<b>Table 4. FLEX 6000 Device Performance for Complex Designs</b> <i>Note (1)</i>					
Application	LEs Used	Performance			Units
		-1 Speed Grade	-2 Speed Grade	-3 Speed Grade	
8-bit, 16-tap parallel finite impulse response (FIR) filter	599	94	80	72	MSPS
8-bit, 512-point fast Fourier transform (FFT) function	1,182	75 63	89 53	109 43	μS MHz
16450 universal asynchronous receiver/transmitter (UART)	487	36	30	25	MHz
PCI bus target with zero wait states	609	56	49	42	MHz

**Note:**

(1) The applications in this table were created using Altera MegaCore™ functions.

FLEX 6000 devices are supported by Altera development systems; 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 Altera 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 Altera software works easily with common gate array EDA tools for synthesis and simulation. For example, the Altera software can generate Verilog HDL files for simulation with tools such as Cadence Verilog-XL. Additionally, the Altera 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 Altera development systems include DesignWare functions that are optimized for the FLEX 6000 architecture.

The Altera development system runs on Windows-based PCs, Sun SPARCstations, and HP 9000 Series 700/800.

**f** See the *MAX+PLUS II Programmable Logic Development System & Software Data Sheet* and the *Quartus Programmable Logic Development System & Software Data Sheet* for more information.

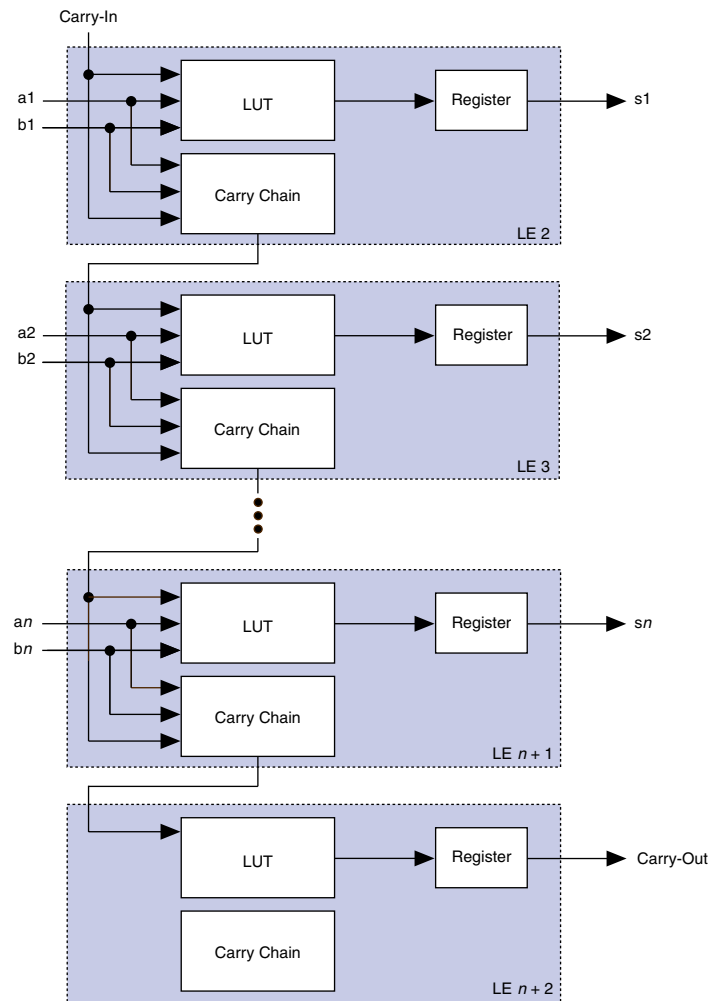
### *Carry Chain*

The carry chain provides a very fast (0.1 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 6000 architecture to implement high-speed counters, adders, and comparators of arbitrary width. Carry chain logic can be created automatically by the Altera software 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 for the appropriate functions.

Because the first LE of each LAB can generate control signals for that LAB, the first LE in each LAB is not included in carry chains. In addition, the inputs of the first LE in each LAB may be used to generate synchronous clear and load enable signals for counters implemented with carry chains.

Carry chains longer than nine LEs are implemented automatically 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 an even-numbered LAB to another even-numbered LAB, or from an odd-numbered LAB to another odd-numbered LAB. For example, the last LE of the first LAB in a row carries to the second LE of the third LAB in the row. In addition, the carry chain does not cross the middle of the row. For instance, in the EPF6016 device, the carry chain stops at the 11th LAB in a row and a new carry chain begins at the 12th LAB.

Figure 5 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. Although the register can be bypassed for simple adders, it can be 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 is driven onto the FastTrack Interconnect.

**Figure 5. Carry Chain Operation**

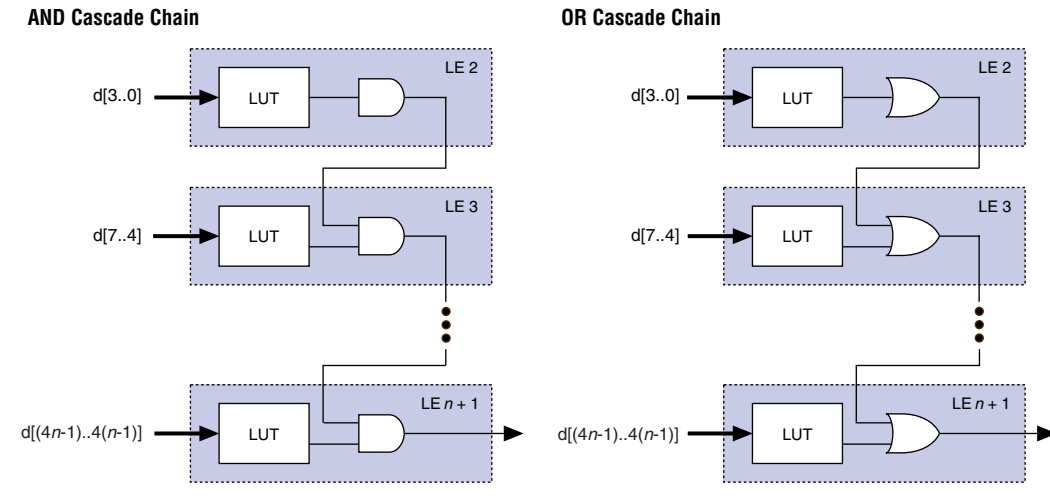
### *Cascade Chain*

The cascade chain enables the FLEX 6000 architecture to implement very wide fan-in functions. Adjacent LUTs can be used to implement 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 gate (via De Morgan's inversion) to connect the outputs of adjacent LEs. Each additional LE provides four more inputs to the effective width of a function, with a delay as low as 0.5 ns per LE. Cascade chain logic can be created automatically by the Altera software during design processing, or manually by the designer during design entry. Parameterized functions such as LPM and DesignWare functions automatically take advantage of cascade chains for the appropriate functions.

A cascade chain implementing an AND gate can use the register in the last LE; a cascade chain implementing an OR gate cannot use this register because of the inversion required to implement the OR gate.

Because the first LE of an LAB can generate control signals for that LAB, the first LE in each LAB is not included in cascade chains. Moreover, cascade chains longer than nine 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 an even-numbered LAB to another even-numbered LAB, or from an odd-numbered LAB to another odd-numbered LAB. For example, the last LE of the first LAB in a row cascades to the second LE of the third LAB. The cascade chain does not cross the center of the row. For example, in an EPF6016 device, the cascade chain stops at the 11th LAB in a row and a new cascade chain begins at the 12th LAB.

Figure 6 shows how the cascade function can connect adjacent LEs to form functions with a wide fan-in. In this example, functions of  $4n$  variables are implemented with  $n$  LEs. The cascade chain requires 3.4 ns to decode a 16-bit address.

**Figure 6. Cascade Chain Operation**

### LE Operating Modes

The FLEX 6000 LE can operate in one of the following three modes:

- Normal mode
- Arithmetic mode
- Counter mode

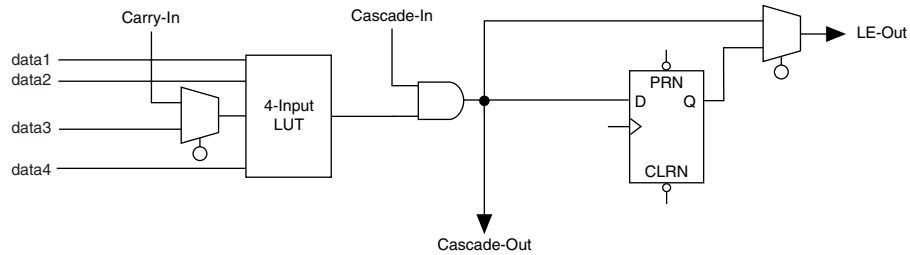
Each of these modes uses LE resources differently. In each mode, seven available inputs to the LE—the four data inputs from the LAB local interconnect, the feedback from the programmable register, and the carry-in and cascade-in from the previous LE—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, synchronous clear, and synchronous load control for the register. The Altera 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.

Figure 7 shows the LE operating modes.

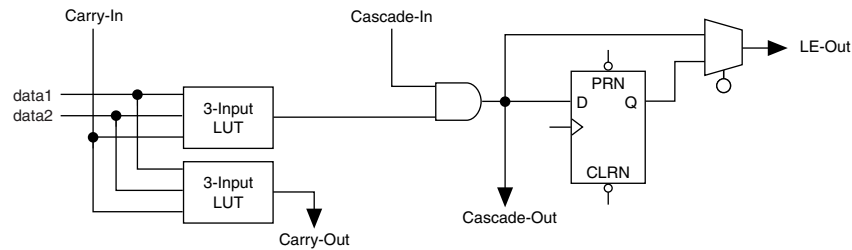


**Figure 7. LE Operating Modes**

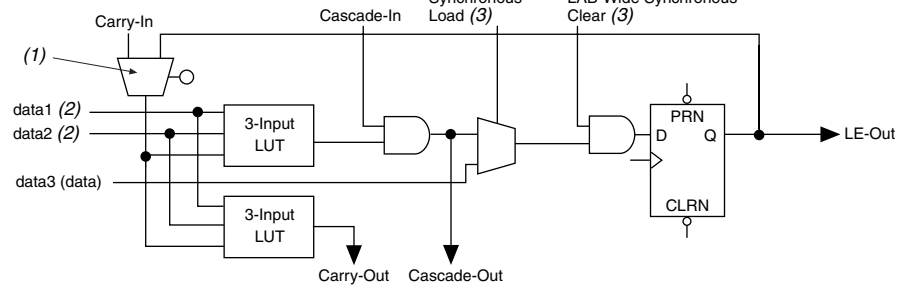
**Normal Mode**



**Arithmetic Mode**



**Counter Mode**



**Notes:**

- (1) The register feedback multiplexer is available on LE 2 of each LAB.
- (2) The data1 and data2 input signals can supply a clock enable, up or down control, or register feedback signals for all LEs other than the second LE in an LAB.
- (3) The LAB-wide synchronous clear and LAB-wide synchronous load affect all registers in an LAB.

**Normal Mode**

The normal mode is suitable for general logic applications, combinatorial functions, or 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 Altera software 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.

**Arithmetic Mode**

The arithmetic mode is ideal for implementing adders, accumulators, and comparators. An LE in arithmetic mode uses two 3-input LUTs. One LUT computes a 3-input function; the other generates a carry output. As shown in [Figure 7](#), 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, when implementing an adder, this output is the sum of three signals: DATA1, DATA2, 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.

The Altera software implements logic functions to use the arithmetic mode automatically where appropriate; the designer does not have to decide how the carry chain will be used.

**Counter Mode**

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

The counter mode uses two 3-input LUTs: one generates the counter data and the other generates the fast carry bit. A 2-to-1 multiplexer provides synchronous loading, and another AND gate provides synchronous clearing. If the cascade function is used by an LE in counter mode, the synchronous clear or load will override any signal carried on the cascade chain. The synchronous clear overrides the synchronous load.

Either the counter enable or the up/down control may be used for a given counter. Moreover, the synchronous load can be used as a count enable by routing the register output into the data input automatically when requested by the designer.

The second LE of each LAB has a special function for counter mode; the carry-in of the LE can be driven by a fast feedback path from the register. This function gives a faster counter speed for counter carry chains starting in the second LE of an LAB.

The Altera software implements functions to use the counter mode automatically where appropriate. The designer does not have to decide how the carry chain will be used.

### *Internal Tri-State Emulation*

Internal tri-state emulation provides internal tri-states 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 Altera 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 LAB-wide signals LABCTRL1 and LABCTRL2. The LE register has an asynchronous clear that can implement an asynchronous preset. Either LABCTRL1 or LABCTRL2 can control the asynchronous clear or preset. Because the clear and preset functions are active-low, the Altera software automatically assigns a logic high to an unused clear or preset signal. The clear and preset logic is implemented in either the asynchronous clear or asynchronous preset mode, which is chosen during design entry (see [Figure 8](#)).

A row channel can be driven by an LE or by one of two column channels. These three signals feed a 3-to-1 multiplexer that connects to six specific row channels. Row channels drive into the local interconnect via multiplexers.

Each column of LABs is served by a dedicated column interconnect. The LEs in an LAB can drive the column interconnect. The LEs in an LAB, a column IOE, or a row interconnect can drive the column interconnect. The column interconnect can then drive another row's interconnect to route the signals to other LABs in the device. A signal from the column interconnect must be routed to the row interconnect before it can enter an LAB.

Each LE has a FastTrack Interconnect output and a local output. The FastTrack interconnect output can drive six row and two column lines directly; the local output drives the local interconnect. Each local interconnect channel driven by an LE can drive four row and two column channels. This feature provides additional flexibility, because each LE can drive any of ten row lines and four column lines.

In addition, LEs can drive global control signals. This feature is useful for distributing internally generated clock, asynchronous clear, and asynchronous preset signals. A pin-driven global signal can also drive data signals, which is useful for high-fan-out data signals.

Each LAB drives two groups of local interconnects, which allows an LE to drive two LABs, or 20 LEs, via the local interconnect. The row-to-local multiplexers are used more efficiently, because the multiplexers can now drive two LABs. [Figure 10](#) shows how an LAB connects to row and column interconnects.

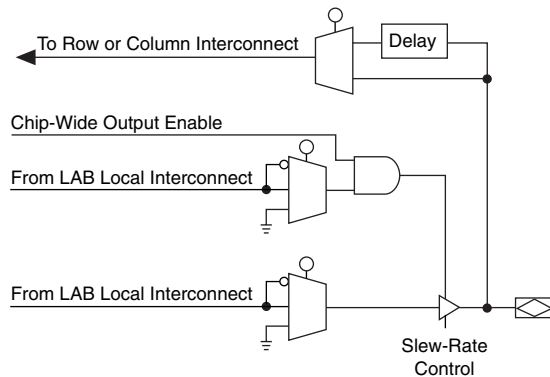
## I/O Elements

An IOE contains a bidirectional I/O buffer and a tri-state buffer. IOEs can be used as input, output, or bidirectional pins. An IOE receives its data signals from the adjacent local interconnect, which can be driven by a row or column interconnect (allowing any LE in the device to drive the IOE) or by an adjacent LE (allowing fast clock-to-output delays). A FastFLEX™ I/O pin is a row or column output pin that receives its data signals from the adjacent local interconnect driven by an adjacent LE. The IOE receives its output enable signal through the same path, allowing individual output enables for every pin and permitting emulation of open-drain buffers. The Altera Compiler uses programmable inversion to invert the data or output enable signals automatically where appropriate. Open-drain emulation is provided by driving the data input low and toggling the OE of each IOE. This emulation is possible because there is one OE per pin.

A chip-wide output enable feature allows the designer to disable all pins of the device by asserting one pin (DEV\_OE). This feature is useful during board debugging or testing.

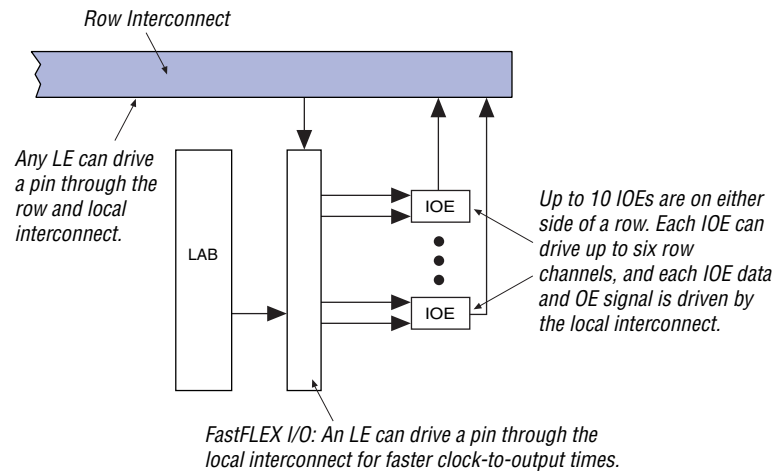
Figure 12 shows the IOE block diagram.

**Figure 12. IOE Block Diagram**



Each IOE drives a row or column interconnect when used as an input or bidirectional pin. A row IOE can drive up to six row lines; a column IOE can drive up to two column lines. The input path from the I/O pad to the FastTrack Interconnect has a programmable delay element that can be used to guarantee a zero hold time. Depending on the placement of the IOE relative to what it is driving, the designer may choose to turn on the programmable delay to ensure a zero hold time. Figure 13 shows how an IOE connects to a row interconnect, and Figure 14 shows how an IOE connects to a column interconnect.

**Figure 13. IOE Connection to Row Interconnect**



Open-drain output pins on 5.0-V or 3.3-V FLEX 6000 devices (with a pull-up resistor to the 5.0-V supply) can drive 5.0-V CMOS input pins that require a  $V_{IH}$  of 3.5 V. When the open-drain pin is active, it will drive low. When the pin is inactive, the trace will be pulled up to 5.0 V by the resistor. The open-drain pin will only drive low or tri-state; it will never drive high. The rise time is dependent on the value of the pull-up resistor and load impedance. The  $I_{OL}$  current specification should be considered when selecting a pull-up resistor.

Output pins on 5.0-V FLEX 6000 devices with  $V_{CCIO} = 3.3$  V or 5.0 V (with a pull-up resistor to the 5.0-V supply) can also drive 5.0-V CMOS input pins. In this case, the pull-up transistor will turn off when the pin voltage exceeds 3.3 V. Therefore, the pin does not have to be open-drain.

### Power Sequencing & Hot-Socketing

Because FLEX 6000 family devices can be used in a mixed-voltage environment, they have been designed specifically to tolerate any possible power-up sequence. The  $V_{CCIO}$  and  $V_{CCINT}$  power planes can be powered in any order.

Signals can be driven into 3.3-V FLEX 6000 devices before and during power up without damaging the device. Additionally, FLEX 6000 devices do not drive out during power up. Once operating conditions are reached, FLEX 6000 devices operate as specified by the user.

### IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All FLEX 6000 devices provide JTAG BST circuitry that comply with the IEEE Std. 1149.1-1990 specification. [Table 8](#) shows JTAG instructions for FLEX 6000 devices. JTAG BST can be performed before or after configuration, but not during configuration (except when you disable JTAG support in user mode).

- 1 See [Application Note 39 \(IEEE 1149.1 \(JTAG\) Boundary-Scan Testing in Altera Devices\)](#) for more information on JTAG BST circuitry.

**Table 8. FLEX 6000 JTAG Instructions**

JTAG Instruction	Description
SAMPLE/PRELOAD	Allows a snapshot of the signals at the device pins to be captured and examined during normal device operation, and permits an initial data pattern to be output at the device pins.
EXTEST	Allows the external circuitry and board-level interconnections to be tested by forcing a test pattern at the output pins and capturing test result at the input pins.
BYPASS	Places the 1-bit bypass register between the $TDI$ and $TDO$ pins, which allows the BST data to pass synchronously through the selected device to adjacent devices during normal device operation.

The instruction register length for FLEX 6000 devices is three bits. [Table 9](#) shows the boundary-scan register length for FLEX 6000 devices.

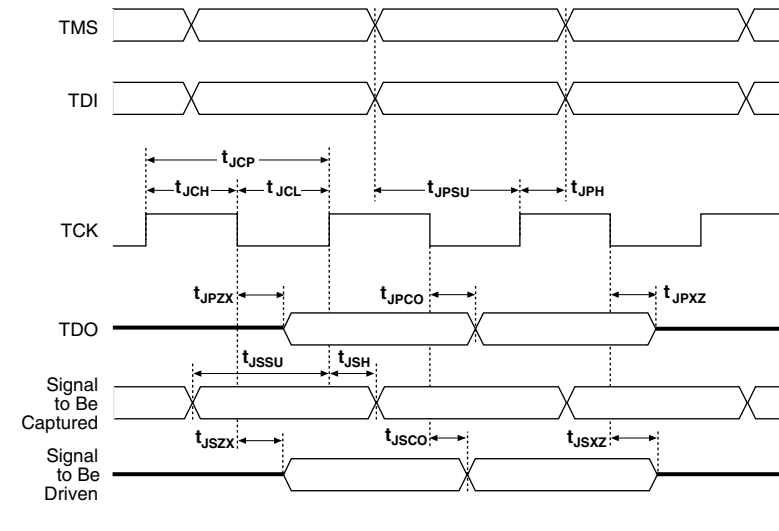
<b>Table 9. FLEX 6000 Device Boundary-Scan Register Length</b>	
<b>Device</b>	<b>Boundary-Scan Register Length</b>
EPF6010A	522
EPF6016	621
EPF6016A	522
EPF6024A	666

FLEX 6000 devices include a weak pull-up on JTAG pins.

f See [Application Note 39 \(IEEE 1149.1 \(JTAG\) Boundary-Scan Testing in Altera Devices\)](#) for more information.

[Figure 16](#) shows the timing requirements for the JTAG signals.

**Figure 16. JTAG Waveforms**



[Table 10](#) shows the JTAG timing parameters and values for FLEX 6000 devices.



## Operating Conditions

Tables 11 through 18 provide information on absolute maximum ratings, recommended operating conditions, operating conditions, and capacitance for 5.0-V and 3.3-V FLEX 6000 devices.

**Table 11. FLEX 6000 5.0-V Device Absolute Maximum Ratings** *Note (1)*

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{CC}$	Supply voltage	With respect to ground (2)	−2.0	7.0	V
$V_I$	DC input voltage		−2.0	7.0	V
$I_{OUT}$	DC output current, per pin		−25	25	mA
$T_{STG}$	Storage temperature	No bias	−65	150	° C
$T_{AMB}$	Ambient temperature	Under bias	−65	135	° C
$T_J$	Junction temperature	PQFP, TQFP, and BGA packages		135	° C

**Table 12. FLEX 6000 5.0-V Device Recommended Operating Conditions**

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{CCINT}$	Supply voltage for internal logic and input buffers	(3), (4)	4.75 (4.50)	5.25 (5.50)	V
$V_{CCIO}$	Supply voltage for output buffers, 5.0-V operation	(3), (4)	4.75 (4.50)	5.25 (5.50)	V
	Supply voltage for output buffers, 3.3-V operation	(3), (4)	3.00 (3.00)	3.60 (3.60)	V
$V_I$	Input voltage		−0.5	$V_{CCINT} + 0.5$	V
$V_O$	Output voltage		0	$V_{CCIO}$	V
$T_J$	Operating temperature	For commercial use	0	85	° C
		For industrial use	−40	100	° C
$t_R$	Input rise time			40	ns
$t_F$	Input fall time			40	ns

**Table 13. FLEX 6000 5.0-V Device DC Operating Conditions** *Notes (5), (6)*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IH}$	High-level input voltage		2.0		$V_{CCINT} + 0.5$	V
$V_{IL}$	Low-level input voltage		-0.5		0.8	V
$V_{OH}$	5.0-V high-level TTL output voltage	$I_{OH} = -8$ mA DC, $V_{CCIO} = 4.75$ V (7)	2.4			V
	3.3-V high-level TTL output voltage	$I_{OH} = -8$ mA DC, $V_{CCIO} = 3.00$ V (7)	2.4			V
	3.3-V high-level CMOS output voltage	$I_{OH} = -0.1$ mA DC, $V_{CCIO} = 3.00$ V (7)	$V_{CCIO} - 0.2$			V
$V_{OL}$	5.0-V low-level TTL output voltage	$I_{OL} = 8$ mA DC, $V_{CCIO} = 4.75$ V (8)			0.45	V
	3.3-V low-level TTL output voltage	$I_{OL} = 8$ mA DC, $V_{CCIO} = 3.00$ V (8)			0.45	V
	3.3-V low-level CMOS output voltage	$I_{OL} = 0.1$ mA DC, $V_{CCIO} = 3.00$ V (8)			0.2	V
$I_I$	Input pin leakage current	$V_I = V_{CC}$ or ground (8)	-10		10	$\mu$ A
$I_{OZ}$	Tri-stated I/O pin leakage current	$V_O = V_{CC}$ or ground (8)	-40		40	$\mu$ A
$I_{CC0}$	$V_{CC}$ supply current (standby)	$V_I =$ ground, no load		0.5	5	mA

**Table 14. FLEX 6000 5.0-V Device Capacitance** *Note (9)*

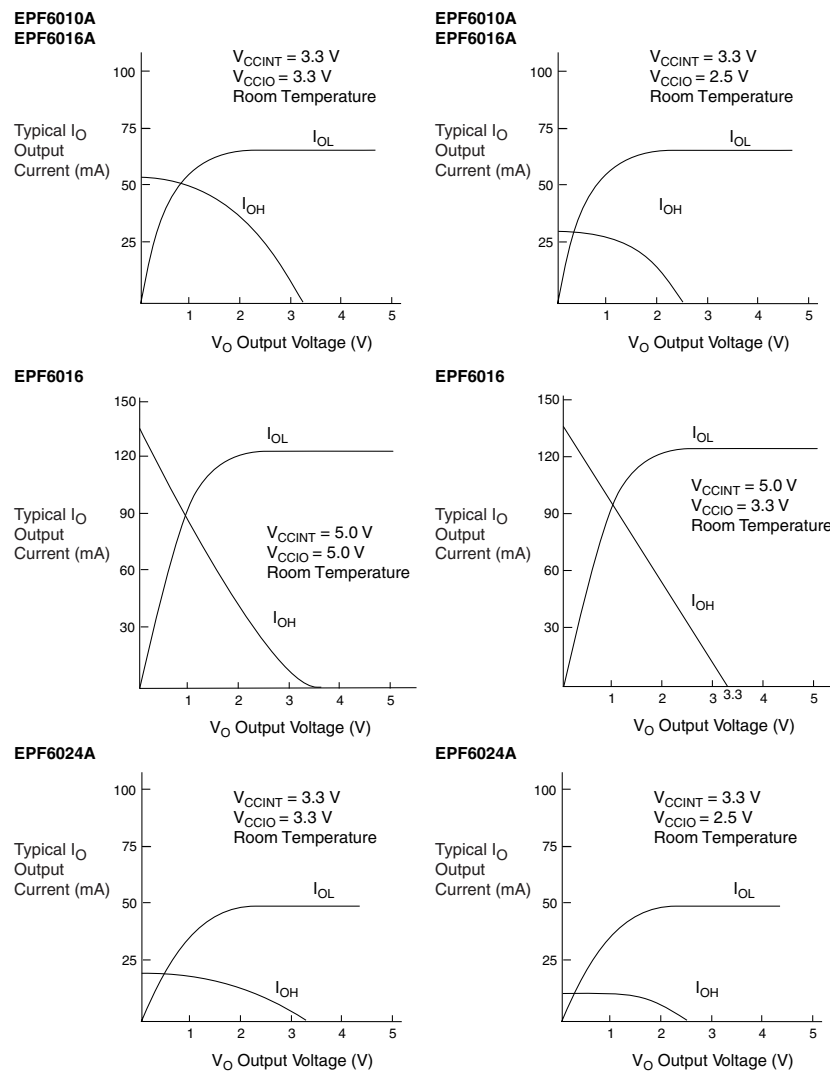
Symbol	Parameter	Conditions	Min	Max	Unit
$C_{IN}$	Input capacitance for I/O pin	$V_{IN} = 0$ V, $f = 1.0$ MHz		8	pF
$C_{INCLK}$	Input capacitance for dedicated input	$V_{IN} = 0$ V, $f = 1.0$ MHz		12	pF
$C_{OUT}$	Output capacitance	$V_{OUT} = 0$ V, $f = 1.0$ MHz		8	pF

**Notes to tables:**

- (1) See the *Operating Requirements for Altera Devices Data Sheet*.
- (2) Minimum DC input is -0.5 V. During transitions, the inputs may undershoot to -2.0 V or overshoot to 7.0 V for input currents less than 100 mA and periods shorter than 20 ns.
- (3) Numbers in parentheses are for industrial-temperature-range devices.
- (4) Maximum  $V_{CC}$  rise time to 100 ms.  $V_{CC}$  must rise monotonically.
- (5) Typical values are for  $T_A = 25^\circ$  C and  $V_{CC} = 5.0$  V.
- (6) These values are specified under the FLEX 6000 Recommended Operating Conditions shown in Table 12 on page 31.
- (7) The  $I_{OH}$  parameter refers to high-level TTL or CMOS output current.
- (8) 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.
- (9) Capacitance is sample-tested only.

Figure 18 shows the typical output drive characteristics of 5.0-V and 3.3-V FLEX 6000 devices with 5.0-V, 3.3-V, and 2.5-V  $V_{CCIO}$ . When  $V_{CCIO} = 5.0$  V on EPF6016 devices, the output driver is compliant with the *PCI Local Bus Specification, Revision 2.2* for 5.0-V operation. When  $V_{CCIO} = 3.3$  V on the EPF6010A and EPF6016A devices, the output driver is compliant with the *PCI Local Bus Specification, Revision 2.2* for 3.3-V operation.

Figure 18. Output Drive Characteristics



**Table 35. IOE Timing Microparameters for EPF6024A Devices**

Table 35. IOE Timing Microparameters for EPF6024A Devices							
Parameter	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{OD1}$		1.9		2.1		2.5	ns
$t_{OD2}$		4.0		4.4		5.3	ns
$t_{OD3}$		7.0		7.8		9.3	ns
$t_{XZ}$		4.3		4.8		5.8	ns
$t_{XZ1}$		4.3		4.8		5.8	ns
$t_{XZ2}$		6.4		7.1		8.6	ns
$t_{XZ3}$		9.4		10.5		12.6	ns
$t_{IOE}$		0.5		0.6		0.7	ns
$t_{IN}$		3.3		3.7		4.4	ns
$t_{IN\_DELAY}$		5.3		5.9		7.0	ns

**Table 36. Interconnect Timing Microparameters for EPF6024A Devices**

Table 36. Interconnect Timing Microparameters for EPF6024A Devices							
Parameter	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{LOCAL}$		0.8		0.8		1.1	ns
$t_{ROW}$		3.0		3.1		3.3	ns
$t_{COL}$		3.0		3.2		3.4	ns
$t_{DIN\_D}$		5.4		5.6		6.2	ns
$t_{DIN\_C}$		4.6		5.1		6.1	ns
$t_{LEGLOBAL}$		3.1		3.5		4.3	ns
$t_{LABCARRY}$		0.6		0.7		0.8	ns
$t_{LABCASC}$		0.3		0.3		0.4	ns

**Table 37. External Reference Timing Parameters for EPF6024A Devices**

Table 37. External Reference Timing Parameters for EPF6024A Devices							
Parameter	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t <sub>1</sub>		45.0		50.0		60.0	ns

## Operating Modes

The FLEX 6000 architecture uses SRAM configuration elements that require configuration data to be loaded every time the circuit powers up. This process of physically loading the SRAM data into a FLEX 6000 device is known as configuration. During initialization—a process that occurs immediately after configuration—the device resets registers, enables I/O pins, and begins to operate as a logic device. The I/O pins are tri-stated during power-up, and before and during configuration. The configuration and initialization processes of a device are referred to as *command mode*; normal device operation is called *user mode*.

SRAM configuration elements allow FLEX 6000 devices to be reconfigured in-circuit by loading new configuration data into the device. Real-time reconfiguration is performed by forcing the device into command mode with a device pin, loading different configuration data, reinitializing the device, and resuming user-mode operation. The entire reconfiguration process requires less than 100 ms and is used to dynamically reconfigure an entire system. Also, in-field system upgrades can be performed by distributing new configuration files.

## Configuration Schemes

The configuration data for a FLEX 6000 device can be loaded with one of three configuration schemes, which is chosen on the basis of the target application. An EPC1 or EPC1441 configuration device or intelligent controller can be used to control the configuration of a FLEX 6000 device, allowing automatic configuration on system power-up.

Multiple FLEX 6000 devices can be configured in any of the three configuration schemes by connecting the configuration enable input (nCE) and configuration enable output (nCEO) pins on each device.

Table 40 shows the data sources for each configuration scheme.

<b>Table 40. Configuration Schemes</b>	
<b>Configuration Scheme</b>	<b>Data Source</b>
Configuration device	EPC1 or EPC1441 configuration device
Passive serial (PS)	BitBlaster™, ByteBlasterMV™, or MasterBlaster™ download cables, or serial data source
Passive serial asynchronous (PSA)	BitBlaster, ByteBlasterMV, or MasterBlaster download cables, or serial data source