



Welcome to [E-XFL.COM](#)

Understanding [Embedded - FPGAs \(Field Programmable Gate Array\)](#)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details

Product Status	Obsolete
Number of LABs/CLBs	132
Number of Logic Elements/Cells	1320
Total RAM Bits	-
Number of I/O	171
Number of Gates	16000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	208-BFQFP
Supplier Device Package	208-PQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/intel/epf6016aqc208-1n

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.

Table 4. FLEX 6000 Device Performance for Complex Designs <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.

Functional Description

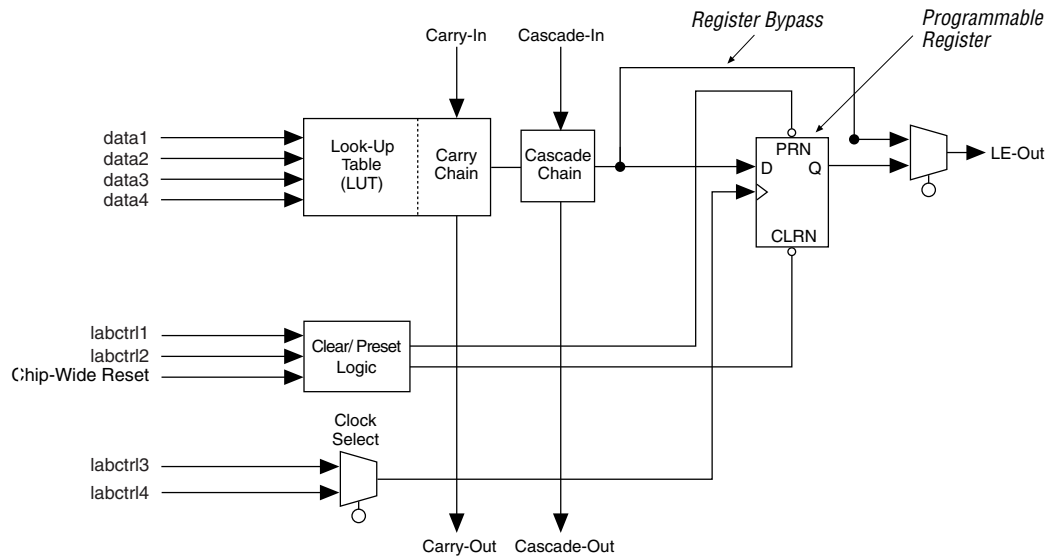
The FLEX 6000 OptiFLEX architecture consists of logic elements (LEs). Each LE includes a 4-input look-up table (LUT), which can implement any 4-input function, a register, and dedicated paths for carry and cascade chain functions. Because each LE contains a register, a design can be easily pipelined 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 Altera 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 17](#) 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 as low as 2 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 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.

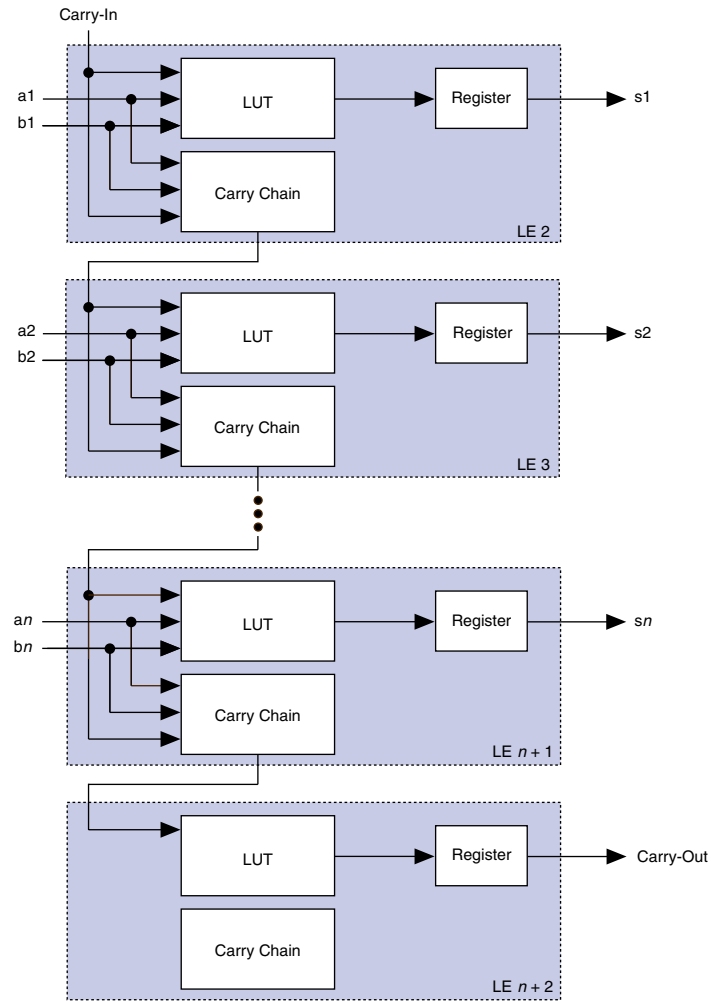
Carry Chain

The carry chain provides a very fast (0.1 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.

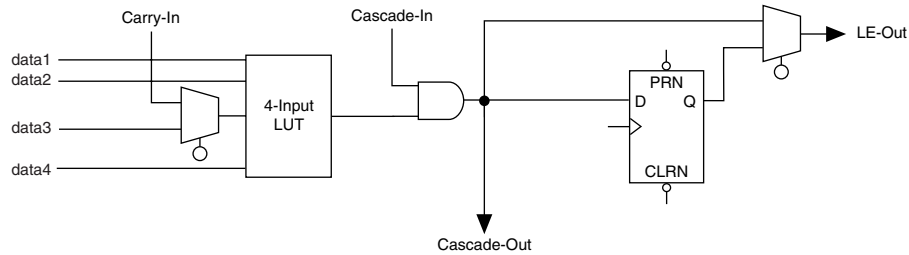
A cascade chain implementing an AND gate can use the register in the last LE; a cascade chain implementing an OR gate cannot use this register because of the inversion required to implement the OR gate.

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.

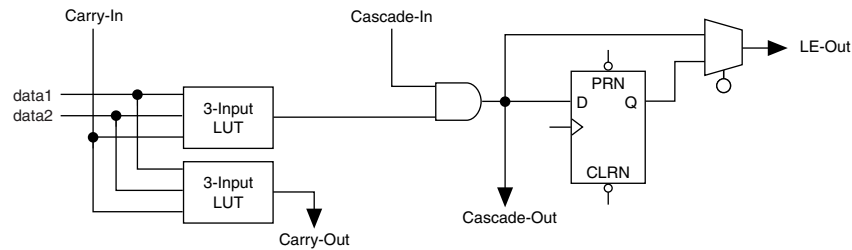
Figure 6 shows how the cascade function can connect adjacent LEs to form functions with a wide fan-in. In this example, functions of $4n$ variables are implemented with n LEs. The cascade chain requires 3.4 ns to decode a 16-bit address.

Figure 7. LE Operating Modes

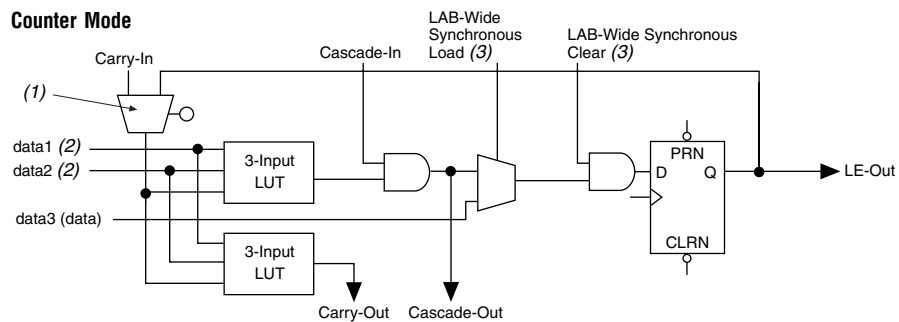
Normal Mode



Arithmetic Mode

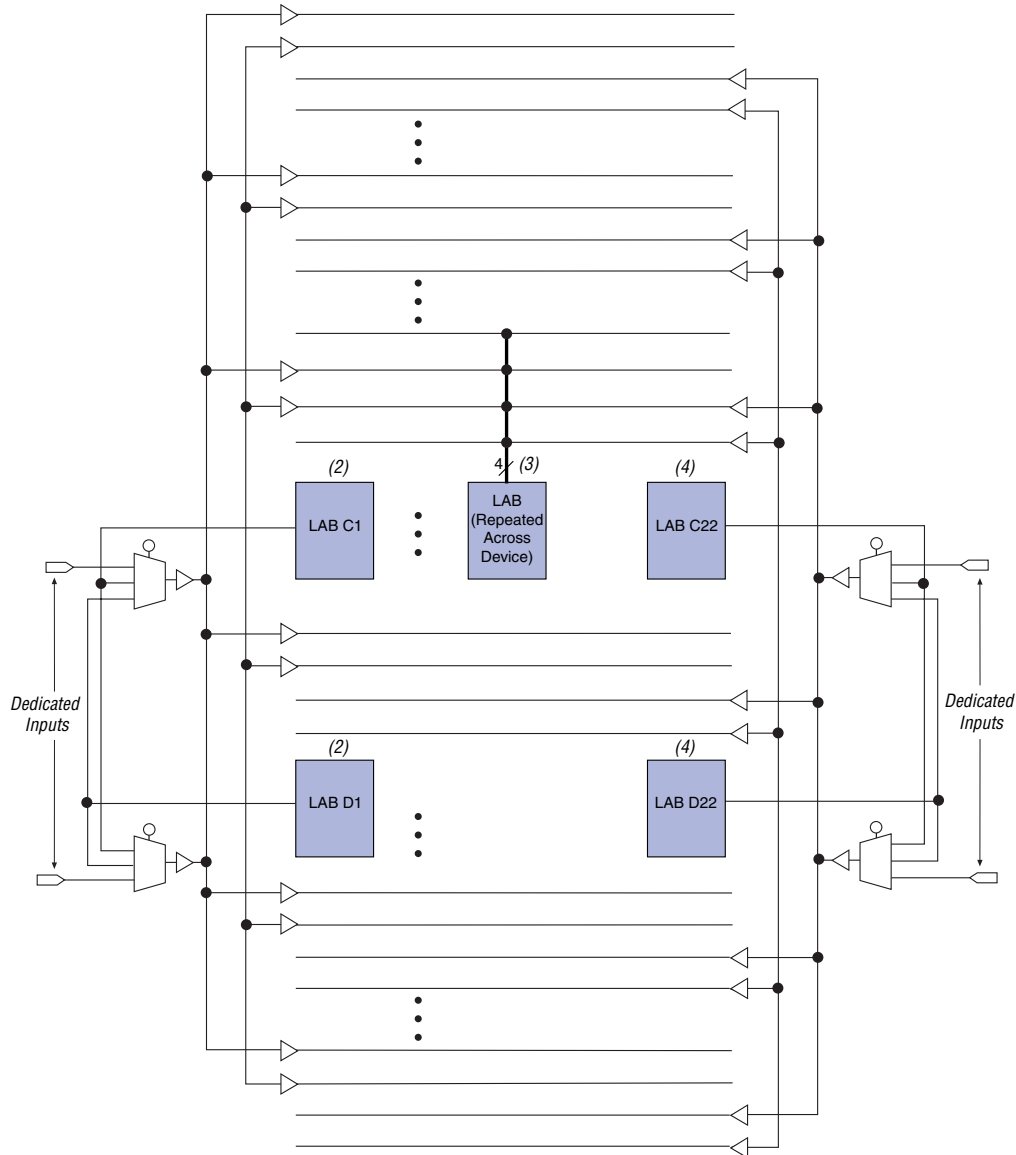


Counter Mode



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.

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 EPF6010A devices, LABs in rows B and C drive global signals. 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 signals on the right side.

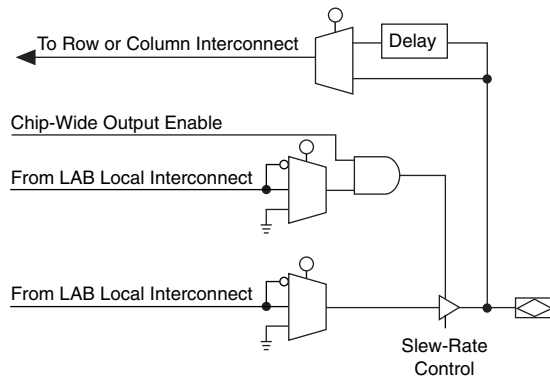
I/O Elements

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). 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

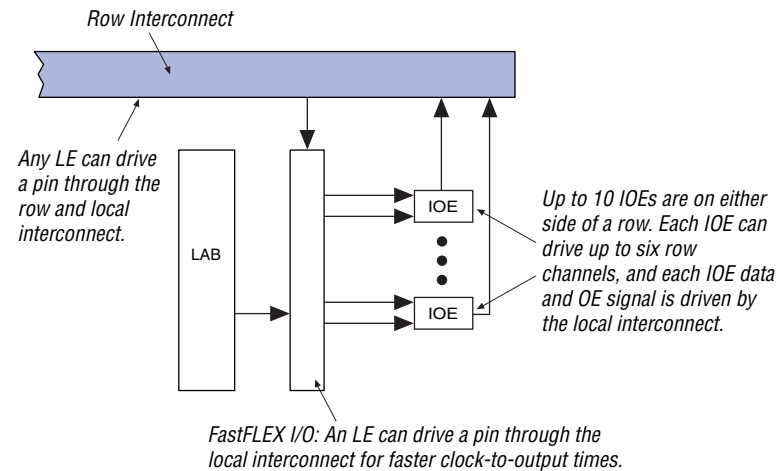


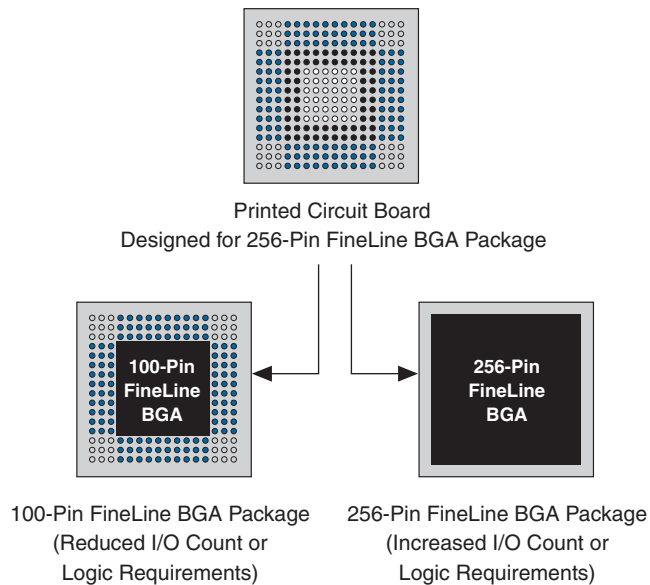
Figure 15. SameFrame Pin-Out Example

Table 6 lists the 3.3-V FLEX 6000 devices with the SameFrame pin-out feature.

Table 6. 3.3-V FLEX 6000 Devices with SameFrame Pin-Outs

Device	100-Pin FineLine BGA	256-Pin FineLine BGA
EPF6016A	V	V
EPF6024A		V

Output Configuration

This section discusses slew-rate control, the MultiVolt I/O interface, power sequencing, and hot-socketing 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 6.8 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.

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

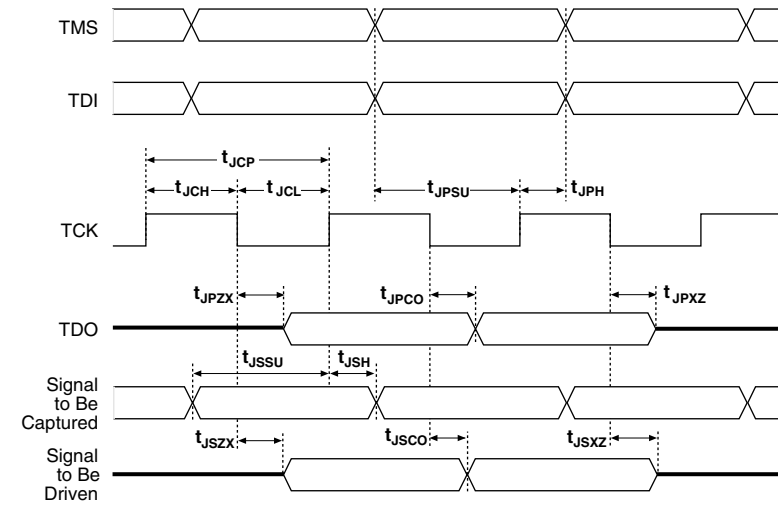
Table 9. FLEX 6000 Device Boundary-Scan Register Length	
Device	Boundary-Scan Register Length
EPF6010A	522
EPF6016	621
EPF6016A	522
EPF6024A	666

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.

Table 17. FLEX 6000 3.3-V Device DC Operating Conditions <i>Notes (5), (6)</i>						
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	High-level input voltage		1.7		5.75	V
V_{IL}	Low-level input voltage		-0.5		0.8	V
V_{OH}	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
	2.5-V high-level output voltage	$I_{OH} = -100$ μ A DC, $V_{CCIO} = 2.30$ V (7)	2.1			V
		$I_{OH} = -1$ mA DC, $V_{CCIO} = 2.30$ V (7)	2.0			V
		$I_{OH} = -2$ mA DC, $V_{CCIO} = 2.30$ V (7)	1.7			V
V_{OL}	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
	2.5-V low-level output voltage	$I_{OL} = 100$ μ A DC, $V_{CCIO} = 2.30$ V (8)			0.2	V
		$I_{OL} = 1$ mA DC, $V_{CCIO} = 2.30$ V (8)			0.4	V
		$I_{OL} = 2$ mA DC, $V_{CCIO} = 2.30$ V (8)			0.7	V
I_I	Input pin leakage current	$V_I = 5.3$ V to ground (8)	-10		10	μ A
I_{OZ}	Tri-stated I/O pin leakage current	$V_O = 5.3$ V to ground (8)	-10		10	μ A
I_{CC0}	V_{CC} supply current (standby)	$V_I =$ ground, no load		0.5	5	mA

Table 18. FLEX 6000 3.3-V Device Capacitance <i>Note (9)</i>					
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) The minimum DC input voltage is -0.5 V. During transitions, the inputs may undershoot to -2.0 V or overshoot to 5.75 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 is 100 ms. V_{CC} must rise monotonically.
- (5) Typical values are for $T_A = 25^\circ$ C and $V_{CC} = 3.3$ V.
- (6) These values are specified under [Table 16 on page 33](#).
- (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.

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

Figure 19 shows the overall timing model, which maps the possible routing paths to and from the various elements of the FLEX 6000 device.

Table 30. IOE Timing Microparameters for EPF6016 Devices

Table 30. IOE Timing Microparameters for EPF6016 Devices					
Parameter	Speed Grade				Unit
	-2		-3		
	Min	Max	Min	Max	
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 31. Interconnect Timing Microparameters for EPF6016 Devices

Table 31. Interconnect Timing Microparameters for EPF6016 Devices					
Parameter	Speed Grade				Unit
	-2		-3		
	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 32. External Reference Timing Parameters for EPF6016 Devices

Table 32. External Reference Timing Parameters for EPF6016 Devices					
Parameter	Speed Grade				Unit
	-2		-3		
	Min	Max	Min	Max	
t ₁		53.0		65.0	ns
t _{DDR}		16.0		20.0	ns

Table 33. External Timing Parameters for EPF6016 Devices

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

Tables 34 through 38 show the timing information for EPF6024A devices.

Table 34. LE Timing Microparameters for EPF6024A Devices

Parameter	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{REG_TO_REG}$		1.2		1.3		1.6	ns
$t_{CASC_TO_REG}$		0.7		0.8		1.0	ns
$t_{CARRY_TO_REG}$		1.6		1.8		2.2	ns
$t_{DATA_TO_REG}$		1.3		1.4		1.7	ns
$t_{CASC_TO_OUT}$		1.2		1.3		1.6	ns
$t_{CARRY_TO_OUT}$		2.0		2.2		2.6	ns
$t_{DATA_TO_OUT}$		1.8		2.1		2.6	ns
$t_{REG_TO_OUT}$		0.3		0.3		0.4	ns
t_{SU}	0.9		1.0		1.2		ns
t_H	1.3		1.4		1.7		ns
t_{CO}		0.2		0.3		0.3	ns
t_{CLR}		0.3		0.3		0.4	ns
t_C		1.9		2.1		2.5	ns
t_{LD_CLR}		1.9		2.1		2.5	ns
$t_{CARRY_TO_CARRY}$		0.2		0.2		0.3	ns
$t_{REG_TO_CARRY}$		1.4		1.6		1.9	ns
$t_{DATA_TO_CARRY}$		1.3		1.4		1.7	ns
$t_{CARRY_TO_CASC}$		1.1		1.2		1.4	ns
$t_{CASC_TO_CASC}$		0.7		0.8		1.0	ns
$t_{REG_TO_CASC}$		1.4		1.6		1.9	ns
$t_{DATA_TO_CASC}$		1.0		1.1		1.3	ns
t_{CH}	2.5		3.0		3.5		ns
t_{CL}	2.5		3.0		3.5		ns

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.

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 LEs drive FastTrack Interconnect channels. In contrast, the power model of segmented FPGAs assumes that all LEs 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.

Figure 20 shows the relationship between the current and operating frequency for EPF6010A, EPF6016, EPF6016A, and EPF6024A devices.

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.

Table 40. Configuration Schemes	
Configuration Scheme	Data Source
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



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
(888) 3-ALTERA
lit_req@altera.com

Altera, BitBlaster, ByteBlasterMV, FastFlex, FastTrack, FineLine BGA, FLEX, MasterBlaster, MAX+PLUS II, MegaCore, MultiVolt, OptiFLEX, Quartus, SameFrame, and specific device designations are trademarks and/or service marks of Altera Corporation in the United States and other countries. Altera acknowledges the trademarks of other organizations for their respective products or services mentioned in this document, specifically: Verilog is a registered trademark of and Verilog-XL is a trademarks of Cadence Design Systems, Inc. DATA I/O is a registered trademark of Data I/O Corporation. HP is a registered trademark of Hewlett-Packard Company. Exemplar Logic is a registered trademark of Exemplar Logic, Inc. Pentium is a registered trademark of Intel Corporation. Mentor Graphics is a registered trademark of Mentor Graphics Corporation. OrCAD is a registered trademark of OrCAD Systems, Corporation. SPARCstation is a registered trademark of SPARC International, Inc. and is licensed exclusively to Sun Microsystems, Inc. Sun Workstation is a registered trademark of, and Sun is a registered trademark of Sun Microsystems, Inc. Synopsys is a registered trademark and DesignTime, HDL Compiler, and DesignWare are trademarks of Synopsys, Inc. VeriBest is a registered trademark of Viewlogic Systems, Inc. Viewlogic is a registered trademark of Viewlogic Systems, Inc. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Copyright © 2001 Altera Corporation. All rights reserved.



I.S. EN ISO 9001

