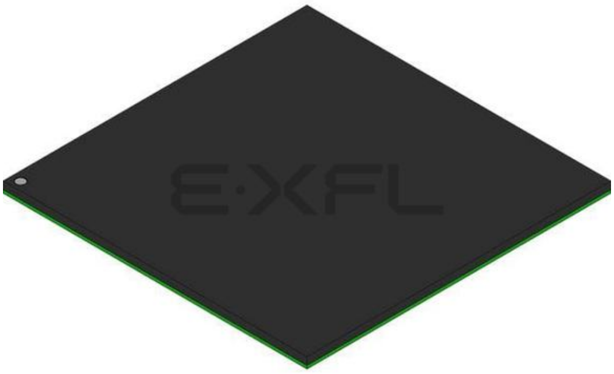


Welcome to E-XFL.COM

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	Active
Number of LABs/CLBs	624
Number of Logic Elements/Cells	-
Total RAM Bits	-
Number of I/O	274
Number of Gates	-
Voltage - Supply	2.375V ~ 2.625V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	356-LBGA
Supplier Device Package	356-BGA (35x35)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=epf10k100ebc356-2

For more information on FLEX device configuration, 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*
- *MasterBlaster Download Cable Data Sheet*
- *Application Note 116 (Configuring APEX 20K, FLEX 10K, & FLEX 6000 Devices)*

FLEX 10KE devices are supported by the Altera development systems, which are 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 system includes DesignWare functions that are optimized for the FLEX 10KE architecture.

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

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

Each FLEX 10KE device contains an enhanced 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 4,096 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.

The logic array consists of logic array blocks (LABs). Each LAB contains eight LEs and a local interconnect. An LE consists of a four-input look-up table (LUT), a programmable flipflop, and dedicated signal paths for carry and cascade functions. The eight LEs can be used to create medium-sized blocks of logic—such as 8-bit counters, address decoders, or state machines—or combined across LABs to create larger logic blocks. Each LAB represents about 96 usable gates of logic.

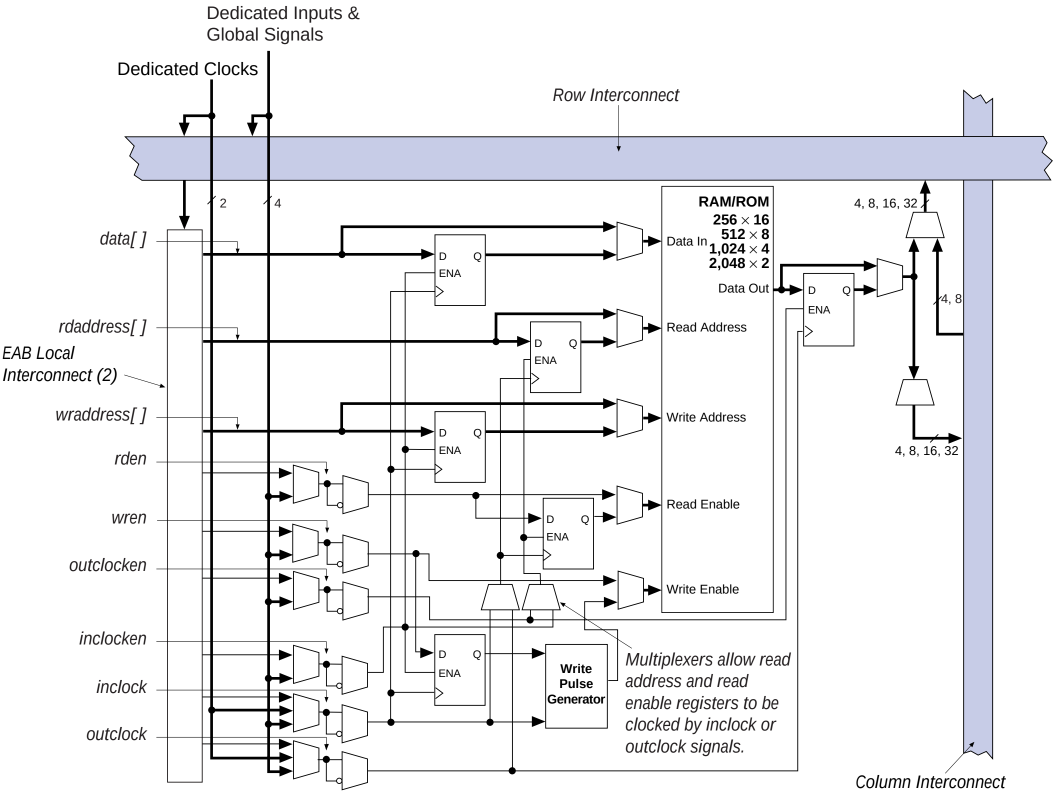
Signal interconnections within FLEX 10KE devices (as well as to and from device pins) are provided by the FastTrack Interconnect routing structure, which is a series of fast, continuous row and column channels that run the entire length and width of the device.

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 routing structure. Each IOE contains a bidirectional I/O buffer and a flipflop that can be used as either an output or input register to feed input, output, or bidirectional signals. When used with a dedicated clock pin, these registers provide exceptional performance. As inputs, they provide setup times as low as 0.9 ns and hold times of 0 ns. As outputs, these registers provide clock-to-output times as low as 3.0 ns. IOEs provide a variety of features, such as JTAG BST support, slew-rate control, tri-state buffers, and open-drain outputs.

The EAB can also be used for bidirectional, dual-port memory applications where two ports read or write simultaneously. To implement this type of dual-port memory, two EABs are used to support two simultaneous read or writes.

Alternatively, one clock and clock enable can be used to control the input registers of the EAB, while a different clock and clock enable control the output registers (see Figure 2).

Figure 2. FLEX 10KE Device in Dual-Port RAM Mode *Notes (1)*



Multiplexers allow read address and read enable registers to be clocked by inclock or outclock signals.

Notes:
(1) All registers can be asynchronously cleared by EAB local interconnect signals, global signals, or the chip-wide reset.
(2) EPF10K30E and EPF10K50E devices have 88 EAB local interconnect channels; EPF10K100E, EPF10K130E, and EPF10K200E devices have 104 EAB local interconnect channels.

The programmable flipflop in the LE can be configured for D, T, JK, or SR operation. The clock, clear, and preset 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 output of the LE.

The LE has two outputs that drive the interconnect: one drives the local interconnect and the other drives either the row or column FastTrack Interconnect routing structure. The two outputs can be controlled independently. For example, the LUT can drive one output while the register drives the other output. This feature, called register packing, can improve LE utilization because the register and the LUT can be used for unrelated functions.

The FLEX 10KE architecture provides two types of dedicated high-speed data paths that connect adjacent LEs without using local interconnect paths: carry chains and cascade chains. The carry chain supports high-speed counters and adders and the cascade chain implements wide-input functions with minimum delay. Carry and cascade chains connect all LEs in a LAB as well as all LABs in the same row. Intensive use of carry and cascade chains can reduce routing flexibility. Therefore, the use of these chains should be limited to speed-critical portions of a design.

Carry Chain

The carry chain provides a very fast (as low as 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 10KE architecture to implement high-speed counters, adders, and comparators of arbitrary width efficiently. Carry chain logic can be created automatically by the Altera 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.

Carry chains longer than eight 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 even-numbered LAB to even-numbered LAB, or from odd-numbered LAB to odd-numbered LAB. For example, the last LE of the first LAB in a row carries to the first LE of the third LAB in the row. The carry chain does not cross the EAB at the middle of the row. For instance, in the EPF10K50E device, the carry chain stops at the eighteenth LAB and a new one begins at the nineteenth LAB.

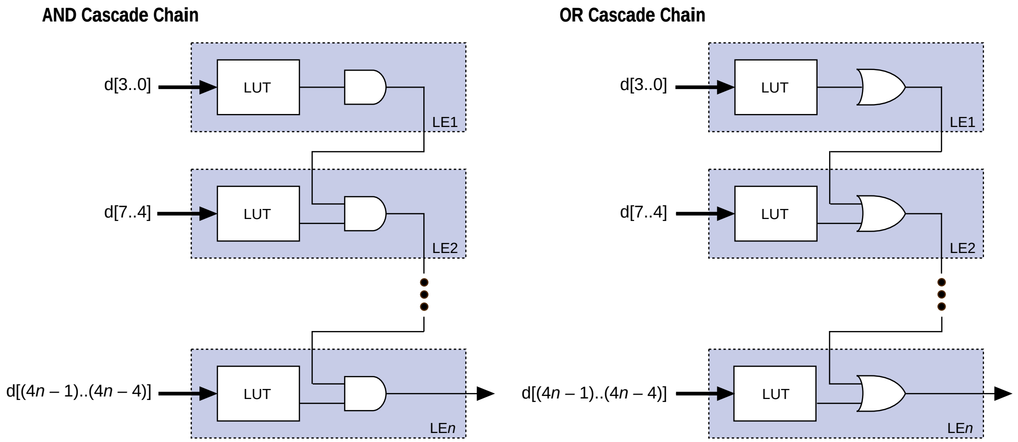
Cascade Chain

With the cascade chain, the FLEX 10KE 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. An a delay as low as 0.6 ns per LE, each additional LE provides four more inputs to the effective width of a function. Cascade chain logic can be created automatically by the Altera 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 EPF10K50E 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 10 shows how the cascade function can connect adjacent LEs to form functions with a wide fan-in. These examples show functions of 4*n* variables implemented with *n* LEs. The LE delay is 0.9 ns; the cascade chain delay is 0.6 ns. With the cascade chain, 2.7 ns are needed to decode a 16-bit address.

Figure 10. FLEX 10KE Cascade Chain Operation



LE Operating Modes

The FLEX 10KE LE can operate in the following four modes:

- Normal mode
- Arithmetic mode
- Up/down counter mode
- Clearable 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. Three inputs to the LE provide clock, clear, and preset control for the register. The Altera software, in conjunction with parameterized functions such as LPM and DesignWare functions, automatically chooses the appropriate mode for common functions such as counters, adders, and multipliers. If required, the designer can also create special-purpose functions that use a specific LE operating mode for optimal performance.

The architecture provides a synchronous clock enable to the register in all four modes. The Altera software can set `DATA1` to enable the register synchronously, providing easy implementation of fully synchronous designs.

Asynchronous Clear

The flipflop can be cleared by either `LABCTRL1` or `LABCTRL2`. In this mode, the preset signal is tied to `VCC` to deactivate it.

Asynchronous Preset

An asynchronous preset is implemented as an asynchronous load, or with an asynchronous clear. If `DATA3` is tied to `VCC`, 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 `VCC`, so that 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.

Tables 12 and 13 summarize the ClockLock and ClockBoost parameters for -1 and -2 speed-grade devices, respectively.

Table 12. ClockLock & ClockBoost Parameters for -1 Speed-Grade Devices

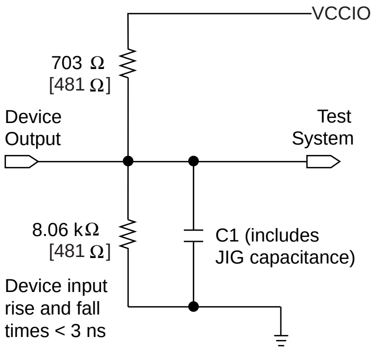
Symbol	Parameter	Condition	Min	Typ	Max	Unit
t_R	Input rise time				5	ns
t_F	Input fall time				5	ns
t_{INDUTY}	Input duty cycle		40		60	%
f_{CLK1}	Input clock frequency (ClockBoost clock multiplication factor equals 1)		25		180	MHz
f_{CLK2}	Input clock frequency (ClockBoost clock multiplication factor equals 2)		16		90	MHz
f_{CLKDEV}	Input deviation from user specification in the MAX+PLUS II software *(1)*				25,000 *(2)*	PPM
$t_{INCLKSTB}$	Input clock stability (measured between adjacent clocks)				100	ps
t_{LOCK}	Time required for ClockLock or ClockBoost to acquire lock *(3)*				10	µs
t_{JITTER}	Jitter on ClockLock or ClockBoost-generated clock *(4)*	$t_{INCLKSTB} < 100$			250	ps
		$t_{INCLKSTB} < 50$			200 *(4)*	ps
$t_{OUTDUTY}$	Duty cycle for ClockLock or ClockBoost-generated clock		40	50	60	%

Generic Testing

Each FLEX 10KE device is functionally tested. Complete testing of each configurable static random access memory (SRAM) bit and all logic functionality ensures 100% yield. AC test measurements for FLEX 10KE devices are made under conditions equivalent to those shown in Figure 21. Multiple test patterns can be used to configure devices during all stages of the production flow.

Figure 21. FLEX 10KE 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 brackets are for 2.5-V devices or outputs. Numbers without brackets are for 3.3-V. devices or outputs.



Operating Conditions

Tables 19 through 23 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 2.5-V FLEX 10KE devices.

Table 19. FLEX 10KE 2.5-V Device Absolute Maximum Ratings *Note (1)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage	With respect to ground *(2)*	–0.5	3.6	V
V_{CCIO}			–0.5	4.6	V
V_I	DC input voltage		–2.0	5.75	V
I_{OUT}	DC output current, per pin		–25	25	mA
T_{STG}	Storage temperature	No bias	–65	150	° C
T_{AMB}	Ambient temperature	Under bias	–65	135	° C
T_J	Junction temperature	PQFP, TQFP, BGA, and FineLine BGA packages, under bias		135	° C
		Ceramic PGA packages, under bias		150	° C

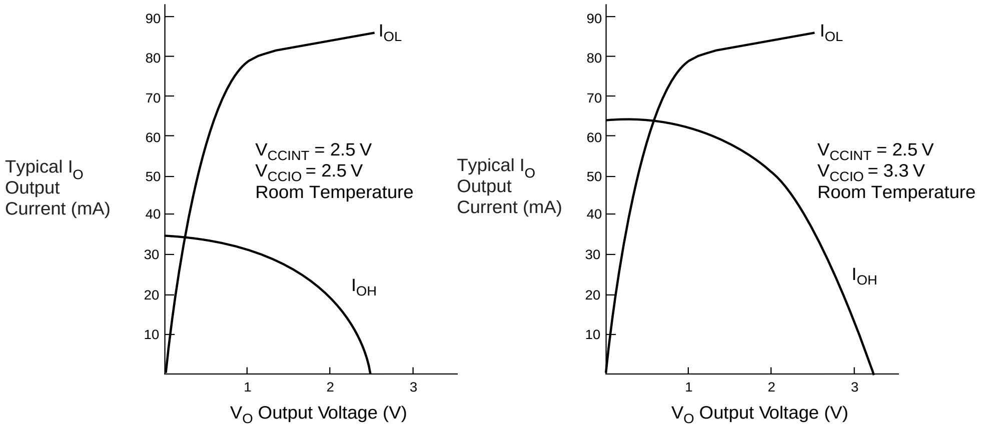
Table 20. 2.5-V EPF10K50E & EPF10K200E Device Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	(3), (4)	2.30 (2.30)	2.70 (2.70)	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 21. 2.5-V EPF10K30E, EPF10K50S, EPF10K100E, EPF10K130E & EPF10K200S Device Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	(3), (4)	2.375 (2.375)	2.625 (2.625)	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)	2.375 (2.375)	2.625 (2.625)	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

Figure 23. Output Drive Characteristics of FLEX 10KE Devices *Note (1)*



Note:
(1) These are transient (AC) currents.

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})
- Interconnect delay ($t_{SAMEROW}$)
- LE look-up table delay (t_{LUT})
- 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.

Table 30. External Bidirectional Timing Parameters *Note (9)*

Symbol	Parameter	Conditions
$t_{INSUBIDIR}$	Setup time for bi-directional pins with global clock at same-row or same-column LE register	
$t_{INHBIDIR}$	Hold time for bidirectional pins with global clock at same-row or same-column LE register	
t_{INH}	Hold time with global clock at IOE register	
$t_{OUTCOBIDIR}$	Clock-to-output delay for bidirectional pins with global clock at IOE register	C1 = 35 pF
$t_{XZBIDIR}$	Synchronous IOE output buffer disable delay	C1 = 35 pF
$t_{ZXBIDIR}$	Synchronous IOE output buffer enable delay, slow slew rate= off	C1 = 35 pF

Notes to tables:

(1) Microparameters are timing delays contributed by individual architectural elements. These parameters cannot be measured explicitly.
(2) Operating conditions: VCCIO = 3.3 V ±10% for commercial or industrial use.
(3) Operating conditions: VCCIO = 2.5 V ±5% for commercial or industrial use in EPF10K30E, EPF10K50S, EPF10K100E, EPF10K130E, and EPF10K200S devices.
(4) Operating conditions: VCCIO = 3.3 V.
(5) Because the RAM in the EAB is self-timed, this parameter can be ignored when the `WE` signal is registered.
(6) EAB macroparameters are internal parameters that can simplify predicting the behavior of an EAB at its boundary; these parameters are calculated by summing selected microparameters.
(7) These parameters are worst-case values for typical applications. Post-compilation timing simulation and timing analysis are required to determine actual worst-case performance.
(8) Contact Altera Applications for test circuit specifications and test conditions.
(9) This timing parameter is sample-tested only.
(10) This parameter is measured with the measurement and test conditions, including load, specified in the PCI Local Bus Specification, revision 2.2.

Table 34. EPF10K30E Device EAB Internal Timing Macroparameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{EABAA}		6.4		7.6		8.8	ns
$t_{EABRCOMB}$	6.4		7.6		8.8		ns
$t_{EABRCREG}$	4.4		5.1		6.0		ns
t_{EABWP}	2.5		2.9		3.3		ns
$t_{EABWCOMB}$	6.0		7.0		8.0		ns
$t_{EABWCREG}$	6.8		7.8		9.0		ns
t_{EABDD}		5.7		6.7		7.7	ns
$t_{EABDATACO}$		0.8		0.9		1.1	ns
$t_{EABDATASU}$	1.5		1.7		2.0		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	1.3		1.4		1.7		ns
t_{EABWEH}	0.0		0.0		0.0		ns
$t_{EABWDSU}$	1.5		1.7		2.0		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	3.0		3.6		4.3		ns
t_{EABWAH}	0.5		0.5		0.4		ns
t_{EABWO}		5.1		6.0		6.8	ns

Table 45. EPF10K100E Device LE Timing Microparameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{CGENR}		0.1		0.1		0.2	ns
t_{CASC}		0.6		0.9		1.2	ns
t_{C}		0.8		1.0		1.4	ns
t_{CO}		0.6		0.8		1.1	ns
t_{COMB}		0.4		0.5		0.7	ns
t_{SU}	0.4		0.6		0.7		ns
t_{H}	0.5		0.7		0.9		ns
t_{PRE}		0.8		1.0		1.4	ns
t_{CLR}		0.8		1.0		1.4	ns
t_{CH}	1.5		2.0		2.5		ns
t_{CL}	1.5		2.0		2.5		ns

Table 46. EPF10K100E 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}		1.7		2.0		2.6	ns
t_{IOC}		0.0		0.0		0.0	ns
t_{IOCO}		1.4		1.6		2.1	ns
t_{IOCOMB}		0.5		0.7		0.9	ns
t_{IOSU}	0.8		1.0		1.3		ns
t_{IOH}	0.7		0.9		1.2		ns
t_{IOCLR}		0.5		0.7		0.9	ns
t_{OD1}		3.0		4.2		5.6	ns
t_{OD2}		3.0		4.2		5.6	ns
t_{OD3}		4.0		5.5		7.3	ns
t_{XZ}		3.5		4.6		6.1	ns
t_{ZX1}		3.5		4.6		6.1	ns
t_{ZX2}		3.5		4.6		6.1	ns
t_{ZX3}		4.5		5.9		7.8	ns
t_{INREG}		2.0		2.6		3.5	ns
t_{IOFD}		0.5		0.8		1.2	ns
t_{INCOMB}		0.5		0.8		1.2	ns

Table 61. EPF10K200E Device EAB Internal Microparameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		2.0		2.4		3.2	ns
$t_{EABDATA1}$		0.4		0.5		0.6	ns
t_{EABWE1}		1.4		1.7		2.3	ns
t_{EABWE2}		0.0		0.0		0.0	ns
t_{EABRE1}		0		0		0	ns
t_{EABRE2}		0.4		0.5		0.6	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		0.8		0.9		1.2	ns
$t_{EABBYPASS}$		0.0		0.1		0.1	ns
t_{EABSU}	0.9		1.1		1.5		ns
t_{EABH}	0.4		0.5		0.6		ns
t_{EABCLR}	0.8		0.9		1.2		ns
t_{AA}		3.1		3.7		4.9	ns
t_{WP}	3.3		4.0		5.3		ns
t_{RP}	0.9		1.1		1.5		ns
t_{WDSU}	0.9		1.1		1.5		ns
t_{WDH}	0.1		0.1		0.1		ns
t_{WASU}	1.3		1.6		2.1		ns
t_{WAH}	2.1		2.5		3.3		ns
t_{RASU}	2.2		2.6		3.5		ns
t_{RAH}	0.1		0.1		0.2		ns
t_{WO}		2.0		2.4		3.2	ns
t_{DD}		2.0		2.4		3.2	ns
t_{EABOUT}		0.0		0.1		0.1	ns
t_{EABCH}	1.5		2.0		2.5		ns
t_{EABCL}	3.3		4.0		5.3		ns

Table 62. EPF10K200E Device EAB Internal Timing Macroparameters (Part 1 of 2) *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{EABAA}		5.1		6.4		8.4	ns
$t_{EABRCOMB}$	5.1		6.4		8.4		ns
$t_{EABRCREG}$	4.8		5.7		7.6		ns
t_{EABWP}	3.3		4.0		5.3		ns

Table 74. EPF10K200S Device IOE Timing Microparameters (Part 2 of 2) *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{ZX2}		4.5		4.8		6.6	ns
t_{ZX3}		6.6		7.6		10.1	ns
t_{INREG}		3.7		5.7		7.7	ns
t_{IOFD}		1.8		3.4		4.0	ns
t_{INCOMB}		1.8		3.4		4.0	ns

Table 75. EPF10K200S 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		2.4		3.2	ns
$t_{EABDATA1}$		0.4		0.5		0.6	ns
t_{EABWE1}		1.1		1.7		2.3	ns
t_{EABWE2}		0.0		0.0		0.0	ns
t_{EABRE1}		0		0		0	ns
t_{EABRE2}		0.4		0.5		0.6	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		0.8		0.9		1.2	ns
$t_{EABBYPASS}$		0.0		0.1		0.1	ns
t_{EABSU}	0.7		1.1		1.5		ns
t_{EABH}	0.4		0.5		0.6		ns
t_{EABCLR}	0.8		0.9		1.2		ns
t_{AA}		2.1		3.7		4.9	ns
t_{WP}	2.1		4.0		5.3		ns
t_{RP}	1.1		1.1		1.5		ns
t_{WDSU}	0.5		1.1		1.5		ns
t_{WDH}	0.1		0.1		0.1		ns
t_{WASU}	1.1		1.6		2.1		ns
t_{WAH}	1.6		2.5		3.3		ns
t_{RASU}	1.6		2.6		3.5		ns
t_{RAH}	0.1		0.1		0.2		ns
t_{WO}		2.0		2.4		3.2	ns
t_{DD}		2.0		2.4		3.2	ns
t_{EABOUT}		0.0		0.1		0.1	ns
t_{EABCH}	1.5		2.0		2.5		ns
t_{EABCL}	2.1		2.8		3.8		ns

Table 77. EPF10K200S Device Interconnect Timing Microparameters (Part 2 of 2) Note (1)

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{LABCASC}$		0.5		1.0		1.4	ns

Table 78. EPF10K200S External Timing Parameters Note (1)

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{DRR}		9.0		12.0		16.0	ns
t_{INSU} (2)	3.1		3.7		4.7		ns
t_{INH} (2)	0.0		0.0		0.0		ns
t_{OUTCO} (2)	2.0	3.7	2.0	4.4	2.0	6.3	ns
t_{INSU}(3)	2.1		2.7		–		ns
t_{INH} (3)	0.0		0.0		–		ns
t_{OUTCO}(3)	0.5	2.7	0.5	3.4	–	–	ns
t_{PCISU}	3.0		4.2		–		ns
t_{PCIH}	0.0		0.0		–		ns
t_{PCICO}	2.0	6.0	2.0	8.9	–	–	ns

Table 79. EPF10K200S External Bidirectional Timing Parameters Note (1)

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$ (2)	2.3		3.4		4.4		ns
$t_{INHBIDIR}$ (2)	0.0		0.0		0.0		ns
$t_{INSUBIDIR}$ (3)	3.3		4.4		–		ns
$t_{INHBIDIR}$ (3)	0.0		0.0		–		ns
$t_{OUTCOBIDIR}$ (2)	2.0	3.7	2.0	4.4	2.0	6.3	ns
$t_{XZBIDIR}$ (2)		6.9		7.6		9.2	ns
$t_{ZXBIDIR}$ (2)		5.9		6.6		–	ns
$t_{OUTCOBIDIR}$ (3)	0.5	2.7	0.5	3.4	–	–	ns
$t_{XZBIDIR}$ (3)		6.9		7.6		9.2	ns
$t_{ZXBIDIR}$ (3)		5.9		6.6		–	ns

Notes to tables:

(1) All timing parameters are described in Tables 24 through 30 in this data sheet.
(2) This parameter is measured without the use of the ClockLock or ClockBoost circuits.
(3) This parameter is measured with the use of the ClockLock or ClockBoost circuits.

Power Consumption

The supply power (P) for FLEX 10KE devices can be calculated with the following equation:

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

The $I_{CCACTIVE}$ value depends on the switching frequency and the application logic. This value is calculated 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)*.

Compared to the rest of the device, the embedded array consumes a negligible amount of power. Therefore, the embedded array can be ignored when calculating supply current.

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 LEs used in the device
$\mathbf{tog_{LC}}$ = Average percent of LEs toggling at each clock (typically 12.5%)
K = Constant

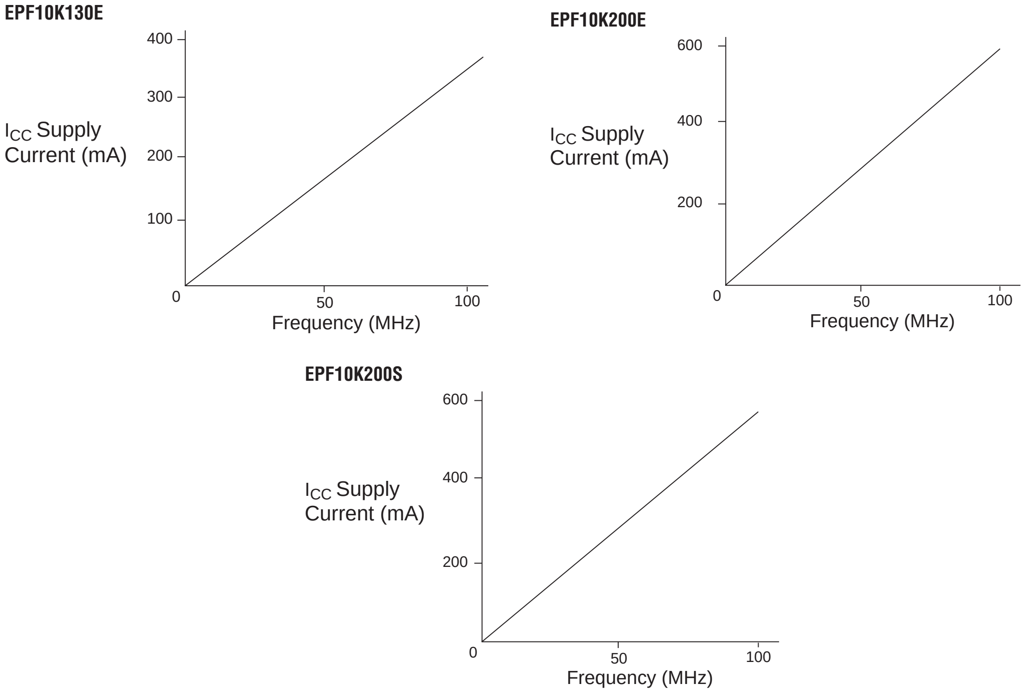
Table 80 provides the constant (K) values for FLEX 10KE devices.

Table 80. FLEX 10KE K Constant Values

Device	K Value
EPF10K30E	4.5
EPF10K50E	4.8
EPF10K50S	4.5
EPF10K100E	4.5
EPF10K130E	4.6
EPF10K200E	4.8
EPF10K200S	4.6

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.

Figure 31. FLEX 10KE $I_{CCACTIVE}$ vs. Operating Frequency (Part 2 of 2)



Configuration & Operation

The FLEX 10KE architecture supports several configuration schemes. This section summarizes the device operating modes and available device configuration schemes.

Operating Modes

The FLEX 10KE architecture uses SRAM configuration elements that require configuration data to be loaded every time the circuit powers up. The process of physically loading the SRAM data into the device is called *configuration*. Before configuration, as V_{CC} rises, the device initiates a Power-On Reset (POR). This POR event clears the device and prepares it for configuration. The FLEX 10KE POR time does not exceed 50 µs.

When configuring with a configuration device, refer to the respective configuration device data sheet for POR timing information.

During initialization, which 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. Together, the configuration and initialization processes are called *command mode*; normal device operation is called *user mode*.

SRAM configuration elements allow FLEX 10KE 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 85 ms and can be used to reconfigure an entire system dynamically. In-field upgrades can be performed by distributing new configuration files.

Before and during configuration, all I/O pins (except dedicated inputs, clock, or configuration pins) are pulled high by a weak pull-up resistor.

Programming Files

Despite being function- and pin-compatible, FLEX 10KE devices are not programming- or configuration file-compatible with FLEX 10K or FLEX 10KA devices. A design therefore must be recompiled before it is transferred from a FLEX 10K or FLEX 10KA device to an equivalent FLEX 10KE device. This recompilation should be performed both to create a new programming or configuration file and to check design timing in FLEX 10KE devices, which has different timing characteristics than FLEX 10K or FLEX 10KA devices.

FLEX 10KE devices are generally pin-compatible with equivalent FLEX 10KA devices. In some cases, FLEX 10KE devices have fewer I/O pins than the equivalent FLEX 10KA devices. Table 81 shows which FLEX 10KE devices have fewer I/O pins than equivalent FLEX 10KA devices. However, power, ground, JTAG, and configuration pins are the same on FLEX 10KA and FLEX 10KE devices, enabling migration from a FLEX 10KA design to a FLEX 10KE design.