FLEX 6000

Programmable Logic Device Family

February 1998, ver. 3.01 Data Sheet

Features...

- Provides an ideal low-cost, programmable alternative to high-volume gate array applications and allows fast design changes during prototyping or design testing
- Product features
  - Register-rich, look-up table (LUT)-based architecture
  - OptiFLEX™ architecture that increases device area efficiency
  - Usable gates ranging from 8,000 to 24,000 gates (see Table 1)
  - Built-in low-skew clock distribution tree
  - 100% functional testing of all devices; test vectors or scan chains are not required
  - Advanced 3.2-mil (81-micron) bond pad pitch on 5.0-V devices for reduced die size
- System-level features
  - In-circuit reconfigurability (ICR) via external Configuration EPROM or intelligent controller
  - 5.0-V devices are fully compliant with peripheral component interconnect Special Interest Group's (PCI-SIG) ***PCI Local Bus Specification, Revision 2.1***
  - Built-in Joint Test Action Group (JTAG) boundary-scan test (BST) circuitry compliant with IEEE Std. 1149.1-1990, available without consuming additional device logic
  - MultiVolt™ I/O interface operation, allowing a device to bridge between systems operating at different voltages
  - Low power consumption (typical specification less than 0.5 mA in standby mode)

Table 1. FLEX 6000 Device Features

Feature	EPF6016	EPF6016A	EPF6024A
Typical gates (logic and RAM) *Note (1)*	8,000 to 16,000	8,000 to 16,000	12,000 to 24,000
Logic elements (LEs)	1,320	1,320	1,960
Maximum I/O pins	204	171	218
Supply voltage (V_{CCINT})	5.0 V	3.3 V	3.3 V

Note:

(1) Gate count varies from 6 to 12 gates per LE, depending on design method and contents. For instance, a pipelined design with many LE registers will have a higher gate count than a fully combinatorial design. For designs that require JTAG BST, the built-in JTAG circuitry contributes up to 14,000 additional gates.

...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 MAX+PLUS® II development system for 486- and Pentium-based PCs and Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations
- 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
- Extensive package options
  - 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

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

Device	144-Pin TQFP	208-Pin PQFP	240-Pin PQFP	256-Pin BGA
EPF6016	117	171	199	204
EPF6016A	117	171	–	–
EPF6024A	–	171	199	218

General Description

The Altera® FLEX 6000 programmable logic device (PLD) family provides a low-cost alternative to high-volume gate arrays designs. FLEX 6000 devices are based on the OptiFLEX architecture, which minimizes die size while maintaining high performance and routability. The devices also 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, 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 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
		-2 Speed Grade	-3 Speed Grade	
16-bit loadable counter	16	125	80	MHz
16-bit accumulator	16	125	80	MHz
24-bit accumulator	24	100	62	MHz
16-to-1 multiplexer (pin-to-pin), *Note (1)*	10	13.3	16.4	ns
16 × 16 multiplier with a 4-stage pipeline	592	71	49	MHz

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

Table 4 shows FLEX 6000 performance for more complex designs.

Table 4. FLEX 6000 Device Performance for Complex Designs *Note (1)*

Application	LEs Used	Performance		Units
		-2 Speed Grade	-3 Speed Grade	
16-bit, 8-tap parallel finite impulse response (FIR) filter	599	77	57	MSPS
8-bit, 512-point fast Fourier transform (FFT) function	1,166	108 44	135 35	µS MHz
`a16450` universal asynchronous receiver/transmitter (UART)	478	22	18	MHz
PCI bus target with zero wait states	596	33	25	MHz

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

FLEX 6000 devices are supported by Altera's 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 interfaces 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 6000 architecture.

The MAX+PLUS II software runs on 486- and Pentium-based PCs, and 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 this data book for more information.

Functional Description

The FLEX 6000 OptiFLEX architecture consists of logic elements (LEs). Each LE includes a 4-input look-up table (LUT), which can implement any 4-input function, a register, and dedicated paths for carry and cascade chain functions. Because each LE contains a register, a design can be easily pipelined without consuming more LEs. The specified gate count for FLEX 6000 devices includes all LUTs and registers.

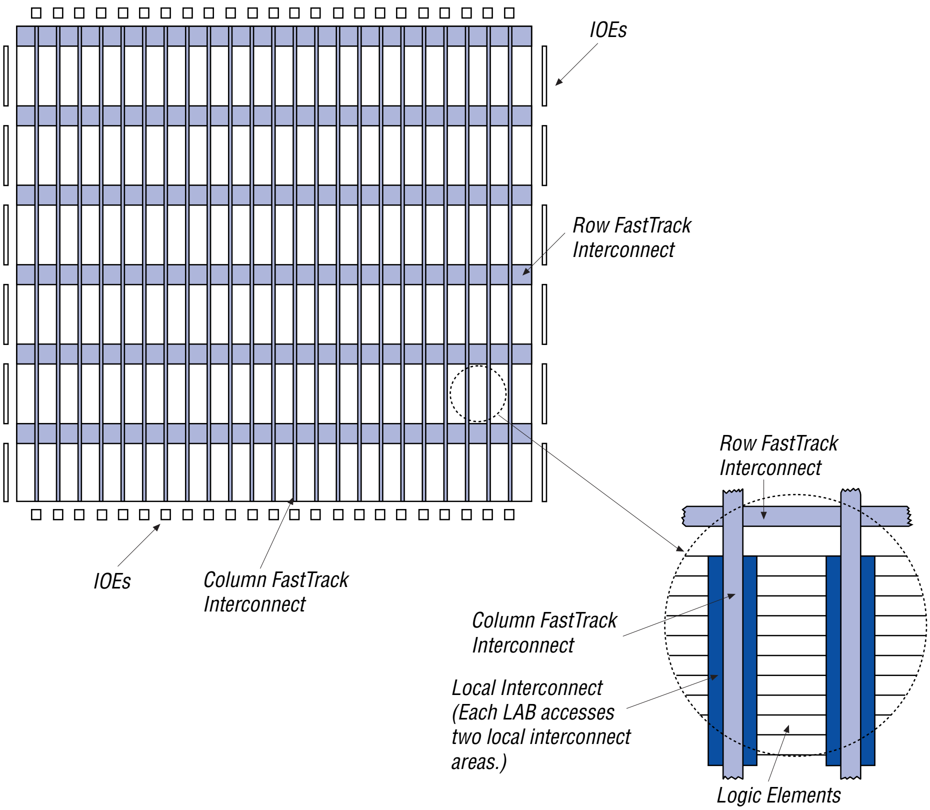
LEs are combined into groups called logic array blocks (LABs); each LAB contains 10 LEs. The MAX+PLUS II software automatically places related LEs into the same LAB, minimizing the number of required interconnects. Each LAB can implement a medium-sized block of logic, such as a counter or multiplexer.

Signal interconnections within FLEX 6000 devices—and to and from device pins—are provided via the routing structure of the FastTrack Interconnect. The routing structure is a series of fast, continuous row and column channels that run the entire length and width of the device. Any LE or pin can feed or be fed by any other LE or pin via the FastTrack Interconnect. See "FastTrack Interconnect" on page 213 of this data sheet for more information.

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. Each IOE contains a bidirectional I/O buffer. Each IOE is placed next to an LAB, where it can be driven by the local interconnect of that LAB. This feature allows fast clock-to-output times of less than 8 ns when a pin is driven by any of the 10 LEs in the adjacent LAB. Also, any LE can drive any pin via the row and column interconnect. I/O pins can drive the LE registers via the row and column interconnect, providing setup times of less than 4 ns and hold times of 0 ns. IOEs provide a variety of features, such as JTAG BST support, slew-rate control, and tri-state buffers.

Figure 1 shows a block diagram of the FLEX 6000 OptiFLEX architecture. Each group of ten LEs is combined into an LAB, and the LABs are arranged into rows and columns. The LABs are interconnected by the FastTrack Interconnect. IOEs are located at the end of each FastTrack Interconnect row and column.

Figure 1. OptiFLEX Architecture Block Diagram



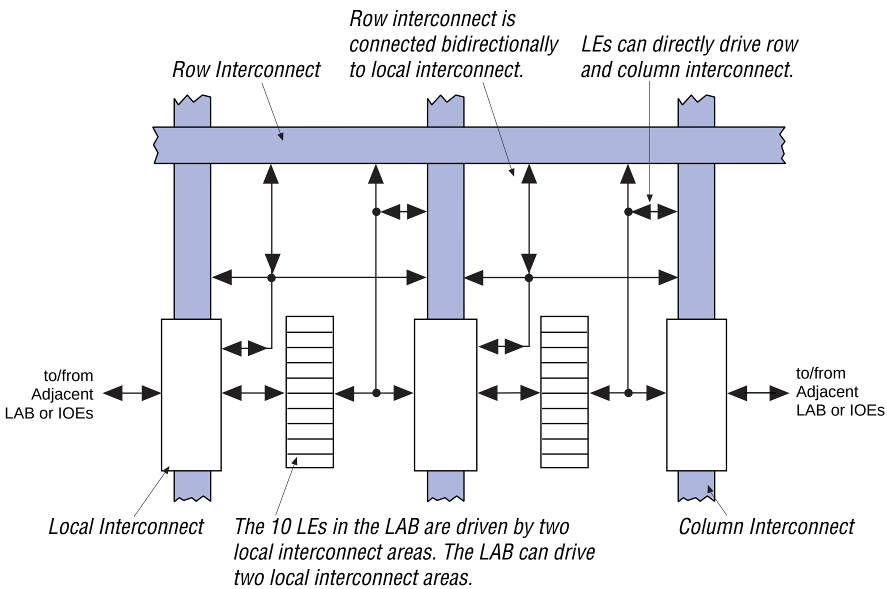
FLEX 6000 devices provide four dedicated, global inputs that drive the control inputs of the flipflops to ensure efficient distribution of high-speed, low-skew control signals. These inputs use dedicated routing channels that provide shorter delays and lower skews than the FastTrack Interconnect. These inputs 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. The dedicated global routing structure is built into the device, eliminating the need to create a clock tree.

Logic Array Block

An LAB consists of ten LEs, their associated carry and cascade chains, the LAB control signals, and the LAB local interconnect. The LAB provides the coarse-grained structure of the FLEX 6000 architecture, and facilitates efficient routing with optimum device utilization and high performance.

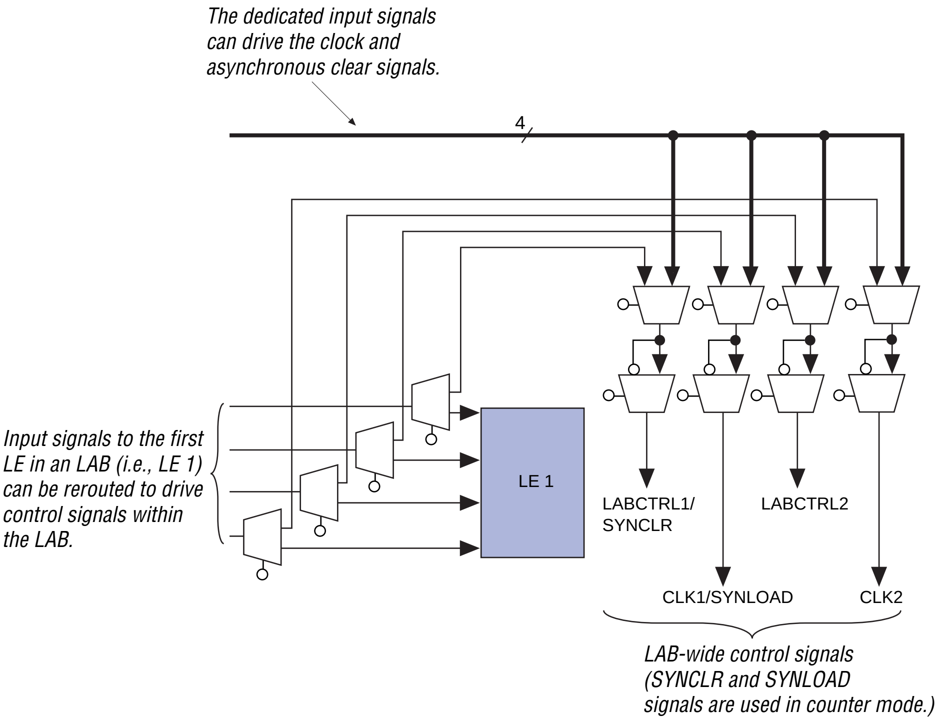
The interleaved LAB structure—an innovative feature of the FLEX 6000 architecture—allows each LAB to drive two local interconnects. This feature minimizes the use of the FastTrack Interconnect, providing higher performance. An LAB can drive 20 LEs via the local interconnect, which maximizes fitting flexibility while minimizing die size. See Figure 2.

Figure 2. Logic Array Block



In most designs, the registers only use global clock and clear signals. However, in some cases, other clock or asynchronous clear signals are needed. In addition, counters may also have synchronous clear or load signals. In a design that uses non-global clock and clear signals, inputs from the first LE in an LAB are re-routed to drive the control signals for that LAB. See Figure 3.

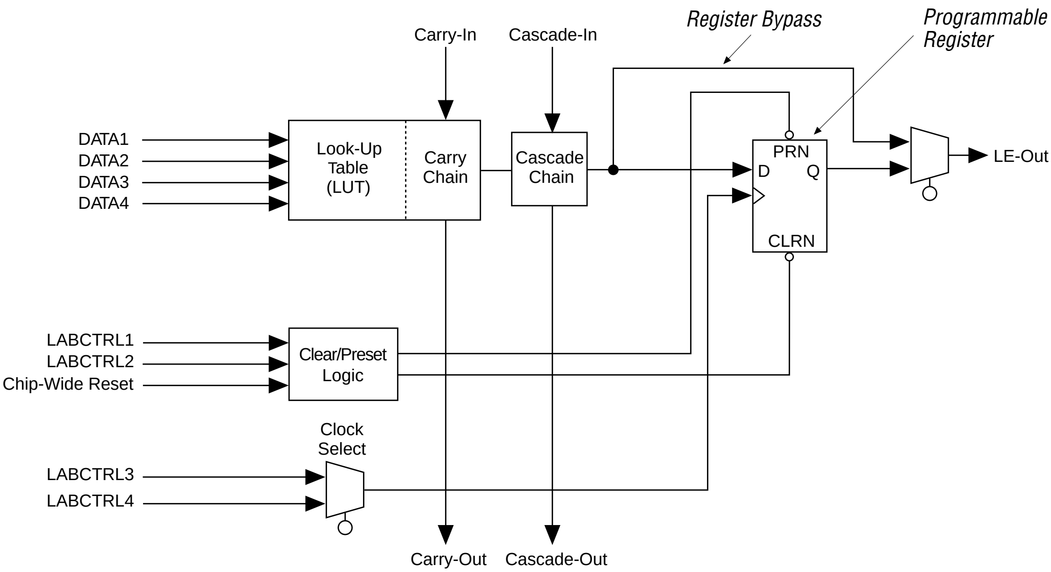
Figure 3. LAB Control Signals



Logic Element

An LE, the smallest unit of logic in the FLEX 6000 architecture, has a compact size that provides efficient logic usage. Each LE contains a four-input LUT, which is a function generator that can quickly implement any function of four variables. In addition, each LE contains a programmable flipflop, carry and cascade chains, and each LE drives both the local and the FastTrack Interconnect. See Figure 4.

Figure 4. Logic Element



The programmable flipflop in the LE can be configured for D, T, JK, or SR operation. The clock and clear 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 outputs of the LE. The LE output can drive both the local interconnect and the FastTrack interconnect.

The FLEX 6000 architecture provides two types of dedicated high-speed data paths that connect adjacent LEs without using local interconnect paths: carry chains and cascade chains. A carry chain supports high-speed arithmetic functions such as counters and adders, while a cascade chain implements wide-input functions such as equivalent comparators with minimum delay. Carry and cascade chains connect LEs 2 through 10 in an LAB and all LABs in the same half of the row. Because extensive use of carry and cascade chains can reduce routing flexibility, these chains should be limited to speed-critical portions of a design.

Carry Chain

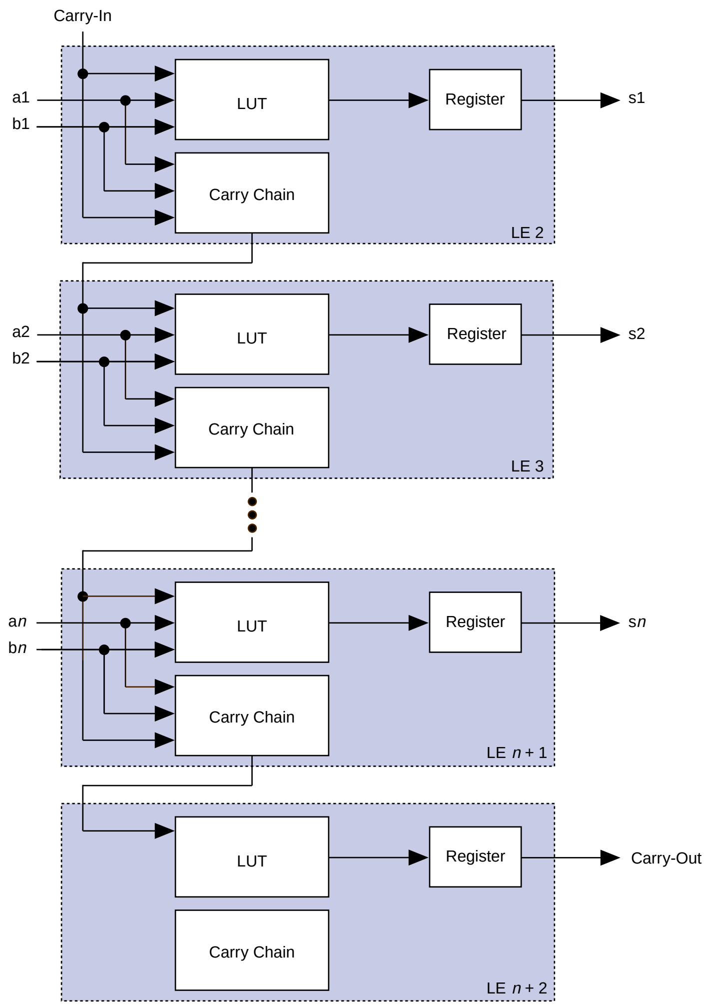
The carry chain provides a very fast (0.2 ns) carry-forward function between LEs. The carry-in signal from a lower-order bit drives forward into the higher-order bit via the carry chain, and feeds into both the LUT and the next portion of the carry chain. This feature allows the FLEX 6000 architecture to implement high-speed counters, adders, and comparators of arbitrary width. 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 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 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 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

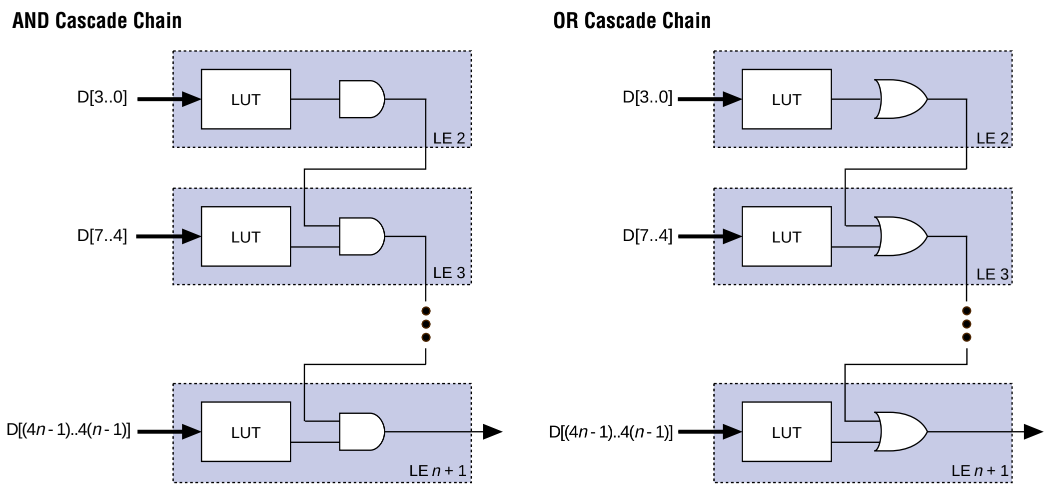
The cascade chain enables the FLEX 6000 architecture to implement very wide fan-in functions. Adjacent LUTs can be used to implement portions of the function in parallel; the cascade chain serially connects the intermediate values. The cascade chain can use a logical AND or logical OR gate (via De Morgan's 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 0.9 ns per LE. Cascade 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 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 one 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 4*n* variables are implemented with *n* LEs. With the cascade chain, 4.8 ns are needed to decode a 16-bit address.

Figure 6. Cascade Chain Operation



LE Operating Modes

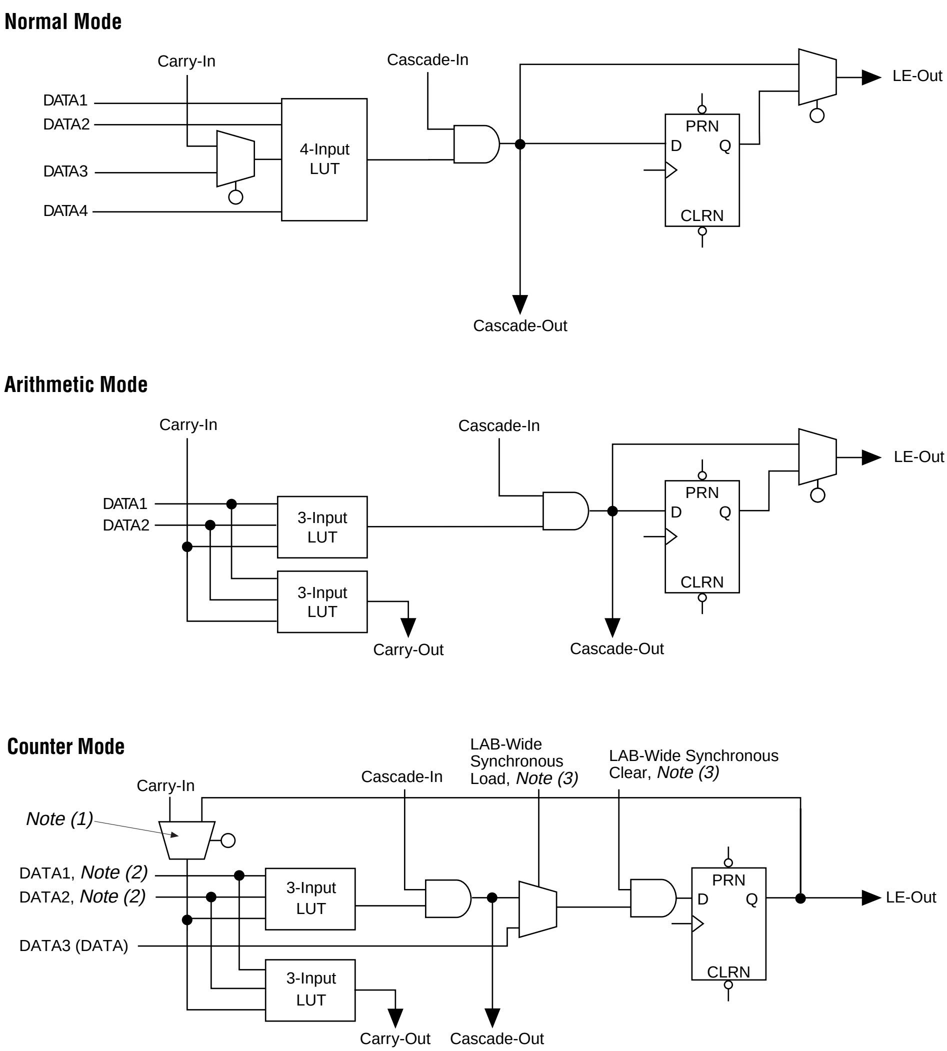
The FLEX 6000 LE can operate in one of the following three modes:

- 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 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.

Figure 7 shows the LE operating modes.

Figure 7. LE Operating Modes



Notes:
(1) A register feedback multiplexer is available on LE2 of each LAB.
(2) The `DATA1` and `DATA2` input signals can supply clock enable, up or down control, or register feedback signals for LEs other than the second LE in a 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 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.

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 MAX+PLUS II software implements parameterized 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 an 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 MAX+PLUS II development system will automatically place registers that are not in the counter into other LABs.

The counter mode uses two 3-input LUTs: one generates the counter data, 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 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. Counter functions will perform this function 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 MAX+PLUS II 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.

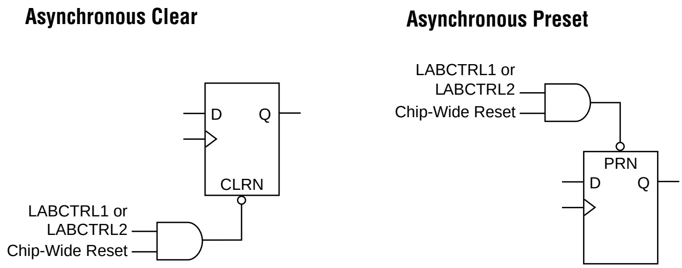
Internal Tri-State Emulation

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 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 LAB-wide signals `LABCTRL1` and `LABCTRL2`. The LE register has an asynchronous clear that can implement asynchronous preset. Either `LABCTRL1` or `LABCTRL2` can control the asynchronous clear or preset. Because the clear and preset functions are active-low, the MAX+PLUS II Compiler 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).

Figure 8. LE Clear & Preset Modes



Asynchronous Clear

The flipflop can be cleared by either `LABCTRL1` or `LABCTRL2`.

Asynchronous Preset

An asynchronous preset is implemented with an asynchronous clear. The MAX+PLUS II software provides 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, this technique can be used when a register drives logic or drives a pin.

In addition to the two clear and preset modes, FLEX 6000 devices provide a chip-wide reset pin (`DEV_CLRn`) that can reset all registers in the device. The option to use this pin is set in the MAX+PLUS II software before compilation. The chip-wide reset overrides all other control signals. Any register with an asynchronous preset will be preset when the chip-wide reset is asserted, which results from the inversion technique used to implement the asynchronous preset.

The MAX+PLUS II software can use a programmable `NOT`-gate-push back technique to emulate simultaneous preset and clear or asynchronous load. However, this technique uses an additional three LEs per register.

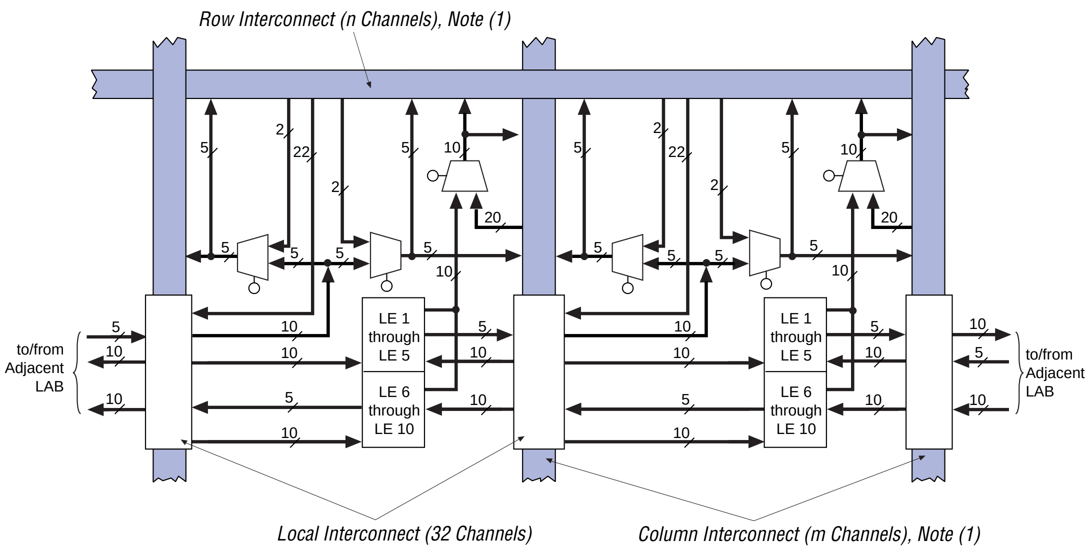
FastTrack Interconnect

In the FLEX 6000 OptiFLEX architecture, connections between LEs and device I/O pins are provided by the FastTrack Interconnect, a series of continuous horizontal and vertical routing channels that traverse the device. This global routing structure provides predictable performance, even for 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 consists of column and row interconnect channels that span the entire device. Each row of LABs is served by a dedicated row interconnect, which routes signals between LABs in the same row, and also routes signals from I/O pins to LABs. Additionally, the local interconnect routes signals between LEs in the same LAB and in adjacent LABs. The column interconnect routes signals between rows and routes signals from I/O pins to rows.

LEs 1 through 5 of an LAB drive the local interconnect to the right, while LEs 6 through 10 drive the local interconnect to the left. The `DATA1` and `DATA3` inputs of each LE are driven by the local interconnect to the left; `DATA2` and `DATA4` are driven by the local interconnect to the right. The local interconnect also routes signals from LEs to I/O pins. Figure 9 shows an overview of the FLEX 6000 interconnect architecture. LEs in the first and last columns have drivers on both sides so that all LEs in the LAB can drive I/O pins via the local interconnect.

Figure 9. FastTrack Interconnect Architecture



Note:

(1) For EPF6016 and EPF6016A devices, $n = 144$ channels and $m = 20$ channels; for EPF6024A devices, $n = 186$ channels and $m = 30$ channels.

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.

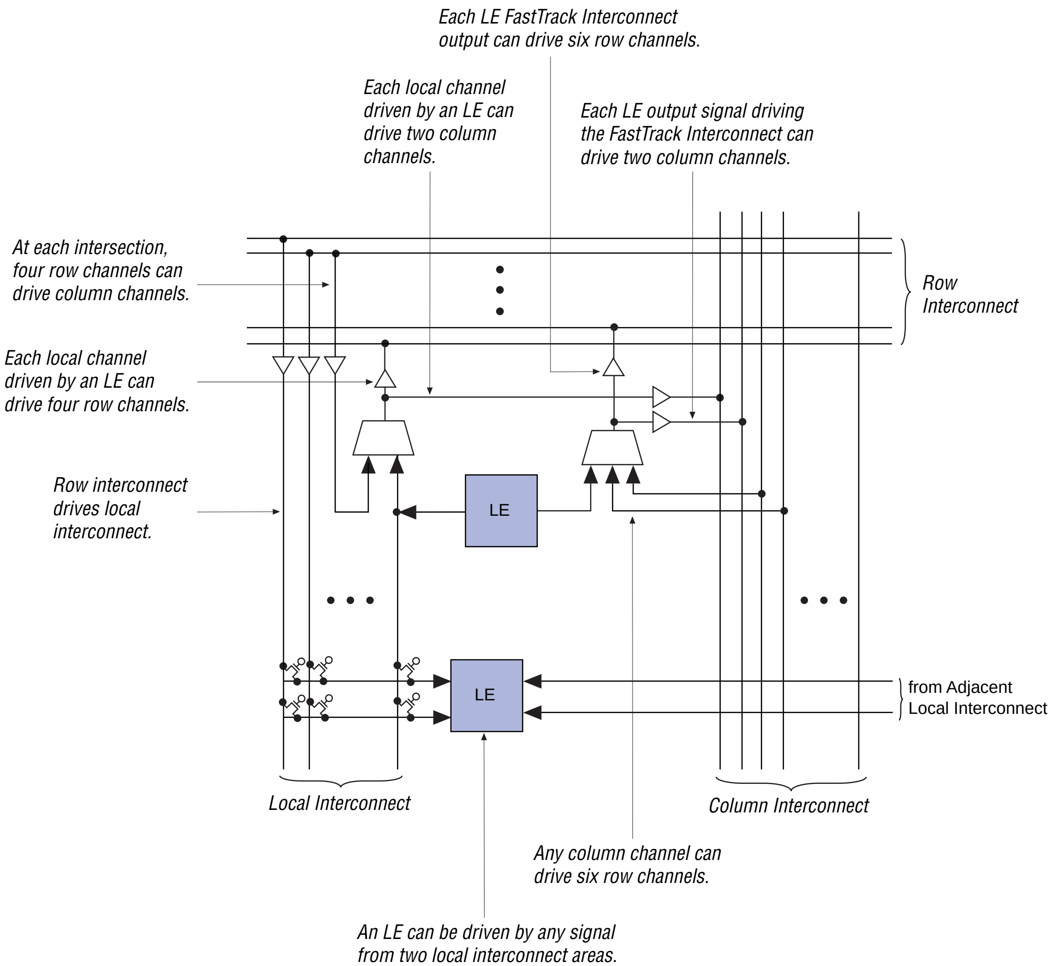
Each column of LABs is served by a dedicated column interconnect. The LEs in an LAB can drive the column interconnect. The LEs in an LAB, a column IOE, or a row interconnect can drive the column interconnect. The column interconnect can then drive another row's interconnect to route the signals to other 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 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.

Figure 10. LAB Connections to Row & Column Interconnects

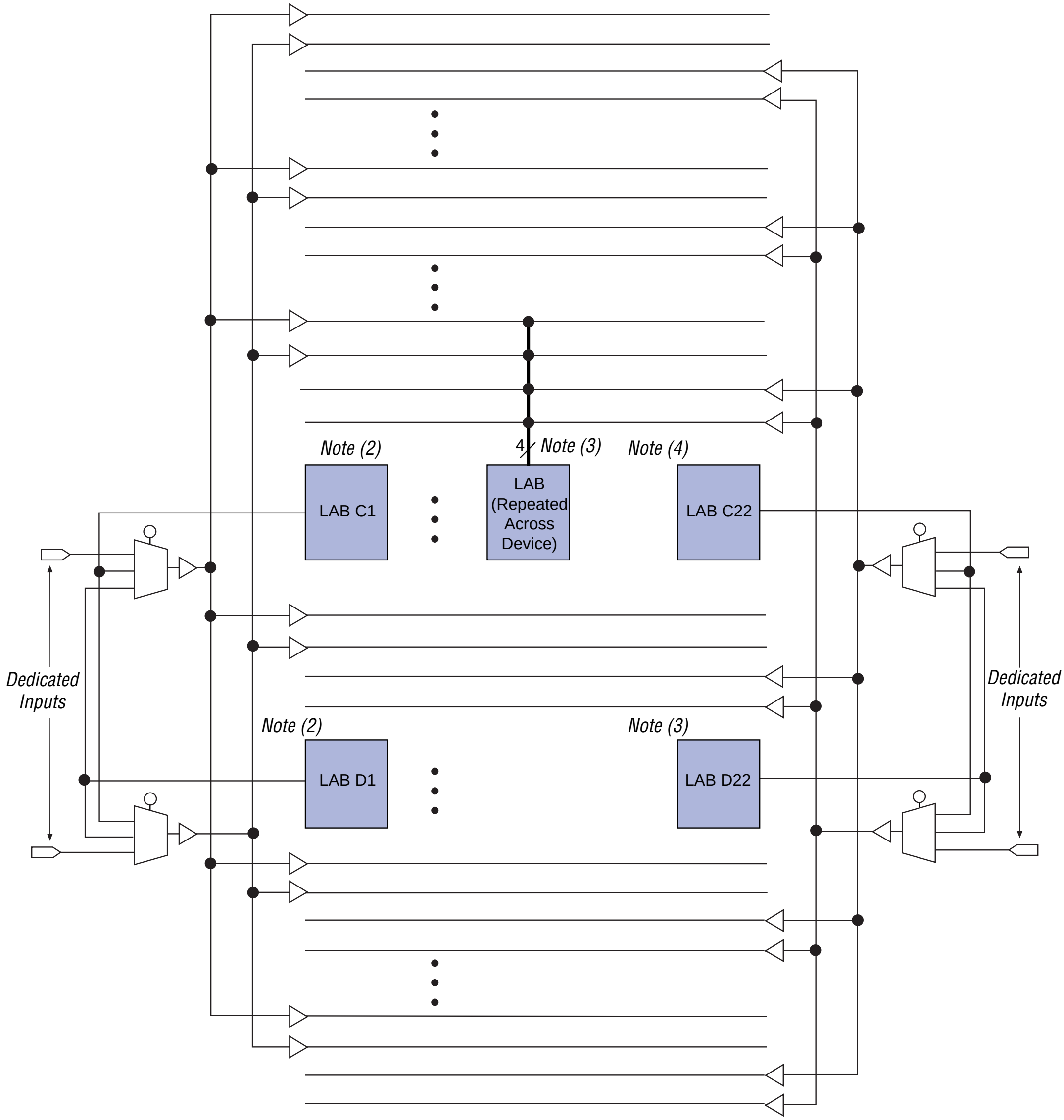


For improved routability, the row interconnect is comprised of full-length and half-length channels. The full-length channels connect to all LABs in a row; the half length channels connect to the LABs in half of the row. In addition to providing a predictable, row-wide interconnect, this architecture provides increased routing resources. Two neighboring LABs can be connected using a half-row channel, which saves the other half of the channel for the other half of the row. One-third of row channels are half-length channels.

In addition to general-purpose I/O pins, FLEX 6000 devices have four dedicated input pins that provide low-skew signal distribution across the device. These four inputs can be used for global clock and asynchronous clear control signals. These signals are available as control signals for all LEs 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. Using dedicated inputs to route data signals provides a fast path for high fan-out signals. However, the use of dedicated inputs as data inputs can introduce additional delay into the control signal network.

The local interconnect from LABs located at either end of two rows can drive a global control signal. For instance, in an EPF6016 device, LABs C1, D1, C22, and D22 can all drive global control signals. When an LE drives a global control signal, the dedicated input pin that drives that signal cannot be used. Any LE in the device can drive a global control signal by driving the FastTrack Interconnect into the appropriate LAB. To minimize delay, however, the MAX+PLUS II software will place the driving LE in the appropriate LAB. This LE-driving-global control feature is controlled by the designer and is not automatically used by the MAX+PLUS II software. See Figure 11.

Figure 11. Global Clock & Clear Distribution *Note (1)*



Notes:

(1) The global clock and clear distribution signals are shown for EPF6016 and EPF6016A devices. In EPF6024A devices, LABs in rows C and E drive global signals.

(2) The local interconnect from LABs C1 and D1 can drive two global control signals on the left side.

(3) Global signals drive into every LAB as clock, asynchronous clear, preset, and data signals.

(4) The local interconnect from LABs C22 and D22 can drive two global control lines on the right side.

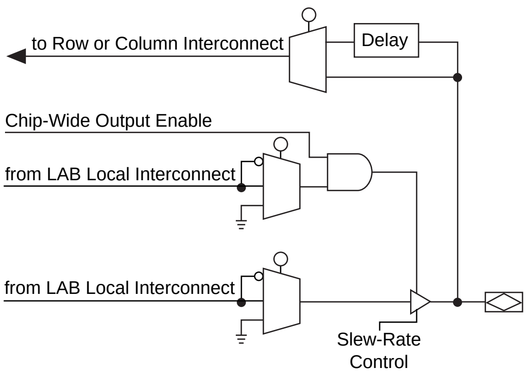
I/O Elements

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). LEs in LABs on both edges of a row have local drivers on both sides so that all LEs in an LAB can drive I/O pins via the local interconnect. 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 MAX+PLUS II Compiler uses the programmable inversion option to automatically invert the data or output enable signals where appropriate.

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. 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.

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 drives one of 6 row lines; a column IOE drives one of 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

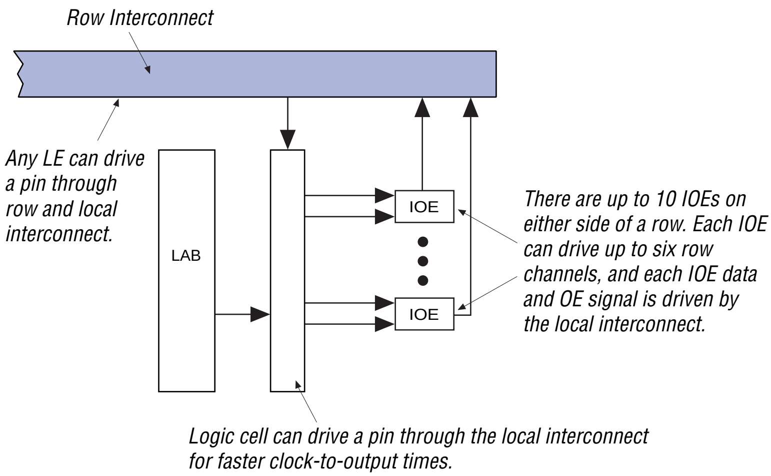
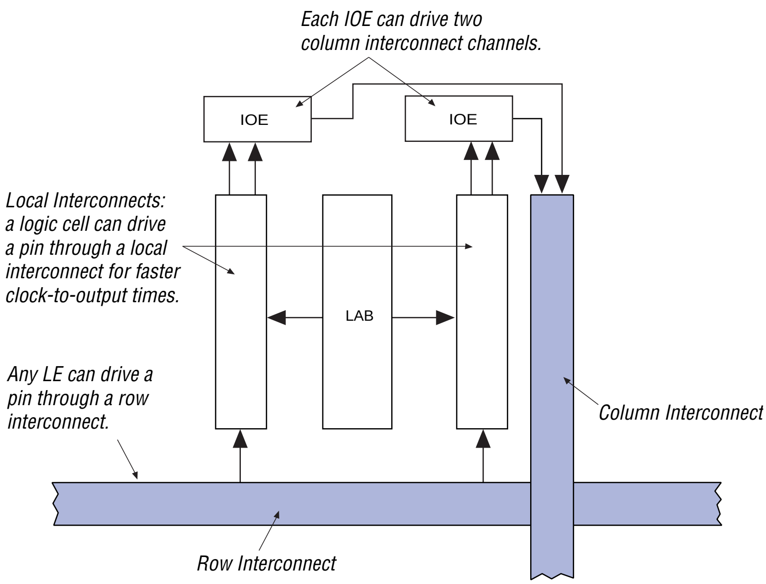


Figure 14. IOE Connection to Column Interconnect



Output Configuration

This section discusses slew-rate control and MultiVolt I/O interface for FLEX 6000 devices.

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 2.4 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 on a pin-by-pin basis during design entry or assign a default slew rate to all pins on a device-wide basis. The slew rate setting affects only the falling edge of the output.

MultiVolt I/O Interface

The FLEX 6000 device architecture supports the MultiVolt I/O interface feature, which allows FLEX 6000 devices to interface of systems with differing supply voltages. The EPF6016 device can be set for 3.3-V or 5.0-V I/O pin operation. This device has one set of V_{CC} pins for internal operation and input buffers (`VCCINT`), and another set for output drivers (`VCCIO`).

The `VCCINT` pins on 5.0-V FLEX 6000 devices must always be connected to a 5.0-V power supply. With a 5.0-V V_{CCINT} level, input voltages are at TTL levels and are therefore compatible with 3.3-V and 5.0-V inputs.

The `VCCIO` pins on 5.0-V FLEX 6000 devices can be connected to either a 3.3-V or 5.0-V power supply, depending on the output requirements. When the `VCCIO` pins are connected to a 5.0-V power supply, the output levels are compatible with 5.0-V systems. When the `VCCIO` pins are connected to a 3.3-V power supply, the output high is 3.3 V and is therefore compatible with 3.3-V or 5.0-V systems. Devices operating with V_{CCIO} levels lower than 4.75 V incur a nominally greater timing delay of t_{OD2} instead of t_{OD1}.

Additionally, 3.3-V FLEX 6000A devices can interface with 2.5-V, 3.3-V, or 5.0-V systems when `VCCIO` is tied to 2.5 V. The output will drive 2.5-V systems, and the inputs can be driven by 2.5-V, 3.3-V, or 5.0-V systems. When `VCCIO` is tied to 3.3 V, the output can drive 3.3-V or 5.0-V systems.

IEEE 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 5 shows JTAG instructions for FLEX 6000 devices. JTAG BST can be performed before or after configuration, but not during configuration.

Table 5. 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 `TDI` and `TDO` pins, which allows the BST data to pass synchronously through the selected device to adjacent devices during normal device operation.

Go to *Application Note 39 (JTAG Boundary-Scan Testing in Altera Devices)* for more information.

Figure 15 shows the timing requirements for the JTAG signals.

Figure 15. JTAG Waveforms

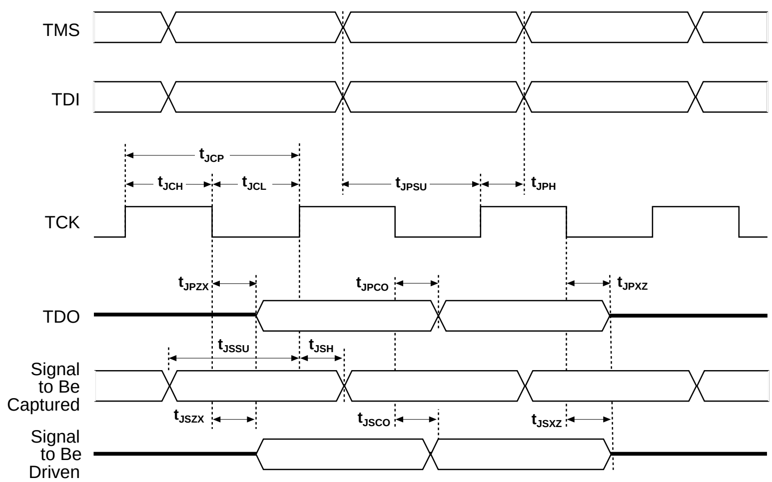


Table 6 shows the JTAG timing parameters and values for FLEX 6000 devices.

Table 6. JTAG Timing Parameters & Values

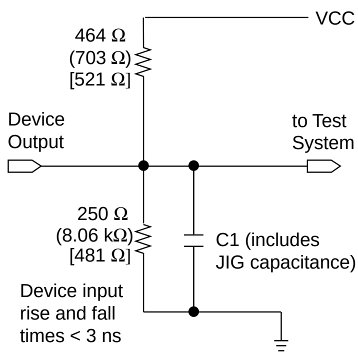
Symbol	Parameter	Min	Max	Unit
t_{JCP}	TCK clock period	100		ns
t_{JCH}	TCK clock high time	50		ns
t_{JCL}	TCK clock low time	50		ns
t_{JPSU}	JTAG port setup time	20		ns
t_{JPH}	JTAG port hold time	45		ns
t_{JPCO}	JTAG port clock to output		25	ns
t_{JPZX}	JTAG port high impedance to valid output		25	ns
t_{JPXZ}	JTAG port valid output to high impedance		25	ns
t_{JSSU}	Capture register setup time	20		ns
t_{JSH}	Capture register hold time	45		ns
t_{JSCO}	Update register clock to output		35	ns
t_{JSZX}	Update register high-impedance to valid output		35	ns
t_{JSXZ}	Update register valid output to high impedance		35	ns

Generic Testing

Each FLEX 6000 device is functionally tested. Complete testing of each configurable SRAM bit and all logic functionality ensures 100% configuration yield. AC test measurements for FLEX 6000 devices are made under conditions equivalent to those shown in Figure 16. Multiple test patterns can be used to configure devices during all stages of the production flow.

Figure 16. AC Test Conditions

Power supply transients can affect AC measurements. Simultaneous transitions of multiple outputs should be avoided for accurate measurement. Threshold tests must not be performed under AC conditions. Large-amplitude, fast-ground-current transients normally occur as the device outputs discharge the load capacitances. When these transients flow through the parasitic inductance between the device ground pin and the test system ground, significant reductions in observable noise immunity can result. Numbers in parentheses are for 3.3-V devices or outputs. Numbers in brackets are for 2.5-V devices or outputs.



Operating Conditions

The following tables provide information on absolute maximum ratings, recommended operating conditions, operating conditions, and capacitance for 5.0-V and 3.3-V FLEX 6000 devices.

5.0-V Device Absolute Maximum Ratings *Note (1)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CC}	Supply voltage	With respect to ground, *Note (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, PLCC, and BGA packages		135	° C

5.0-V Device Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	*Notes (3)*, *(4)*	4.75 (4.50)	5.25 (5.50)	V
V_{CCIO}	Supply voltage for output buffers, 5.0-V operation	*Notes (3)*, *(4)*	4.75 (4.50)	5.25 (5.50)	V
	Supply voltage for output buffers, 3.3-V operation	*Notes (3)*, *(4)*	3.00 (3.00)	3.60 (3.60)	V
V_I	Input voltage		0	V_{CCINT}	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

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} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		0.8	V
V_{OH}	5.0-V high-level TTL output voltage	I_{OH} = –4 mA DC, V_{CCIO} = 4.75 V, *Note (7)*	2.4			V
	3.3-V high-level TTL output voltage	I_{OH} = –4 mA DC, V_{CCIO} = 3.00 V, *Note (7)*	2.4			V
	3.3-V high-level CMOS output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 3.00 V, *Note (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, *Note (8)*			0.45	V
	3.3-V low-level TTL output voltage	I_{OL} = 8 mA DC, V_{CCIO} = 3.00 V, *Note (8)*			0.45	V
	3.3-V low-level CMOS output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 3.00 V, *Note (8)*			0.2	V
I_I	Input pin leakage current	V_I = V_{CC} or ground	–10		10	μA
I_{OZ}	Tri-stated I/O pin leakage current	V_O = V_{CC} or ground	–40		40	μA
I_{CC0}	V_{CC} supply current (standby)	V_I = ground, no load		0.5	5	mA

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 *Operating Requirements for Altera Devices* in this data book.
(2) Minimum DC input is –0.3 V. During transitions, the inputs may undershoot to –2.0 V or overshoot to 7.0 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 to 100 ms. V_{CC} must rise monotonically.
(5) Typical values are for T_A = 25° C and V_{CC} = 5.0 V.
(6) These values are specified under "5.0-V Device recommended Operating Conditions" on page 224.
(7) The I_{OH} parameter refers to high-level TTL or CMOS output current.
(8) The I_{OL} parameter refers to low-level TTL or CMOS output current. This parameter applies to open-drain pins as well as output pins.
(9) Capacitance is sample-tested only.

3.3-V Device Absolute Maximum Ratings *Notes (1), (3)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CC}	Supply voltage	With respect to ground, *Note (2)*	–0.5	4.6	V
V_I	DC input voltage		–0.5	5.7	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, PLCC, and BGA packages		135	° C

3.3-V Device Recommended Operating Conditions *Note (3)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	*Notes (4), (5)*	3.00 (3.00)	3.60 (3.60)	V
V_{CCIO}	Supply voltage for output buffers, 3.3-V operation	*Notes (4), (5)*	3.00 (3.00)	3.60 (3.60)	V
	Supply voltage for output buffers, 2.5-V operation	*Notes (4), (5)*	2.30 (2.30)	2.70 (2.70)	V
V_I	Input voltage		0	5.3	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

3.3-V Device DC Operating Conditions *Notes (3), (6), (7)*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	High-level input voltage		1.7		5.3	V
V_{IL}	Low-level input voltage		–0.3		0.8	V
V_{OH}	3.3-V high-level TTL output voltage	I_{OH} = –4 mA DC, V_{CCIO} = 3.00 V, *Note (8)*	2.4			V
	3.3-V high-level CMOS output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 3.00 V, *Note (8)*	V_{CCIO} – 0.2			V
	2.5-V high-level output voltage	I_{OH} = –100 µA DC, V_{CCIO} = 2.30 V, *Note (8)*	2.1			V
		I_{OH} = –1 mA DC, V_{CCIO} = 2.30 V, *Note (8)*	2.0			V
		I_{OH} = –2 mA DC, V_{CCIO} = 2.30 V, *Note (8)*	1.7			V
V_{OL}	3.3-V low-level TTL output voltage	I_{OL} = 4 mA DC, V_{CCIO} = 3.00 V, *Note (9)*			0.45	V
	3.3-V low-level CMOS output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 3.00 V, *Note (9)*			0.2	V
	2.5-V low-level output voltage	I_{OL} = 100 µA DC, V_{CCIO} = 2.30 V, *Note (9)*			0.2	V
		I_{OL} = 1 mA DC, V_{CCIO} = 2.30 V, *Note (9)*			0.4	V
		I_{OL} = 2 mA DC, V_{CCIO} = 2.30 V, *Note (9)*			0.7	V
I_I	Input pin leakage current	$V_I = V_{CC}$ or ground	–10		10	µA
I_{OZ}	Tri-stated I/O pin leakage current	$V_O = V_{CC}$ or ground	–40		40	µA
I_{CC0}	V_{CC} supply current (standby)	V_I = ground, no load		0.5		mA

3.3-V Device Capacitance *Notes (3), (10)*

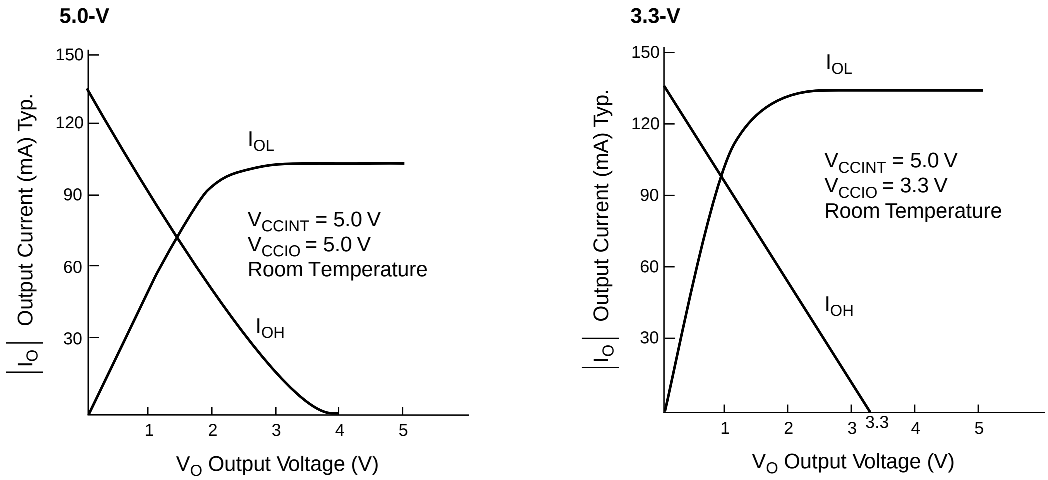
Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Input capacitance for I/O pin	V_{IN} = 0 V, f = 1.0 MHz		8	pF
C_{INCLK}	Input capacitance for dedicated input	V_{IN} = 0 V, f = 1.0 MHz		12	pF
C_{OUT}	Output capacitance	V_{OUT} = 0 V, f = 1.0 MHz		8	pF

Notes to tables:

(1) See *Operating Requirements for Altera Devices* in this data book.
(2) Minimum DC input is –0.3 V. During transitions, the inputs may undershoot to –0.5 V or overshoot to 5.7 V for periods shorter than 20 ns under no-load conditions.
(3) This information is preliminary. For the most up-to-date information, contact Altera Applications at (800) 800-EPLD.
(4) Numbers in parentheses are for industrial-temperature-range devices.
(5) Maximum V_{CC} rise time is 100 ms. V_{CC} must rise monotonically.
(6) Typical values are for T_A = 25° C and V_{CC} = 3.3 V.
(7) These values are specified under "3.3-V Device recommended Operating Conditions" on page 226.
(8) The I_{OH} parameter refers to high-level TTL or CMOS output current.
(9) The I_{OL} parameter refers to low-level TTL or CMOS output current. This parameter applies to open-drain pins as well as output pins.
(10) Capacitance is sample-tested only.

Figure 17 shows the typical output drive characteristics of FLEX 6000 devices with 5.0-V and 3.3-V V_{CCIO}. When V_{CCIO} = 5.0 V, the output driver is compatible with the ***PCI Local Bus Specification, Revision 2.1***.

Figure 17. Output Drive Characteristics



Timing Model

The continuous, high-performance FastTrack Interconnect routing resources ensure predictable performance and accurate simulation and timing analysis. This predictable performance contrasts with that of FPGAs, which use a segmented connection scheme and therefore have unpredictable performance.

Device performance can be estimated by following the signal path from a source, through the interconnect, to the destination. For example, the registered performance between two LEs on the same row can be calculated by adding the following parameters:

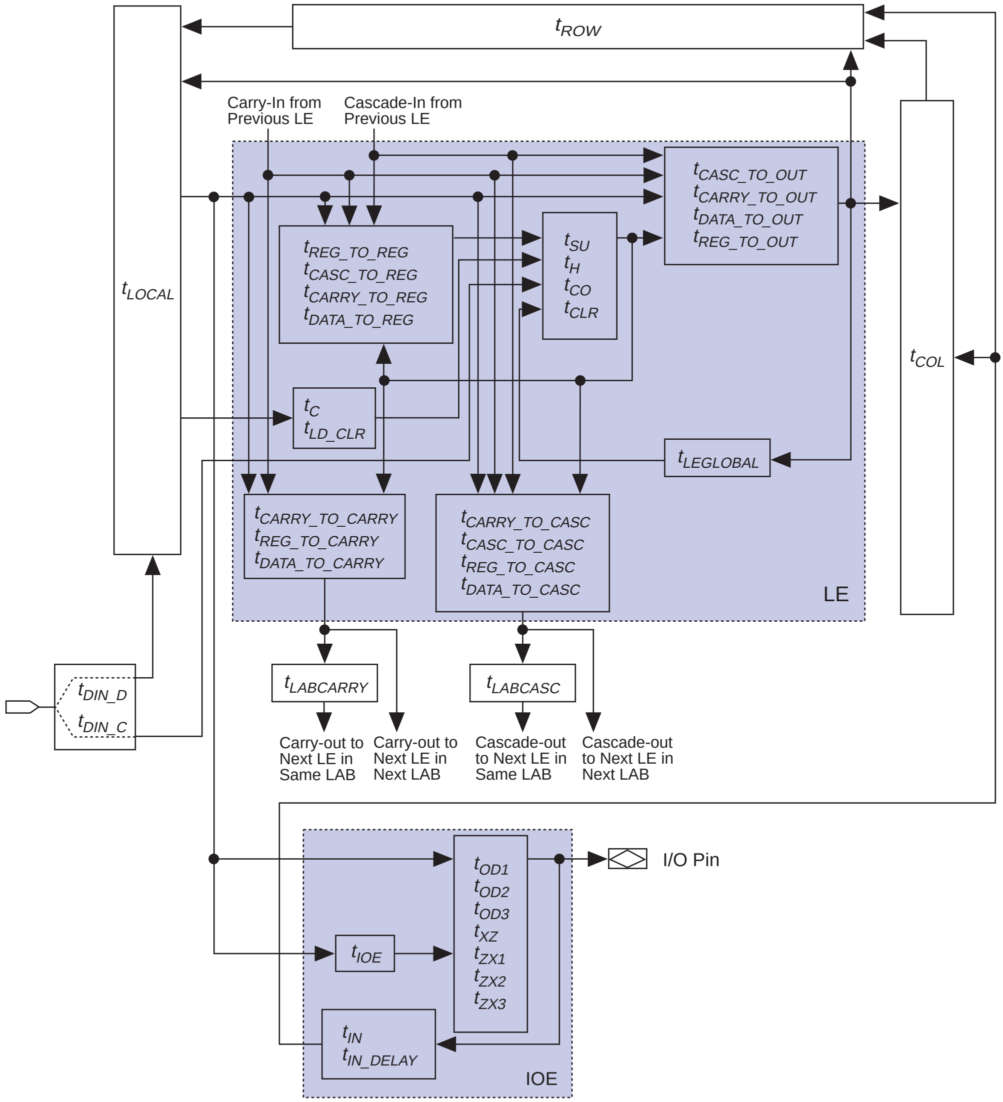
- LE register clock-to-output delay ($t_{CO} + t_{REG\_TO\_OUT}$)
- Routing delay ($t_{ROW} + t_{LOCAL}$)
- LE LUT delay ($t_{DATA\_TO\_REG}$)
- LE register setup time (t_{SU})

The routing delay depends on the placement of the source and destination LEs. A more complex registered path may involve multiple combinatorial LEs between the source and destination LEs.

Timing simulation and delay prediction are available with the MAX+PLUS II Simulator and Timing Analyzer, or with industry-standard EDA tools. The MAX+PLUS II Simulator offers both pre-synthesis functional simulation to evaluate logic design accuracy and post-synthesis timing simulation with 0.1-ns resolution. The MAX+PLUS II Timing Analyzer provides point-to-point timing delay information, setup and hold time analysis, and device-wide performance analysis.

Figure 18 shows the overall timing model, which maps the possible routing paths to and from the various elements of the FLEX 6000 device.

Figure 18. FLEX 6000 Timing Model



Tables 7 through 9 describe the FLEX 6000 internal timing microparameters, which are expressed as worst-case values. Using hand calculations, these parameters can be used to estimate design performance. However, before committing designs to silicon, actual worst-case performance should be modeled using timing simulation and timing analysis. Tables 10 and 11 describe FLEX 6000 external timing parameters.

Table 7. LE Timing Microparameters *Note (1)*

Symbol	Parameter	Conditions
$t_{REG\_TO\_REG}$	LUT delay for LE register feedback in carry chain	
$t_{CASC\_TO\_REG}$	Cascade-in to register delay	
$t_{CARRY\_TO\_REG}$	Carry-in to register delay	
$t_{DATA\_TO\_REG}$	LE input to register delay	
$t_{CASC\_TO\_OUT}$	Cascade-in to LE output delay	
$t_{CARRY\_TO\_OUT}$	Carry-in to LE output delay	
$t_{DATA\_TO\_OUT}$	LE input to LE output delay	
$t_{REG\_TO\_OUT}$	Register output to LE output delay	
t_{SU}	LE register setup time before clock; LE register recovery time after asynchronous clear	
t_{H}	LE register hold time after clock	
t_{CO}	LE register clock-to-output delay	
t_{CLR}	LE register clear delay	
t_{C}	LE register control signal delay	
t_{LD_CLR}	Synchronous load or clear delay in counter mode	
$t_{CARRY\_TO\_CARRY}$	Carry-in to carry-out delay	
$t_{REG\_TO\_CARRY}$	Register output to carry-out delay	
$t_{DATA\_TO\_CARRY}$	LE input to carry-out delay	
$t_{CARRY\_TO\_CASC}$	Carry-in to cascade-out delay	
$t_{CASC\_TO\_CASC}$	Cascade-in to cascade-out delay	
$t_{REG\_TO\_CASC}$	Register-out to cascade-out delay	
$t_{DATA\_TO\_CASC}$	LE input to cascade-out delay	
t_{CH}	LE register clock high time	
t_{CL}	LE register clock low time	

Table 8. IOE Timing Microparameters *Note (1)*

Symbol	Parameter	Conditions
t_{OD1}	Output buffer and pad delay, slow slew rate = off, V_{CCIO} = V_{CCINT}	C1 = 35 pF, *Note (2)*
t_{OD2}	Output buffer and pad delay, slow slew rate = off, V_{CCIO} = low voltage	C1 = 35 pF, *Note (3)*
t_{OD3}	Output buffer and pad delay, slow slew rate = on	C1 = 35 pF, *Note (4)*
t_{XZ}	Output buffer disable delay	C1 = 5 pF
t_{ZX1}	Output buffer enable delay, slow slew rate = off, V_{CCIO} = V_{CCINT}	C1 = 35 pF, *Note (2)*
t_{ZX2}	Output buffer enable delay, slow slew rate = off, V_{CCIO} = low voltage	C1 = 35 pF, *Note (3)*
t_{ZX3}	IOE output buffer enable delay, slow slew rate = on	C1 = 35 pF, *Note (4)*
t_{IOE}	Output enable control delay	
t_{IN}	Input pad and buffer to FastTrack Interconnect delay	
t_{IN_DELAY}	Input pad and buffer to FastTrack Interconnect delay with additional delay turned on	

Table 9. Interconnect Timing Microparameters *Note (1)*

Symbol	Parameter	Conditions
t_{LOCAL}	LAB local interconnect delay	
t_{ROW}	Row interconnect routing delay	*Note (5)*
t_{COL}	Column interconnect routing delay	*Note (5)*
t_{DIN_D}	Dedicated input to LE data delay	*Note (5)*
t_{DIN_C}	Dedicated input to LE control delay	
$t_{LEGLOBAL}$	LE output to LE control via internally-generated global signal delay	*Note (5)*
$t_{LABCARRY}$	Routing delay for the carry-out of an LE driving the carry-in signal of a different LE in a different LAB	
$t_{LABCASC}$	Routing delay for the cascade-out signal of an LE driving the cascade-in signal of a different LE in a different LAB	

Table 10. External Reference Timing Parameters		
Symbol	**Parameter**	**Conditions**
t_1	Register-to-register test pattern	Note (6)
t_{DRR}	Register-to-register delay via 4 LEs, 3 row interconnects, and 4 local interconnects	Note (7)

Table 11. External Timing Parameters		
Symbol	**Parameter**	**Conditions**
t_{INSU}	Setup time with global clock at LE register	Note (8)
t_{INH}	Hold time with global clock at LE register	Note (8)
t_{OUTCO}	Clock-to-output delay with global clock with LE register using FastFLEX™ I/O pin	Note (8)
t_{ODH}	Output data hold time after clock	Note (8)

Notes to tables:

(1) Microparameters are timing delays contributed by individual architectural elements and cannot be measured explicitly.

(2) Operating conditions:
V_{CCIO} = 5.0 V ± 5% for commercial use in FLEX 6000 devices.
V_{CCIO} = 5.0 V ± 10% for industrial use in FLEX 6000 devices.
V_{CCIO} = 3.3 V ± 10% for commercial or industrial use in FLEX 6000A devices.

(3) Operating conditions:
V_{CCIO} = 3.3 V ± 10% for commercial or industrial use in FLEX 6000 devices.
V_{CCIO} = 2.5 V ± 0.2 V for commercial or industrial use in FLEX 6000A devices.

(4) Operating conditions: V_{CCIO} = 2.5 V, 3.3 V, or 5.0 V.

(5) These parameters are worst-case values for typical applications. Post-compilation timing simulation and timing analysis are required to determine actual worst-case performance.

(6) This timing parameter shows the delay of a register-to-register test pattern and is used to determine speed grades. There are 12 LEs, including source and destination registers. The row and column interconnects between the registers vary in length.

(7) This timing parameter is shown for reference and is specified by characterization.

(8) This timing parameter is specified by characterization.

Tables 12 through 15 show the timing information for EPF6016 devices.

Table 12. LE Timing Microparameters for EPF6016 Devices (Part 1 of 2)					
Parameter	-2 Speed Grade		-3 Speed Grade		**Unit**
	Min	**Max**	**Min**	**Max**	
$t_{REG\_TO\_REG}$		2.2		2.8	ns
$t_{CASC\_TO\_REG}$		0.9		1.2	ns
$t_{CARRY\_TO\_REG}$		1.6		2.1	ns
$t_{DATA\_TO\_REG}$		2.4		3.0	ns

Table 12. LE Timing Microparameters for EPF6016 Devices (Part 2 of 2)

Parameter	-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	
$t_{CASC\_TO\_OUT}$		1.3		1.7	ns
$t_{CARRY\_TO\_OUT}$		2.4		3.0	ns
$t_{DATA\_TO\_OUT}$		2.7		3.4	ns
$t_{REG\_TO\_OUT}$		0.3		0.5	ns
t_{SU}	1.1		1.6		ns
t_H	1.8		2.3		ns
t_{CO}		0.3		0.4	ns
t_{CLR}		0.5		0.6	ns
t_C		1.2		1.5	ns
t_{LD_CLR}		1.2		1.5	ns
$t_{CARRY\_TO\_CARRY}$		0.2		0.4	ns
$t_{REG\_TO\_CARRY}$		0.8		1.1	ns
$t_{DATA\_TO\_CARRY}$		1.7		2.2	ns
$t_{CARRY\_TO\_CASC}$		1.7		2.2	ns
$t_{CASC\_TO\_CASC}$		0.9		1.2	ns
$t_{REG\_TO\_CASC}$		1.6		2.0	ns
$t_{DATA\_TO\_CASC}$		1.7		2.1	ns
t_{CH}	4.0		4.0		ns
t_{CL}	4.0		4.0		ns
t_{OD1}		2.3		2.8	ns
t_{OD2}		4.6		5.1	ns
t_{OD3}		4.7		5.2	ns
t_{XZ}		2.3		2.8	ns
t_{ZX1}		2.3		2.8	ns
t_{ZX2}		4.6		5.1	ns
t_{ZX3}		4.7		5.2	ns
t_{IOE}		0.5		0.6	ns
t_{IN}		3.3		4.0	ns
t_{IN_DELAY}		4.6		5.6	ns

Table 13. Interconnect Timing Microparameters for EPF6016 Devices

Parameter	-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	
t_{LOCAL}		0.8		1.0	ns
t_{ROW}		2.9		3.3	ns
t_{COL}		2.3		2.5	ns
t_{DIN_D}		4.9		6.0	ns
t_{DIN_C}		4.8		6.0	ns
$t_{LEGLOBAL}$		3.1		3.9	ns
$t_{LABCARRY}$		0.4		0.5	ns
$t_{LABCASC}$		0.8		1.0	ns

Table 14. External Reference Timing Parameters for EPF6016 Devices

Parameter	-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	
t_1		53.0		65.0	ns
t_{DRR}		16.0		20.0	ns

Table 15. External Timing Parameters for EPF6016 Devices

Parameter	-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	
t_{INSU}	3.2		4.1		ns
t_{INH}	0.0		0.0		ns
t_{OUTCO}		7.9		9.9	ns
t_{ODH}	2.0		2.0		ns

Table 16. External Reference Timing Parameters for EPF6024A Devices (1)

Parameter	-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	
t_1		53.0		65.0	ns

Note:
(1) This information is preliminary.

Power Consumption

The supply power (P) for FLEX 6000 devices can be calculated with the following equations:

$$P = P_{INT} + P_{IO}$$
$$P = (I_{CCSTANDBY} + I_{CCACTIVE}) \times V_{CC} + P_{IO}$$

Typical $I_{CCSTANDBY}$ values are shown as I_{CC0} in the "FLEX 6000 Device DC Operating Conditions" table on pages 225 and 227 of this data sheet. The $I_{CCACTIVE}$ value depends on the switching frequency and the application logic. This value is based on the amount of current that each LE typically consumes. The P_{IO} value, which depends on the device output load characteristics and switching frequency, can be calculated using the guidelines given in *Application Note 74 (Evaluating Power for Altera Devices)* in this data book.

The $I_{CCACTIVE}$ value can be calculated with the following equation:

$$I_{CCACTIVE} = K \times \mathbf{f_{MAX}} \times N \times \mathbf{tog_{LC}} \times \frac{\mu A}{MHz \times LE}$$

Where:

$\mathbf{f_{MAX}}$ = Maximum operating frequency in MHz
N = Total number of logic cells used in a FLEX 6000 device
$\mathbf{tog_{LC}}$ = Average percentage of logic cells toggling at each clock (typically 12.5%)
K = Constant

The K value for an EPF6016 device is 88, and the K value for an EPF6024A device is 55.

This calculation provides an I_{CC} estimate based on typical conditions with no output load. The actual I_{CC} should be verified during operation because this measurement is sensitive to the actual pattern in the device and the environmental operating conditions.

In order to better reflect actual designs, the power model (and the constant K in the power calculation equations shown above) for continuous interconnect FLEX devices assumes that logic cells drive FastTrack Interconnect channels. In contrast, the power model of segmented FPGAs assumes that all logic cells drive only one short interconnect segment. This assumption may lead to inaccurate results, compared to measured power consumption for an actual design in a segmented interconnect FPGA.

Figures 19 and 20 show the relationship between the current and operating frequency for EPF6016 and EPF6024A devices.

Figure 19. $I_{CCACTIVE}$ vs. Operating Frequency for EPF6016 Devices

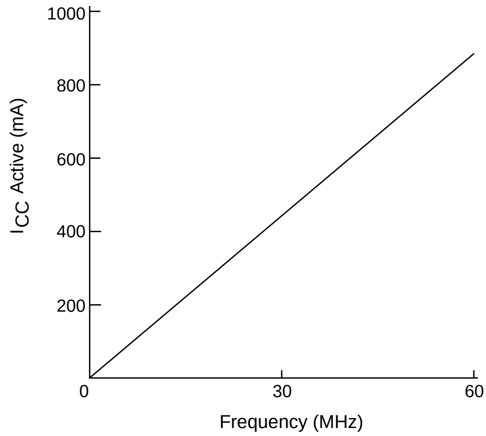
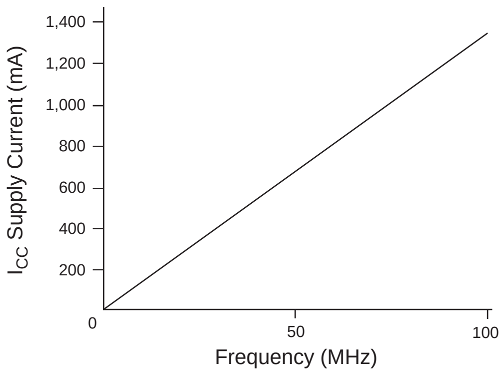


Figure 20. $I_{CCACTIVE}$ vs. Operating Frequency for EPF6024A Devices



Device Configuration & Operation

The FLEX 6000 architecture supports several configuration schemes to load a design into the device(s) on the circuit board. This section summarizes the device operating modes and available device configuration schemes.

Go to *Application Note 87 (Configuring FLEX 6000 Devices)* for detailed information on configuring FLEX 6000 devices, including sample schematics, timing diagrams, and configuration options, pins and parameters.

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 EPROM 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 17 shows the data sources for each configuration scheme.

Table 17. Configuration Schemes

Configuration Scheme	Data Source
Configuration EPROM	EPC1 or EPC1441 Configuration EPROM
Passive serial (PS)	Serial data source
Passive serial asynchronous (PSA)	Serial data source

Device Pin-Outs

Table 18 shows the pin names and numbers for FLEX 6000 device packages.

Table 18. FLEX 6000 Device Pin-Outs (Part 1 of 2) Notes (1), (2)

Pin Name	144-Pin TQFP EPF6016 EPF6016A	208-Pin PQFP EPF6016 EPF6016A EPF6024A	240-Pin PQFP EPF6016 EPF6024A	256-Pin BGA EPF6016	256-Pin BGA EPF6024A
MSEL *(3)*	33	46	52	T3	T3
nSTATUS *(3)*	56	80	92	W11	W11
nCONFIG *(3)*	53	77	89	Y11	Y11
DCLK *(3)*	128	184	212	C10	C10
CONF_DONE *(3)*	105	150	172	E18	E18
INIT_DONE *(4)*	94	135	155	J19	J19
nCE *(3)*	4	6	9	E1	E1
nCEO *(5)*	70	102	117	V18	V18
nWS *(5)*	117	169	195	B15	B15
nRS *(5)*	120	174	200	C13	C13
nCS *(5)*	111	159	184	B17	B17
CS *(5)*	114	162	188	A17	A17
RDYnBSY *(5)*	97	140	161	G20	G20
CLKUSR *(5)*	100	144	166	G17	G17
DATA *(3)*, *(6)*	125	181	209	B10	B10
TDI *(7)*	13	19	22	J3	J3
TDO *(7)*	73	107	124	T17	T17
TCK *(7)*, *(8)*	34	47	54	V1	V1
TMS *(7)*	27	38	44	P3	P3
Dedicated Inputs	17, 20, 89, 92	24, 28, 128, 132	28, 32, 148, 152	K19, L1, L3, L20	K19, L1, L3, L20
DEV_CLRn *(4)*	130	187	216	C9	C9
DEV_OE *(4)*	123	178	205	A12	A12
VCCINT	6, 31, 77, 103	8, 26, 44, 111, 130, 148	11, 30, 50, 130, 150, 170	D20, F3, K20, L2, T20, U1	D20, F3, K20, L2, T20, U1
VCCIO	7, 19, 32, 55, 78, 91, 104, 127	9, 27, 45, 63, 79, 96, 112, 131, 149, 166, 183, 200	12, 31, 51, 72, 91, 110, 131, 151, 171, 192, 211, 230	D6, D11, D15, F4, F17, K4, L17, R4, R17, U6, U10, U15	D6, D11, D15, F4, F17, K4, L17, R4, R17, U6, U10, U15

Table 18. FLEX 6000 Device Pin-Outs (Part 2 of 2) Notes (1), (2)

Pin Name	144-Pin TQFP EPF6016 EPF6016A	208-Pin PQFP EPF6016 EPF6016A EPF6024A	240-Pin PQFP EPF6016 EPF6024A	256-Pin BGA EPF6016	256-Pin BGA EPF6024A
GND	5, 18, 30, 54, 76, 90, 102, 126	7, 25, 43, 62, 78, 95, 110, 129, 147, 165, 182, 199	10, 29, 49, 61, 71, 90, 109, 120, 129, 149, 169, 181, 191, 210, 229, 240	A1, D4, D8, D13, D17, H4, H17, N4, N17, U4, U8, U13, U17	A1, D4, D8, D13, D17, H4, H17, N4, N17, U4, U8, U13, U17
No Connect (N.C.) *(9)*	—	—	—	A11, A16, B4, C7, D12, E20, J20, T2, U12, V8, V14, W5, Y17, Y19	—
Total user I/O pins *(10)*	117	171	199	204	218

Notes:
(1) All pins not listed are user I/O pins.
(2) Pin-out information for FLEX 6000A devices and 256-pin BGA packages are preliminary.
(3) This pin is a dedicated configuration pin; it is not available as a user I/O pin.
(4) This pin can be used as a user I/O pin if not used for its chip-wide or configuration function.
(5) This pin can be used as a user I/O pin after configuration.
(6) This pin is tri-stated in user mode.
(7) If the device is not configured to use the JTAG BST circuitry, this pin is available as a user I/O pin. If the JTAG device option is not set, JTAG testing may still be performed before configuration.
(8) If this pin is used as an input in user mode, ensure that it does not toggle before or during configuration.
(9) To maintain pin compatibility when migrating from a EPF6024ABC256 device to a EPF6016BC256 device, do not use these pins as user I/O pins.
(10) The user I/O count includes dedicated input and I/O pins.

Revision History

The information contained in the *FLEX 6000 Programmable Logic Device Family Data Sheet*, version 3.01 supersedes information published in the *FLEX 6000 Programmable Logic Device Family Data Sheet*, version 3.0, which can be found in the ***1998 Data Book***.

The *FLEX 6000 Programmable Logic Device Family Data Sheet*, version 3.01 contains the following changes:

- In the "Timing Model" section, external reference timing parameters for EPF6024A devices have been added.
- In the "Power Consumption" section, a figure representing the relationship between $I_{CCACTIVE}$ and operating frequency for EPF6024A devices has been added as well as the K value for EPF6024A devices.