

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	102
Number of Gates	31000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	144-LQFP
Supplier Device Package	144-TQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/intel/epf10k10atc144-3

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.

The FLEX 10K architecture is similar to that of embedded gate arrays, the fastest-growing segment of the gate array market. As with standard gate arrays, embedded gate arrays implement general logic in a conventional “sea-of-gates” architecture. In addition, embedded gate arrays have dedicated die areas for implementing large, specialized functions. By embedding functions in silicon, embedded gate arrays provide reduced die area and increased speed compared to standard gate arrays. However, embedded megafunctions typically cannot be customized, limiting the designer’s options. In contrast, FLEX 10K devices are programmable, providing the designer with full control over embedded megafunctions and general logic while facilitating iterative design changes during debugging.

Each FLEX 10K device contains an embedded array and a logic array. The embedded array is used to implement a variety of memory functions or complex logic functions, such as digital signal processing (DSP), microcontroller, wide-data-path manipulation, and data-transformation functions. The logic array performs the same function as the sea-of-gates in the gate array: it is used to implement general logic, such as counters, adders, state machines, and multiplexers. The combination of embedded and logic arrays provides the high performance and high density of embedded gate arrays, enabling designers to implement an entire system on a single device.

FLEX 10K devices are configured at system power-up with data stored in an Altera serial configuration device or provided by a system controller. Altera offers the EPC1, EPC2, EPC16, and EPC1441 configuration devices, which configure FLEX 10K devices via a serial data stream. Configuration data can also be downloaded from system RAM or from Altera’s BitBlaster™ serial download cable or ByteBlasterMV™ parallel port download cable. After a FLEX 10K device has been configured, it can be reconfigured in-circuit by resetting the device and loading new data. Because reconfiguration requires less than 320 ms, real-time changes can be made during system operation.

FLEX 10K devices contain an optimized interface that permits microprocessors to configure FLEX 10K devices serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat a FLEX 10K device as memory and configure the device by writing to a virtual memory location, making it very easy for the designer to reconfigure the device.

Signals on the peripheral control bus can also drive the four global signals, referred to as GLOBAL0 through GLOBAL3 in [Tables 8 and 9](#). The internally generated signal can drive the global signal, providing the same low-skew, low-delay characteristics for an internally generated signal as for a signal driven by an input. This feature is ideal for internally generated clear or clock signals with high fan-out. When a global signal is driven by internal logic, the dedicated input pin that drives that global signal cannot be used. The dedicated input pin should be driven to a known logic state (such as ground) and not be allowed to float.

When the chip-wide output enable pin is held low, it will tri-state all pins on the device. This option can be set in the Global Project Device Options menu. Additionally, the registers in the IOE can be reset by holding the chip-wide reset pin low.

Row-to-IOE Connections

When an IOE is used as an input signal, it can drive two separate row channels. The signal is accessible by all LEs within that row. When an IOE is used as an output, the signal is driven by a multiplexer that selects a signal from the row channels. Up to eight IOEs connect to each side of each row channel. See [Figure 14](#).

Figure 14. FLEX 10K Row-to-IOE Connections

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

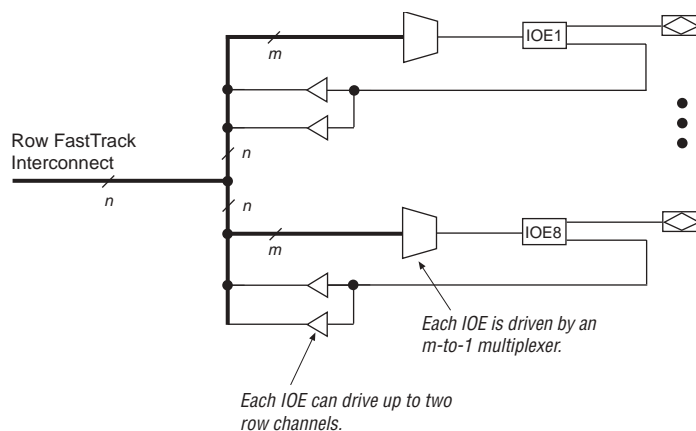


Table 10 lists the FLEX 10K row-to-IOE interconnect resources.

Table 10. FLEX 10K Row-to-IOE Interconnect Resources		
Device	Channels per Row (<i>n</i>)	Row Channels per Pin (<i>m</i>)
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.

ClockLock & ClockBoost Features

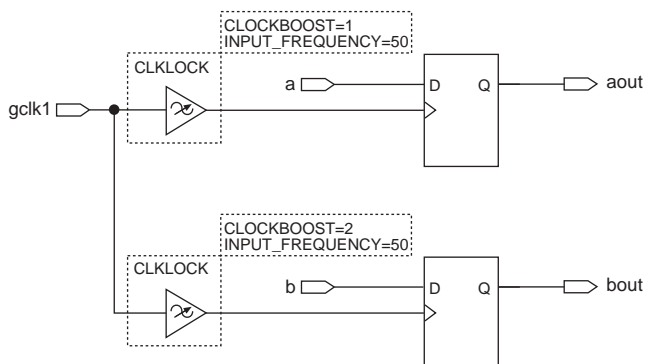
To support high-speed designs, selected FLEX 10K devices offer optional ClockLock and ClockBoost circuitry containing a phase-locked loop (PLL) that is used to increase design speed and reduce resource usage. The ClockLock circuitry uses a synchronizing PLL that reduces the clock delay and skew within a device. This reduction minimizes clock-to-output and setup times while maintaining zero hold times. The ClockBoost circuitry, which provides a clock multiplier, allows the designer to enhance device area efficiency by sharing resources within the device. The ClockBoost feature allows the designer to distribute a low-speed clock and multiply that clock on-device. Combined, the ClockLock and ClockBoost features provide significant improvements in system performance and bandwidth.

The ClockLock and ClockBoost features in FLEX 10K devices are enabled through the Altera software. External devices are not required to use these features. The output of the ClockLock and ClockBoost circuits is not available at any of the device pins.

The ClockLock and ClockBoost circuitry locks onto the rising edge of the incoming clock. The circuit output can only drive the clock inputs of registers; the generated clock cannot be gated or inverted.

The dedicated clock pin (GCLK1) supplies the clock to the ClockLock and ClockBoost circuitry. When the dedicated clock pin is driving the ClockLock or ClockBoost circuitry, it cannot drive elsewhere in the device.

In designs that require both a multiplied and non-multiplied clock, the clock trace on the board can be connected to GCLK1. With the Altera software, GCLK1 can feed both the ClockLock and ClockBoost circuitry in the FLEX 10K device. However, when both circuits are used, the other clock pin (GCLK0) cannot be used. [Figure 17](#) shows a block diagram of how to enable both the ClockLock and ClockBoost circuits in the Altera software. The example shown is a schematic, but a similar approach applies for designs created in AHDL, VHDL, and Verilog HDL. When the ClockLock and ClockBoost circuits are used simultaneously, the input frequency parameter must be the same for both circuits. In [Figure 17](#), the input frequency must meet the requirements specified when the ClockBoost multiplication factor is two.

Figure 17. Enabling ClockLock & ClockBoost in the Same Design

To use both the ClockLock and ClockBoost circuits in the same design, designers must use Revision C EPF10K100GC503-3DX devices and MAX+PLUS II software versions 7.2 or higher. The die revision is indicated by the third digit of the nine-digit code on the top side of the device.

Output Configuration

This section discusses the peripheral component interconnect (PCI) pull-up clamping diode option, slew-rate control, open-drain output option, MultiVolt I/O interface, and power sequencing for FLEX 10K devices. The PCI pull-up clamping diode, slew-rate control, and open-drain output options are controlled pin-by-pin via Altera logic options. The MultiVolt I/O interface is controlled by connecting V_{CCIO} to a different voltage than V_{CCINT} . Its effect can be simulated in the Altera software via the **Global Project Device Options** dialog box (Assign menu).

PCI Clamping Diodes

The EPF10K10A and EPF10K30A devices have a pull-up clamping diode on every I/O, dedicated input, and dedicated clock pin. PCI clamping diodes clamp the transient overshoot caused by reflected waves to the V_{CCIO} value and are required for 3.3-V PCI compliance. Clamping diodes can also be used to limit overshoot in other systems.

Clamping diodes are controlled on a pin-by-pin basis via a logic option in the Altera software. When V_{CCIO} is 3.3 V, a pin that has the clamping diode turned on can be driven by a 2.5-V or 3.3-V signal, but not a 5.0-V signal. When V_{CCIO} is 2.5 V, a pin that has the clamping diode turned on can be driven by a 2.5-V signal, but not a 3.3-V or 5.0-V signal. However, a clamping diode can be turned on for a subset of pins, which allows devices to bridge between a 3.3-V PCI bus and a 5.0-V device.

Table 19. FLEX 10K 5.0-V Device DC Operating Conditions *Notes (5), (6)*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	High-level input voltage		2.0		$V_{CCINT} + 0.5$	V
V_{IL}	Low-level input voltage		-0.5		0.8	V
V_{OH}	5.0-V high-level TTL output voltage	$I_{OH} = -4$ mA DC, $V_{CCIO} = 4.75$ V (7)	2.4			V
	3.3-V high-level TTL output voltage	$I_{OH} = -4$ mA DC, $V_{CCIO} = 3.00$ V (7)	2.4			V
	3.3-V high-level CMOS output voltage	$I_{OH} = -0.1$ mA DC, $V_{CCIO} = 3.00$ V (7)	$V_{CCIO} - 0.2$			V
V_{OL}	5.0-V low-level TTL output voltage	$I_{OL} = 12$ mA DC, $V_{CCIO} = 4.75$ V (8)			0.45	V
	3.3-V low-level TTL output voltage	$I_{OL} = 12$ mA DC, $V_{CCIO} = 3.00$ V (8)			0.45	V
	3.3-V low-level CMOS output voltage	$I_{OL} = 0.1$ mA DC, $V_{CCIO} = 3.00$ V (8)			0.2	V
I_I	Input pin leakage current	$V_I = V_{CC}$ or ground (9)	-10		10	μ A
I_{OZ}	Tri-stated I/O pin leakage current	$V_O = V_{CC}$ or ground (9)	-40		40	μ A
I_{CC0}	V_{CC} supply current (standby)	$V_I =$ ground, no load		0.5	10	mA

Table 20. 5.0-V Device Capacitance of EPF10K10, EPF10K20 & EPF10K30 Devices *Note (10)*

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

Table 21. 5.0-V Device Capacitance of EPF10K40, EPF10K50, EPF10K70 & EPF10K100 Devices *Note (10)*

Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Input capacitance	$V_{IN} = 0$ V, $f = 1.0$ MHz		10	pF
C_{INCLK}	Input capacitance on dedicated clock pin	$V_{IN} = 0$ V, $f = 1.0$ MHz		15	pF
C_{OUT}	Output capacitance	$V_{OUT} = 0$ V, $f = 1.0$ MHz		10	pF

Timing simulation and delay prediction are available with the MAX+PLUS II 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 24 shows the overall timing model, which maps the possible paths to and from the various elements of the FLEX 10K device.

Figure 24. FLEX 10K Device Timing Model

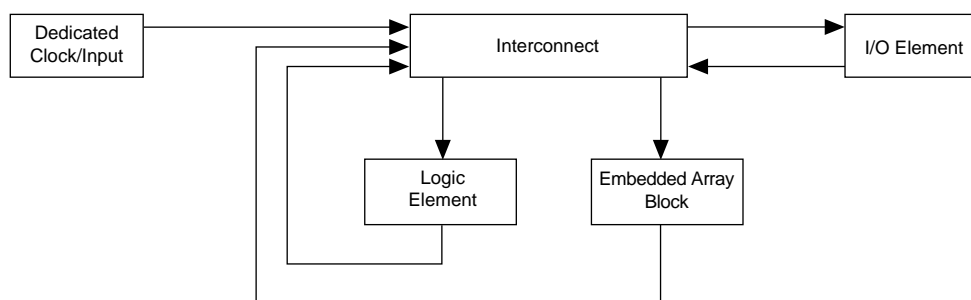


Table 35. EAB Timing Macroparameters *Notes (1), (6)*

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 48 through 56 show EPF10K30, EPF10K40, and EPF10K50 device internal and external timing parameters.

Table 48. EPF10K30, EPF10K40 & EPF10K50 Device LE Timing Microparameters <i>Note (1)</i>					
Symbol	-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	
t_{LUT}		1.3		1.8	ns
t_{CLUT}		0.6		0.6	ns
t_{RLUT}		1.5		2.0	ns
t_{PACKED}		0.5		0.8	ns
t_{EN}		0.9		1.5	ns
t_{CICO}		0.2		0.4	ns
t_{CGEN}		0.9		1.4	ns
t_{CGENR}		0.9		1.4	ns
t_{CASC}		1.0		1.2	ns
t_C		1.3		1.6	ns
t_{CO}		0.9		1.2	ns
t_{COMB}		0.6		0.6	ns
t_{SU}	1.4		1.4		ns
t_H	0.9		1.3		ns
t_{PRE}		0.9		1.2	ns
t_{CLR}		0.9		1.2	ns
t_{CH}	4.0		4.0		ns
t_{CL}	4.0		4.0		ns

Table 50. EPF10K30, EPF10K40 & EPF10K50 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

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 57 through 63 show EPF10K70 device internal and external timing parameters.

Table 57. EPF10K70 Device LE Timing Microparameters *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{LUT}		1.3		1.5		2.0	ns
t_{CLUT}		0.4		0.4		0.5	ns
t_{RLUT}		1.5		1.6		2.0	ns
t_{PACKED}		0.8		0.9		1.3	ns
t_{EN}		0.8		0.9		1.2	ns
t_{CICO}		0.2		0.2		0.3	ns
t_{CGEN}		1.0		1.1		1.4	ns
t_{CGENR}		1.1		1.2		1.5	ns
t_{CASC}		1.0		1.1		1.3	ns
t_C		0.7		0.8		1.0	ns
t_{CO}		0.9		1.0		1.4	ns
t_{COMB}		0.4		0.5		0.7	ns
t_{SU}	1.9		2.1		2.6		ns
t_H	2.1		2.3		3.1		ns
t_{PRE}		0.9		1.0		1.4	ns
t_{CLR}		0.9		1.0		1.4	ns
t_{CH}	4.0		4.0		4.0		ns
t_{CL}	4.0		4.0		4.0		ns

Table 61. EPF10K70 Device Interconnect Timing Microparameters *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		6.6		7.3		8.8	ns
t_{DIN2LE}		4.2		4.8		6.0	ns
$t_{DIN2DATA}$		6.5		7.1		10.8	ns
$t_{DCLK2IOE}$		5.5		6.2		7.7	ns
$t_{DCLK2LE}$		4.2		4.8		6.0	ns
$t_{SAMELAB}$		0.4		0.4		0.5	ns
$t_{SAMEROW}$		4.8		4.9		5.5	ns
$t_{SAMECOLUMN}$		3.3		3.4		3.7	ns
$t_{DIFFROW}$		8.1		8.3		9.2	ns
$t_{TROWROWS}$		12.9		13.2		14.7	ns
$t_{LEPERIPH}$		5.5		5.7		6.5	ns
$t_{LABCARRY}$		0.8		0.9		1.1	ns
$t_{LABCASC}$		2.7		3.0		3.2	ns

Table 62. EPF10K70 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}		17.2		19.1		24.2	ns
t_{INSU} (2), (3)	6.6		7.3		8.0		ns
t_{INH} (3)	0.0		0.0		0.0		ns
t_{OUTCO} (3)	2.0	9.9	2.0	11.1	2.0	14.3	ns

Table 63. EPF10K70 Device External Bidirectional Timing Parameters *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	7.4		8.1		10.4		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$	2.0	9.9	2.0	11.1	2.0	14.3	ns
$t_{XZBIDIR}$		13.7		15.4		18.5	ns
$t_{ZXBIDIR}$		13.7		15.4		18.5	ns

Table 72. EPF10K50V Device IOE Timing Microparameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	Min	Max	
t_{IOD}		1.2		1.6		1.9		2.1	ns
t_{IOC}		0.3		0.4		0.5		0.5	ns
t_{IOCO}		0.3		0.3		0.4		0.4	ns
t_{IOCOMB}		0.0		0.0		0.0		0.0	ns
t_{IOSU}	2.8		2.8		3.4		3.9		ns
t_{IOH}	0.7		0.8		1.0		1.4		ns
t_{IOCLR}		0.5		0.6		0.7		0.7	ns
t_{OD1}		2.8		3.2		3.9		4.7	ns
t_{OD2}		—		—		—		—	ns
t_{OD3}		6.5		6.9		7.6		8.4	ns
t_{XZ}		2.8		3.1		3.8		4.6	ns
t_{ZX1}		2.8		3.1		3.8		4.6	ns
t_{ZX2}		—		—		—		—	ns
t_{ZX3}		6.5		6.8		7.5		8.3	ns
t_{INREG}		5.0		5.7		7.0		9.0	ns
t_{IOFD}		1.5		1.9		2.3		2.7	ns
t_{INCOMB}		1.5		1.9		2.3		2.7	ns

Table 80. EPF10K130V Device EAB Internal Microparameters *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		1.9		2.4		2.4	ns
$t_{EABDATA2}$		3.7		4.7		4.7	ns
t_{EABWE1}		1.9		2.4		2.4	ns
t_{EABWE2}		3.7		4.7		4.7	ns
t_{EABCLK}		0.7		0.9		0.9	ns
t_{EABCO}		0.5		0.6		0.6	ns
$t_{EABYPASS}$		0.6		0.8		0.8	ns
t_{EABSU}	1.4		1.8		1.8		ns
t_{EABH}	0.0		0.0		0.0		ns
t_{AA}		5.6		7.1		7.1	ns
t_{WP}	3.7		4.7		4.7		ns
t_{WDSU}	4.6		5.9		5.9		ns
t_{WDH}	0.0		0.0		0.0		ns
t_{WASU}	3.9		5.0		5.0		ns
t_{WAH}	0.0		0.0		0.0		ns
t_{WO}		5.6		7.1		7.1	ns
t_{DD}		5.6		7.1		7.1	ns
t_{EABOUT}		2.4		3.1		3.1	ns
t_{EABCH}	4.0		4.0		4.0		ns
t_{EABCL}	4.0		4.7		4.7		ns

Table 100. EPF10K100A Device IOE Timing Microparameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{IOD}		2.5		2.9		3.4	ns
t_{IOC}		0.3		0.3		0.4	ns
t_{IOCO}		0.2		0.2		0.3	ns
t_{IOCOMB}		0.5		0.6		0.7	ns
t_{IOSU}	1.3		1.7		1.8		ns
t_{IOH}	0.2		0.2		0.3		ns
t_{IOCLR}		1.0		1.2		1.4	ns
t_{OD1}		2.2		2.6		3.0	ns
t_{OD2}		4.5		5.3		6.1	ns
t_{OD3}		6.8		7.9		9.3	ns
t_{XZ}		2.7		3.1		3.7	ns
t_{ZX1}		2.7		3.1		3.7	ns
t_{ZX2}		5.0		5.8		6.8	ns
t_{ZX3}		7.3		8.4		10.0	ns
t_{INREG}		5.3		6.1		7.2	ns
t_{IOFD}		4.7		5.5		6.4	ns
t_{INCOMB}		4.7		5.5		6.4	ns

Table 101. EPF10K100A Device EAB Internal Microparameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		1.8		2.1		2.4	ns
$t_{EABDATA2}$		3.2		3.7		4.4	ns
t_{EABWE1}		0.8		0.9		1.1	ns
t_{EABWE2}		2.3		2.7		3.1	ns
t_{EABCLK}		0.8		0.9		1.1	ns
t_{EABCO}		1.0		1.1		1.4	ns
$t_{EABYPASS}$		0.3		0.3		0.4	ns
t_{EABSU}	1.3		1.5		1.8		ns
t_{EABH}	0.4		0.5		0.5		ns
t_{AA}		4.1		4.8		5.6	ns
t_{WP}	3.2		3.7		4.4		ns
t_{WDSU}	2.4		2.8		3.3		ns
t_{WDH}	0.2		0.2		0.3		ns
t_{WASU}	0.2		0.2		0.3		ns
t_{WAH}	0.0		0.0		0.0		ns
t_{WO}		3.4		3.9		4.6	ns
t_{DD}		3.4		3.9		4.6	ns
t_{EABOUT}		0.3		0.3		0.4	ns
t_{EABCH}	2.5		3.5		4.0		ns
t_{EABCL}	3.2		3.7		4.4		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

- f_{MAX} = Maximum operating frequency in MHz
 N = Total number of logic cells used in the device
 tog_{LC} = Average percent of logic cells toggling at each clock (typically 12.5%)
 K = Constant, shown in Tables 114 and 115

Table 114. FLEX 10K K Constant Values

Device	K Value
EPF10K10	82
EPF10K20	89
EPF10K30	88
EPF10K40	92
EPF10K50	95
EPF10K70	85
EPF10K100	88

Table 115. FLEX 10KA K Constant Values

Device	K Value
EPF10K10A	17
EPF10K30A	17
EPF10K50V	19
EPF10K100A	19
EPF10K130V	22
EPF10K250A	23

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) 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.

Figure 32 shows the relationship between the current and operating frequency of FLEX 10K devices.

SRAM configuration elements allow FLEX 10K 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 may be completed in less than 320 ms using an EPF10K250A device with a DCLK frequency of 10 MHz. This process can be used to reconfigure an entire system dynamically. In-field upgrades can be performed by distributing new configuration files.



Refer to the configuration device data sheet to obtain the POR delay when using a configuration device method.

Programming Files

Despite being function- and pin-compatible, FLEX 10KA and FLEX 10KE devices are not programming- or configuration-file compatible with FLEX 10K devices. A design should be recompiled before it is transferred from a FLEX 10K device to an equivalent FLEX 10KA or FLEX 10KE device. This recompilation should be performed to create a new programming or configuration file and to check design timing on the faster FLEX 10KA or FLEX 10KE device. The programming or configuration files for EPF10K50 devices can program or configure an EPF10K50V device. However, Altera recommends recompiling a design for the EPF10K50V device when transferring it from the EPF10K50 device.

Configuration Schemes

The configuration data for a FLEX 10K device can be loaded with one of five configuration schemes (see [Table 116](#)), chosen on the basis of the target application. An EPC1, EPC2, EPC16, or EPC1441 configuration device, intelligent controller, or the JTAG port can be used to control the configuration of a FLEX 10K device, allowing automatic configuration on system power-up.