



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

### Applications of Embedded - FPGAs

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

#### Details

Product Status	Obsolete
Number of LABs/CLBs	72
Number of Logic Elements/Cells	576
Total RAM Bits	6144
Number of I/O	134
Number of Gates	31000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	208-BFQFP
Supplier Device Package	208-PQFP (28x28)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/intel/epf10k10aqc208-2">https://www.e-xfl.com/product-detail/intel/epf10k10aqc208-2</a>

**Notes to tables:**

- (1) FLEX 10K and FLEX 10KA device package types include plastic J-lead chip carrier (PLCC), thin quad flat pack (TQFP), plastic quad flat pack (PQFP), power quad flat pack (RQFP), ball-grid array (BGA), pin-grid array (PGA), and FineLine BGA™ packages.
- (2) This option is supported with a 256-pin FineLine BGA package. By using SameFrame pin migration, all FineLine BGA packages are pin compatible. For example, a board can be designed to support both 256-pin and 484-pin FineLine BGA packages. The Altera software automatically avoids conflicting pins when future migration is set.

## General Description

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

FLEX 10K devices are reconfigurable, which allows 100% testing prior to shipment. As a result, the designer is not required to generate test vectors for fault coverage purposes. Additionally, the designer does not need to manage inventories of different ASIC designs; FLEX 10K devices can be configured on the board for the specific functionality required.

Table 6 shows FLEX 10K performance for some common designs. All performance values were obtained with Synopsys DesignWare or LPM functions. No special design technique was required to implement the applications; the designer simply inferred or instantiated a function in a Verilog HDL, VHDL, Altera Hardware Description Language (AHDL), or schematic design file.

**Table 6. FLEX 10K & FLEX 10KA Performance**

Application	Resources Used		Performance				Units
	LEs	EABs	-1 Speed Grade	-2 Speed Grade	-3 Speed Grade	-4 Speed Grade	
16-bit loadable counter (1)	16	0	204	166	125	95	MHz
16-bit accumulator (1)	16	0	204	166	125	95	MHz
16-to-1 multiplexer (2)	10	0	4.2	5.8	6.0	7.0	ns
256 × 8 RAM read cycle speed (3)	0	1	172	145	108	84	MHz
256 × 8 RAM write cycle speed (3)	0	1	106	89	68	63	MHz

**Notes:**

- (1) The speed grade of this application is limited because of clock high and low specifications.
- (2) This application uses combinatorial inputs and outputs.
- (3) This application uses registered inputs and outputs.



For more information, see the following documents:

- *Configuration Devices for APEX & FLEX Devices Data Sheet*
- *BitBlaster Serial Download Cable Data Sheet*
- *ByteBlasterMV Parallel Port Download Cable Data Sheet*
- *Application Note 116 (Configuring APEX 20K, FLEX 10K & FLEX 6000 Devices)*

FLEX 10K devices are supported by Altera development systems; single, integrated packages that offer 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 10K architecture.

The Altera development systems run on Windows-based PCs and Sun SPARCstation, and HP 9000 Series 700/800 workstations.



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

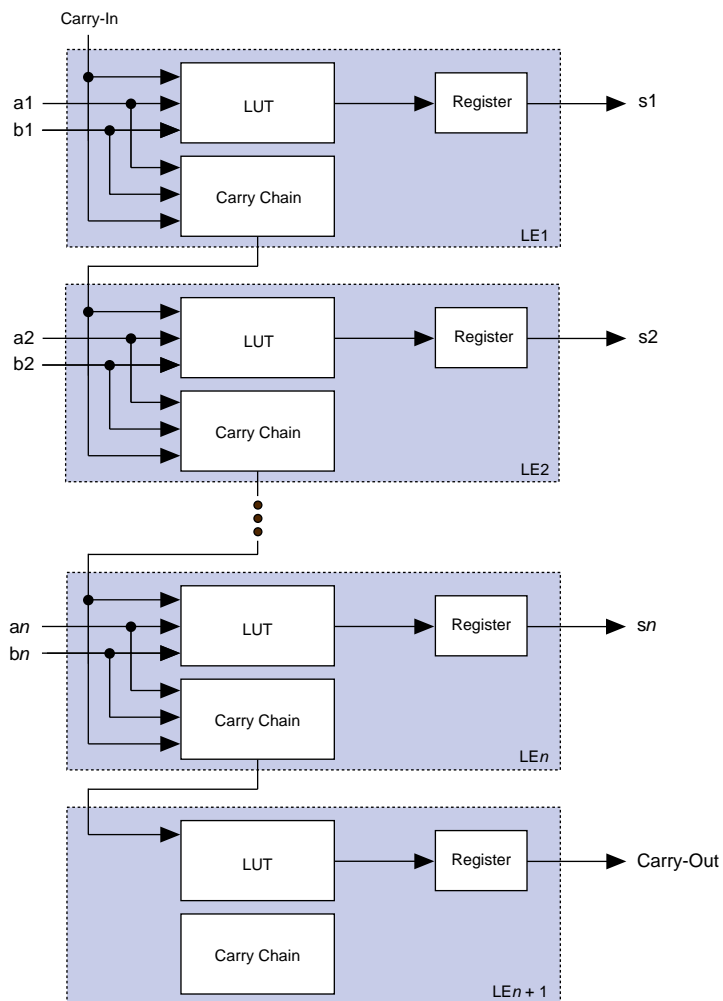
## Functional Description

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

The embedded array consists of a series of EABs. When implementing memory functions, each EAB provides 2,048 bits, which can be used to create RAM, ROM, dual-port RAM, or first-in first-out (FIFO) functions. When implementing logic, each EAB can contribute 100 to 600 gates towards complex logic functions, such as multipliers, microcontrollers, state machines, and DSP functions. EABs can be used independently, or multiple EABs can be combined to implement larger functions.

Figure 7 shows how an  $n$ -bit full adder can be implemented in  $n + 1$  LEs with the carry chain. One portion of the LUT generates the sum of two bits using the input signals and the carry-in signal; the sum is routed to the output of the LE. The register can either be bypassed for simple adders or be used for an accumulator function. The carry chain logic generates the carry-out signal, which is routed directly to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it can be used as a general-purpose signal.

**Figure 7. Carry Chain Operation ( $n$ -bit Full Adder)**



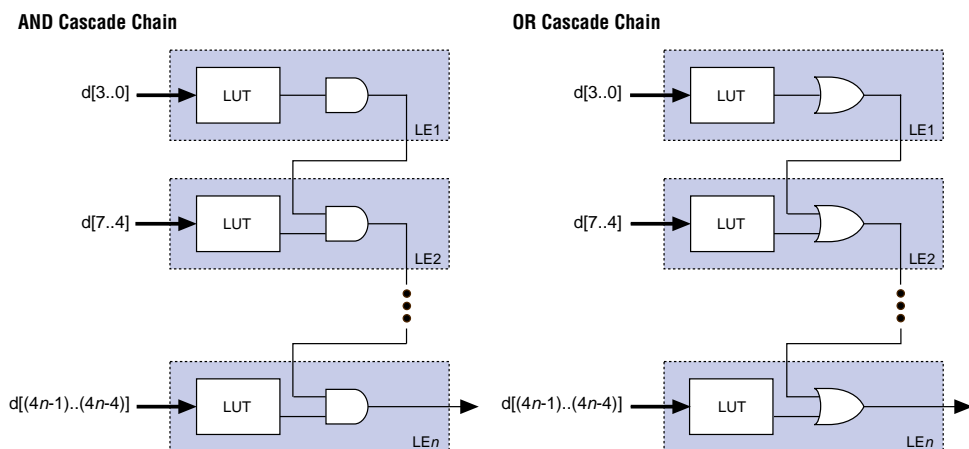
### Cascade Chain

With the cascade chain, the FLEX 10K architecture can implement functions that have a very wide fan-in. Adjacent LUTs can be used to compute portions of the function in parallel; the cascade chain serially connects the intermediate values. The cascade chain can use a logical AND or logical OR (via De Morgan's inversion) to connect the outputs of adjacent LEs. Each additional LE provides four more inputs to the effective width of a function, with a delay as low as 0.7 ns per LE. Cascade chain logic can be created automatically by the Compiler during design processing, or manually by the designer during design entry.

Cascade chains longer than eight bits are implemented automatically by linking several LABs together. For easier routing, a long cascade chain skips every other LAB in a row. A cascade chain longer than one LAB skips either from even-numbered LAB to even-numbered LAB, or from odd-numbered LAB to odd-numbered LAB (e.g., the last LE of the first LAB in a row cascades to the first LE of the third LAB). The cascade chain does not cross the center of the row (e.g., in the EPF10K50 device, the cascade chain stops at the eighteenth LAB and a new one begins at the nineteenth LAB). This break is due to the EAB's placement in the middle of the row.

Figure 8 shows how the cascade function can connect adjacent LEs to form functions with a wide fan-in. These examples show functions of  $4n$  variables implemented with  $n$  LEs. The LE delay is as low as 1.6 ns; the cascade chain delay is as low as 0.7 ns. With the cascade chain, 3.7 ns is needed to decode a 16-bit address.

**Figure 8. Cascade Chain Operation**



### Normal Mode

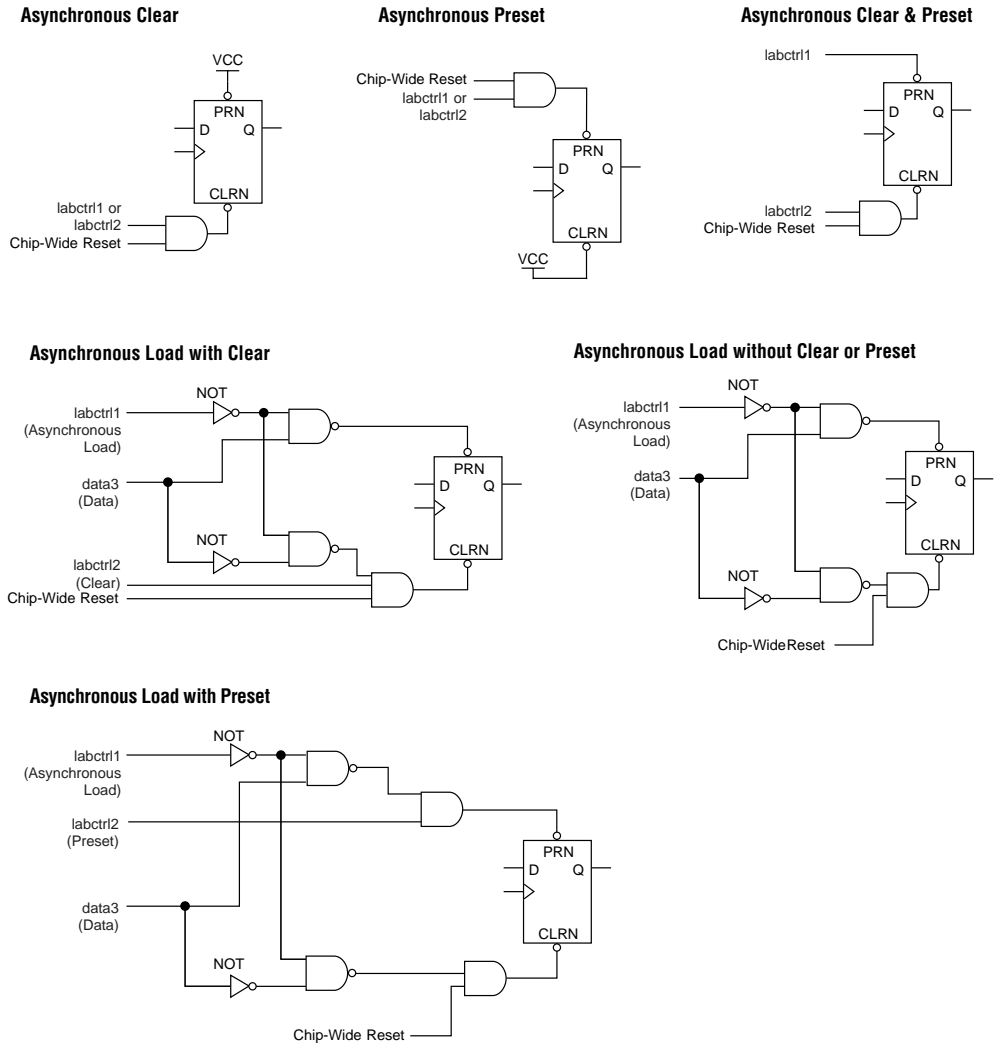
The normal mode is suitable for general logic applications and wide decoding functions that can take advantage of a cascade chain. In normal mode, four data inputs from the LAB local interconnect and the carry-in are inputs to a four-input LUT. The Compiler automatically selects the carry-in or the DATA3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

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

### Arithmetic Mode

The arithmetic mode offers 2 three-input LUTs that are ideal for implementing adders, accumulators, and comparators. One LUT computes a three-input function, and the other generates a carry output. As shown in [Figure 9](#) on page 19, the first LUT uses the carry-in signal and two data inputs from the LAB local interconnect to generate a combinatorial or registered output. For example, in an adder, this output is the sum of three signals: a, b, and carry-in. The second LUT uses the same three signals to generate a carry-out signal, thereby creating a carry chain. The arithmetic mode also supports simultaneous use of the cascade chain.

**Figure 10. LE Clear & Preset Modes**



### Asynchronous Clear

The flipflop can be cleared by either LABCTRL1 or LABCTRL2. In this mode, the preset signal is tied to  $V_{CC}$  to deactivate it.

### **Asynchronous Preset**

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

### **Asynchronous Preset & Clear**

When implementing asynchronous clear and preset, LABCTRL1 controls the preset and LABCTRL2 controls the clear. DATA3 is tied to  $V_{CC}$ , therefore, asserting LABCTRL1 asynchronously loads a one into the register, effectively presetting the register. Asserting LABCTRL2 clears the register.

### **Asynchronous Load with Clear**

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

### **Asynchronous Load with Preset**

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

### **Asynchronous Load without Preset or Clear**

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



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

<b>Table 10. FLEX 10K Row-to-IOE Interconnect Resources</b>		
<b>Device</b>	<b>Channels per Row (<i>n</i>)</b>	<b>Row Channels per Pin (<i>m</i>)</b>
EPF10K10 EPF10K10A	144	18
EPF10K20	144	18
EPF10K30 EPF10K30A	216	27
EPF10K40	216	27
EPF10K50 EPF10K50V	216	27
EPF10K70	312	39
EPF10K100 EPF10K100A	312	39
EPF10K130V	312	39
EPF10K250A	456	57

#### *Column-to-IOE Connections*

When an IOE is used as an input, it can drive up to two separate column channels. When an IOE is used as an output, the signal is driven by a multiplexer that selects a signal from the column channels. Two IOEs connect to each side of the column channels. Each IOE can be driven by column channels via a multiplexer. The set of column channels that each IOE can access is different for each IOE. See Figure 15.

**Table 27. FLEX 10KA 3.3-V Device Recommended Operating Conditions**

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{CCINT}$	Supply voltage for internal logic and input buffers	(3), (4)	3.00 (3.00)	3.60 (3.60)	V
$V_{CCIO}$	Supply voltage for output buffers, 3.3-V operation	(3), (4)	3.00 (3.00)	3.60 (3.60)	V
	Supply voltage for output buffers, 2.5-V operation	(3), (4)	2.30 (2.30)	2.70 (2.70)	V
$V_I$	Input voltage	(5)	−0.5	5.75	V
$V_O$	Output voltage		0	$V_{CCIO}$	V
$T_A$	Ambient temperature	For commercial use	0	70	° C
		For industrial use	−40	85	° C
$T_J$	Operating temperature	For commercial use	0	85	° C
		For industrial use	−40	100	° C
$t_R$	Input rise time			40	ns
$t_F$	Input fall time			40	ns

**Table 35. EAB Timing Macroparameters** *Notes (1), (6)*

Symbol	Parameter	Conditions
$t_{EABAA}$	EAB address access delay	
$t_{EABRCCOMB}$	EAB asynchronous read cycle time	
$t_{EABRCREG}$	EAB synchronous read cycle time	
$t_{EABWP}$	EAB write pulse width	
$t_{EABWCCOMB}$	EAB asynchronous write cycle time	
$t_{EABWCREG}$	EAB synchronous write cycle time	
$t_{EABDD}$	EAB data-in to data-out valid delay	
$t_{EABDATACO}$	EAB clock-to-output delay when using output registers	
$t_{EABDATASU}$	EAB data/address setup time before clock when using input register	
$t_{EABDATAH}$	EAB data/address hold time after clock when using input register	
$t_{EABWESU}$	EAB $\overline{WE}$ setup time before clock when using input register	
$t_{EABWEH}$	EAB $\overline{WE}$ hold time after clock when using input register	
$t_{EABWDSU}$	EAB data setup time before falling edge of write pulse when not using input registers	
$t_{EABWDH}$	EAB data hold time after falling edge of write pulse when not using input registers	
$t_{EABWASU}$	EAB address setup time before rising edge of write pulse when not using input registers	
$t_{EABWAH}$	EAB address hold time after falling edge of write pulse when not using input registers	
$t_{EABWO}$	EAB write enable to data output valid delay	

Tables 39 through 47 show EPF10K10 and EPF10K20 device internal and external timing parameters.

<b>Table 39. EPF10K10 &amp; EPF10K20 Device LE Timing Microparameters</b> <i>Note (1)</i>					
Symbol	-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	
$t_{LUT}$		1.4		1.7	ns
$t_{CLUT}$		0.6		0.7	ns
$t_{RLUT}$		1.5		1.9	ns
$t_{PACKED}$		0.6		0.9	ns
$t_{EN}$		1.0		1.2	ns
$t_{CICO}$		0.2		0.3	ns
$t_{CGEN}$		0.9		1.2	ns
$t_{CGENR}$		0.9		1.2	ns
$t_{CASC}$		0.8		0.9	ns
$t_C$		1.3		1.5	ns
$t_{CO}$		0.9		1.1	ns
$t_{COMB}$		0.5		0.6	ns
$t_{SU}$	1.3		2.5		ns
$t_H$	1.4		1.6		ns
$t_{PRE}$		1.0		1.2	ns
$t_{CLR}$		1.0		1.2	ns
$t_{CH}$	4.0		4.0		ns
$t_{CL}$	4.0		4.0		ns

**Table 41. EPF10K10 & EPF10K20 Device EAB Internal Microparameters** *Note (1)*

Symbol	-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	
$t_{EABDATA1}$		1.5		1.9	ns
$t_{EABDATA2}$		4.8		6.0	ns
$t_{EABWE1}$		1.0		1.2	ns
$t_{EABWE2}$		5.0		6.2	ns
$t_{EABCLK}$		1.0		2.2	ns
$t_{EABCO}$		0.5		0.6	ns
$t_{EABYPASS}$		1.5		1.9	ns
$t_{EABSU}$	1.5		1.8		ns
$t_{EABH}$	2.0		2.5		ns
$t_{AA}$		8.7		10.7	ns
$t_{WP}$	5.8		7.2		ns
$t_{WDSU}$	1.6		2.0		ns
$t_{WDH}$	0.3		0.4		ns
$t_{WASU}$	0.5		0.6		ns
$t_{WAH}$	1.0		1.2		ns
$t_{WO}$		5.0		6.2	ns
$t_{DD}$		5.0		6.2	ns
$t_{EABOUT}$		0.5		0.6	ns
$t_{EABCH}$	4.0		4.0		ns
$t_{EABCL}$	5.8		7.2		ns

**Table 42. EPF10K10 & EPF10K20 Device EAB Internal Timing Macroparameters** *Note (1)*

Symbol	-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	
$t_{EABAA}$		13.7		17.0	ns
$t_{EABRCCOMB}$	13.7		17.0		ns
$t_{EABRCREG}$	9.7		11.9		ns
$t_{EABWP}$	5.8		7.2		ns
$t_{EABWCCOMB}$	7.3		9.0		ns
$t_{EABWCREG}$	13.0		16.0		ns
$t_{EABDD}$		10.0		12.5	ns
$t_{EABDATA CO}$		2.0		3.4	ns
$t_{EABDATASU}$	5.3		5.6		ns
$t_{EABDATAH}$	0.0		0.0		ns
$t_{EABWESU}$	5.5		5.8		ns
$t_{EABWEH}$	0.0		0.0		ns
$t_{EABWDSU}$	5.5		5.8		ns
$t_{EABWDH}$	0.0		0.0		ns
$t_{EABWASU}$	2.1		2.7		ns
$t_{EABWAH}$	0.0		0.0		ns
$t_{EABWO}$		9.5		11.8	ns

**Table 51. EPF10K30, EPF10K40 & EPF10K50 Device EAB Internal Timing Macroparameters***Note (1)*

Symbol	-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	
$t_{EABAA}$		13.7		17.0	ns
$t_{EABRCCOMB}$	13.7		17.0		ns
$t_{EABRCREG}$	9.7		11.9		ns
$t_{EABWP}$	5.8		7.2		ns
$t_{EABWCCOMB}$	7.3		9.0		ns
$t_{EABWCREG}$	13.0		16.0		ns
$t_{EABDD}$		10.0		12.5	ns
$t_{EABDATACO}$		2.0		3.4	ns
$t_{EABDATASU}$	5.3		5.6		ns
$t_{EABDATAH}$	0.0		0.0		ns
$t_{EABWESU}$	5.5		5.8		ns
$t_{EABWEH}$	0.0		0.0		ns
$t_{EABWDSU}$	5.5		5.8		ns
$t_{EABWDH}$	0.0		0.0		ns
$t_{EABWASU}$	2.1		2.7		ns
$t_{EABWAH}$	0.0		0.0		ns
$t_{EABWO}$		9.5		11.8	ns

**Notes to tables:**

- (1) All timing parameters are described in Tables 32 through 38 in this data sheet.
- (2) Using an LE to register the signal may provide a lower setup time.
- (3) This parameter is specified by characterization.

Tables 64 through 70 show EPF10K100 device internal and external timing parameters.

<b>Table 64. EPF10K100 Device LE Timing Microparameters</b> <i>Note (1)</i>							
Symbol	-3DX Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{LUT}$		1.5		1.5		2.0	ns
$t_{CLUT}$		0.4		0.4		0.5	ns
$t_{RLUT}$		1.6		1.6		2.0	ns
$t_{PACKED}$		0.9		0.9		1.3	ns
$t_{EN}$		0.9		0.9		1.2	ns
$t_{CICO}$		0.2		0.2		0.3	ns
$t_{CGEN}$		1.1		1.1		1.4	ns
$t_{CGENR}$		1.2		1.2		1.5	ns
$t_{CASC}$		1.1		1.1		1.3	ns
$t_C$		0.8		0.8		1.0	ns
$t_{CO}$		1.0		1.0		1.4	ns
$t_{COMB}$		0.5		0.5		0.7	ns
$t_{SU}$	2.1		2.1		2.6		ns
$t_H$	2.3		2.3		3.1		ns
$t_{PRE}$		1.0		1.0		1.4	ns
$t_{CLR}$		1.0		1.0		1.4	ns
$t_{CH}$	4.0		4.0		4.0		ns
$t_{CL}$	4.0		4.0		4.0		ns



**Table 82. EPF10K130V Device Interconnect Timing Microparameters** *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		8.0		9.0		9.5	ns
$t_{DIN2LE}$		2.4		3.0		3.1	ns
$t_{DIN2DATA}$		5.0		6.3		7.4	ns
$t_{DCLK2IOE}$		3.6		4.6		5.1	ns
$t_{DCLK2LE}$		2.4		3.0		3.1	ns
$t_{SAMELAB}$		0.4		0.6		0.8	ns
$t_{SAMEROW}$		4.5		5.3		6.5	ns
$t_{SAMECOLUMN}$		9.0		9.5		9.7	ns
$t_{DIFFROW}$		13.5		14.8		16.2	ns
$t_{TWOROWS}$		18.0		20.1		22.7	ns
$t_{LEPERIPH}$		8.1		8.6		9.5	ns
$t_{LABCARRY}$		0.6		0.8		1.0	ns
$t_{LABCASC}$		0.8		1.0		1.2	ns

**Table 83. EPF10K130V Device External Timing Parameters** *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{DRR}$		15.0		19.1		24.2	ns
$t_{INSU}$ (2), (3)	6.9		8.6		11.0		ns
$t_{INH}$ (3)	0.0		0.0		0.0		ns
$t_{OUTCO}$ (3)	2.0	7.8	2.0	9.9	2.0	11.3	ns

**Table 84. EPF10K130V Device External Bidirectional Timing Parameters** *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	6.7		8.5		10.8		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$	2.0	6.9	2.0	8.8	2.0	10.2	ns
$t_{XZBIDIR}$		12.9		16.4		19.3	ns
$t_{ZXBIDIR}$		12.9		16.4		19.3	ns

**Notes to tables:**

- (1) All timing parameters are described in Tables 32 through 38 in this data sheet.
- (2) Using an LE to register the signal may provide a lower setup time.
- (3) This parameter is specified by characterization.

Tables 99 through 105 show EPF10K100A device internal and external timing parameters.

**Table 99. EPF10K100A Device LE Timing Microparameters** *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{LUT}$		1.0		1.2		1.4	ns
$t_{CLUT}$		0.8		0.9		1.1	ns
$t_{RLUT}$		1.4		1.6		1.9	ns
$t_{PACKED}$		0.4		0.5		0.5	ns
$t_{EN}$		0.6		0.7		0.8	ns
$t_{CICO}$		0.2		0.2		0.3	ns
$t_{CGEN}$		0.4		0.4		0.6	ns
$t_{CGENR}$		0.6		0.7		0.8	ns
$t_{CASC}$		0.7		0.9		1.0	ns
$t_C$		0.9		1.0		1.2	ns
$t_{CO}$		0.2		0.3		0.3	ns
$t_{COMB}$		0.6		0.7		0.8	ns
$t_{SU}$	0.8		1.0		1.2		ns
$t_H$	0.3		0.5		0.5		ns
$t_{PRE}$		0.3		0.3		0.4	ns
$t_{CLR}$		0.3		0.3		0.4	ns
$t_{CH}$	2.5		3.5		4.0		ns
$t_{CL}$	2.5		3.5		4.0		ns

**Table 108. EPF10K250A Device EAB Internal Microparameters** *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		1.3		1.5		1.7	ns
$t_{EABDATA2}$		1.3		1.5		1.7	ns
$t_{EABWE1}$		0.9		1.1		1.3	ns
$t_{EABWE2}$		5.0		5.7		6.7	ns
$t_{EABCLK}$		0.6		0.7		0.8	ns
$t_{EABCO}$		0.0		0.0		0.0	ns
$t_{EABYPASS}$		0.1		0.1		0.2	ns
$t_{EABSU}$	3.8		4.3		5.0		ns
$t_{EABH}$	0.7		0.8		0.9		ns
$t_{AA}$		4.5		5.0		5.9	ns
$t_{WP}$	5.6		6.4		7.5		ns
$t_{WDSU}$	1.3		1.4		1.7		ns
$t_{WDH}$	0.1		0.1		0.2		ns
$t_{WASU}$	0.1		0.1		0.2		ns
$t_{WAH}$	0.1		0.1		0.2		ns
$t_{WO}$		4.1		4.6		5.5	ns
$t_{DD}$		4.1		4.6		5.5	ns
$t_{EABOUT}$		0.1		0.1		0.2	ns
$t_{EABCH}$	2.5		3.0		3.5		ns
$t_{EABCL}$	5.6		6.4		7.5		ns

**Notes to tables:**

- (1) All timing parameters are described in Tables 32 through 37 in this data sheet.
- (2) Using an LE to register the signal may provide a lower setup time.
- (3) This parameter is specified by characterization.

## ClockLock & ClockBoost Timing Parameters

For the ClockLock and ClockBoost circuitry to function properly, the incoming clock must meet certain requirements. If these specifications are not met, the circuitry may not lock onto the incoming clock, which generates an erroneous clock within the device. The clock generated by the ClockLock and ClockBoost circuitry must also meet certain specifications. If the incoming clock meets these requirements during configuration, the ClockLock and ClockBoost circuitry will lock onto the clock during configuration. The circuit will be ready for use immediately after configuration. Figure 31 illustrates the incoming and generated clock specifications.

**Figure 31. Specifications for the Incoming & Generated Clocks**

The  $t_I$  parameter refers to the nominal input clock period; the  $t_O$  parameter refers to the nominal output clock period.

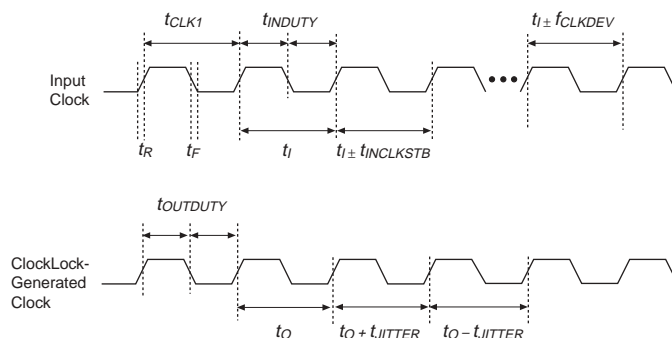


Table 113 summarizes the ClockLock and ClockBoost parameters.

<b>Table 113. ClockLock &amp; ClockBoost Parameters (Part 1 of 2)</b>					
Symbol	Parameter	Min	Typ	Max	Unit
$t_R$	Input rise time			2	ns
$t_F$	Input fall time			2	ns
$t_{INDUTY}$	Input duty cycle	45		55	%
$f_{CLK1}$	Input clock frequency (ClockBoost clock multiplication factor equals 1)	30		80	MHz
$t_{CLK1}$	Input clock period (ClockBoost clock multiplication factor equals 1)	12.5		33.3	ns
$f_{CLK2}$	Input clock frequency (ClockBoost clock multiplication factor equals 2)	16		50	MHz
$t_{CLK2}$	Input clock period (ClockBoost clock multiplication factor equals 2)	20		62.5	ns



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:**  
[lit\\_req@altera.com](mailto:lit_req@altera.com)

Copyright © 2003 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. 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.



I.S. EN ISO 9001