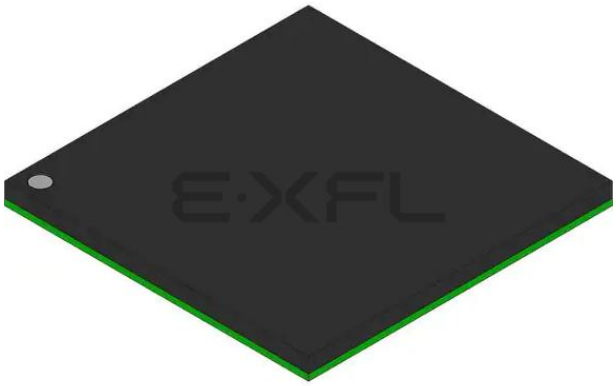
Altera - EPF10K50VFC484-3N Datasheet



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	360
Number of Logic Elements/Cells	2880
Total RAM Bits	40960
Number of I/O	291
Number of Gates	116000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	484-BBGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=epf10k50vfc484-3n

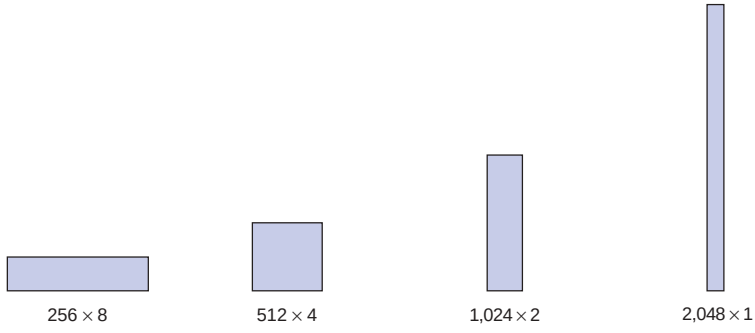
Logic functions are implemented by programming the EAB with a read-only pattern during configuration, creating a large LUT. With LUTs, combinatorial functions are implemented by looking up the results, rather than by computing them. This implementation of combinatorial functions can be faster than using algorithms implemented in general logic, a performance advantage that is further enhanced by the fast access times of EABs. The large capacity of EABs enables designers to implement complex functions in one logic level without the routing delays associated with linked LEs or field-programmable gate array (FPGA) RAM blocks. For example, a single EAB can implement a 4 × 4 multiplier with eight inputs and eight outputs. Parameterized functions such as LPM functions can automatically take advantage of the EAB.

The EAB provides advantages over FPGAs, which implement on-board RAM as arrays of small, distributed RAM blocks. These FPGA RAM blocks contain delays that are less predictable as the size of the RAM increases. In addition, FPGA RAM blocks are prone to routing problems because small blocks of RAM must be connected together to make larger blocks. In contrast, EABs can be used to implement large, dedicated blocks of RAM that eliminate these timing and routing concerns.

EABs can be used to implement synchronous RAM, which is easier to use than asynchronous RAM. A circuit using asynchronous RAM must generate the RAM write enable (WE) signal, while ensuring that its data and address signals meet setup and hold time specifications relative to the WE signal. In contrast, the EAB's synchronous RAM generates its own WE signal and is self-timed with respect to the global clock. A circuit using the EAB's self-timed RAM need only meet the setup and hold time specifications of the global clock.

When used as RAM, each EAB can be configured in any of the following sizes: 256 × 8, 512 × 4, 1,024 × 2, or 2,048 × 1. See Figure 2.

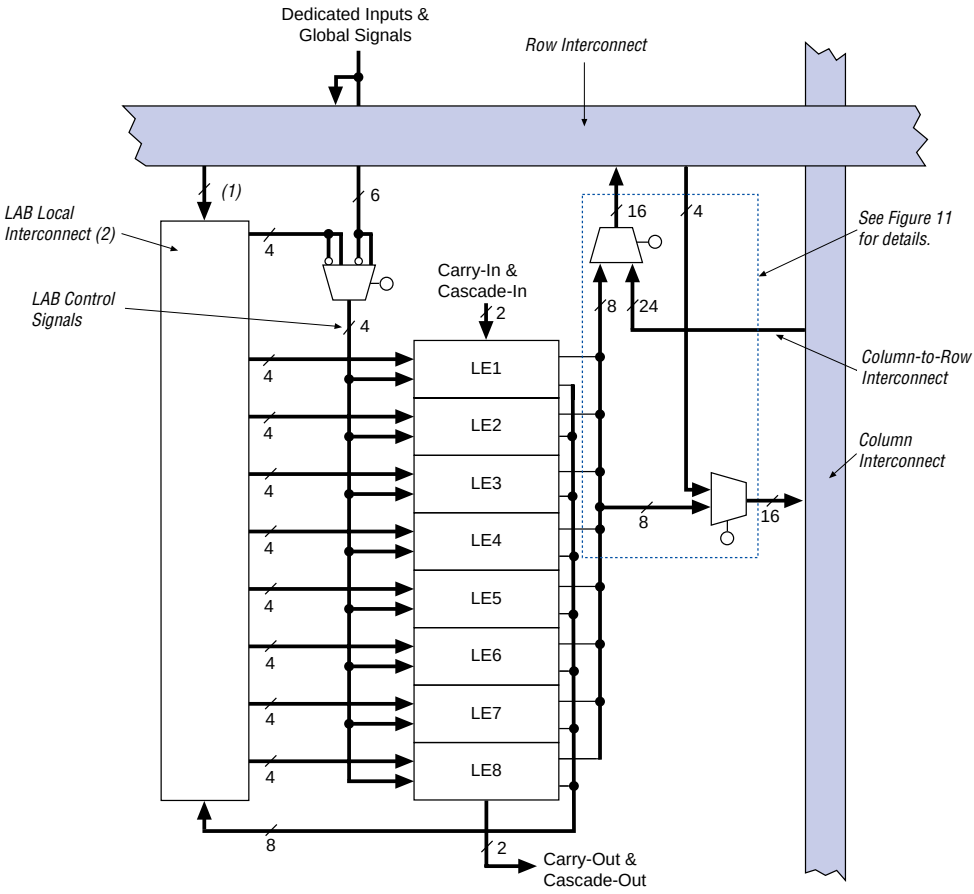
Figure 2. EAB Memory Configurations



Logic Array Block

Each LAB consists of eight LEs, their associated carry and cascade chains, LAB control signals, and the LAB local interconnect. The LAB provides the coarse-grained structure to the FLEX 10K architecture, facilitating efficient routing with optimum device utilization and high performance. See Figure 5.

Figure 5. FLEX 10K LAB



Notes:

(1) EPF10K10, EPF10K10A, EPF10K20, EPF10K30, EPF10K30A, EPF10K40, EPF10K50, and EPF10K50V devices have 22 inputs to the LAB local interconnect channel from the row; EPF10K70, EPF10K100, EPF10K100A, EPF10K130V, and EPF10K250A devices have 26.

(2) EPF10K10, EPF10K10A, EPF10K20, EPF10K30, EPF10K30A, EPF10K40, EPF10K50, and EPF10K50V devices have 30 LAB local interconnect channels; EPF10K70, EPF10K100, EPF10K100A, EPF10K130V, and EPF10K250A devices have 34 LABs.

Up/Down Counter Mode

The up/down counter mode offers counter enable, clock enable, synchronous up/down control, and data loading options. These control signals are generated by the data inputs from the LAB local interconnect, the carry-in signal, and output feedback from the programmable register. The Up/down counter mode uses 2 three-input LUTs: one generates the counter data, and the other generates the fast carry bit. A 2-to-1 multiplexer provides synchronous loading. Data can also be loaded asynchronously with the clear and preset register control signals, without using the LUT resources.

Clearable Counter Mode

The clearable counter mode is similar to the up/down counter mode, but supports a synchronous clear instead of the up/down control. The clear function is substituted for the cascade-in signal in the up/down counter mode. Clearable counter mode uses 2 three-input LUTs: one generates the counter data, and the other generates the fast carry bit. Synchronous loading is provided by a 2-to-1 multiplexer. The output of this multiplexer is ANDed with a synchronous clear signal.

Internal Tri-State Emulation

Internal tri-state emulation provides internal tri-stating without the limitations of a physical tri-state bus. In a physical tri-state bus, the tri-state buffers' output enable (OE) signals select which signal drives the bus. However, if multiple OE signals are active, contending signals can be driven onto the bus. Conversely, if no OE signals are active, the bus will float. Internal tri-state emulation resolves contending tri-state buffers to a low value and floating buses to a high value, thereby eliminating these problems. The Altera software automatically implements tri-state bus functionality with a multiplexer.

Clear & Preset Logic Control

Logic for the programmable register's clear and preset functions is controlled by the DATA3, LABCTRL1, and LABCTRL2 inputs to the LE. The clear and preset control structure of the LE asynchronously loads signals into a register. Either LABCTRL1 or LABCTRL2 can control the asynchronous clear. Alternatively, the register can be set up so that LABCTRL1 implements an asynchronous load. The data to be loaded is driven to DATA3; when LABCTRL1 is asserted, DATA3 is loaded into the register.

During compilation, the Compiler automatically selects the best control signal implementation. Because the clear and preset functions are active-low, the Compiler automatically assigns a logic high to an unused clear or preset.

The clear and preset logic is implemented in one of the following six modes chosen during design entry:

- Asynchronous clear
- Asynchronous preset
- Asynchronous clear and preset
- Asynchronous load with clear
- Asynchronous load with preset
- Asynchronous load without clear or preset

In addition to the six clear and preset modes, FLEX 10K devices provide a chip-wide reset pin that can reset all registers in the device. Use of this feature is set during design entry. In any of the clear and preset modes, the chip-wide reset overrides all other signals. Registers with asynchronous presets may be preset when the chip-wide reset is asserted. Inversion can be used to implement the asynchronous preset. Figure 10 shows examples of how to enter a section of a design for the desired functionality.

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

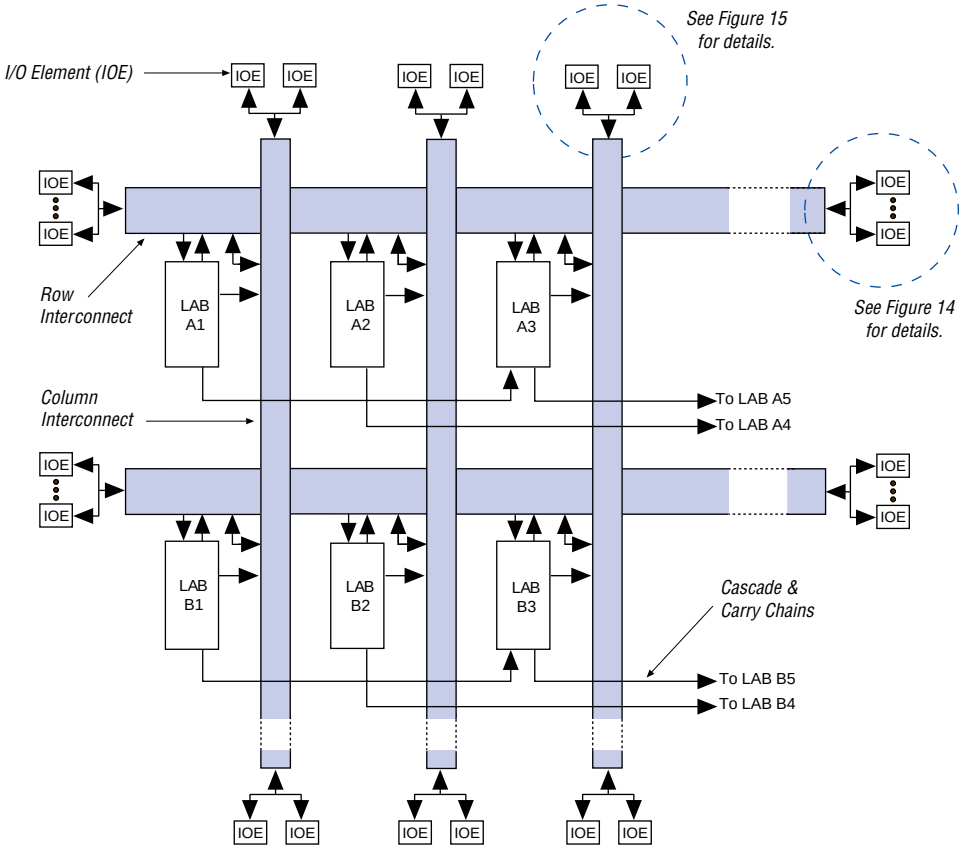
When implementing an asynchronous load in conjunction with preset, the Altera software provides preset control by using the clear and inverting the input and output of the register. Asserting LABCTRL2 presets the register, while asserting LABCTRL1 loads the register. The Altera software inverts the signal 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.

Figure 12 shows the interconnection of adjacent LABs and EABs with row, column, and local interconnects, as well as the associated cascade and carry chains. Each LAB is labeled according to its location: a letter represents the row and a number represents the column. For example, LAB B3 is in row B, column 3.

Figure 12. Interconnect Resources

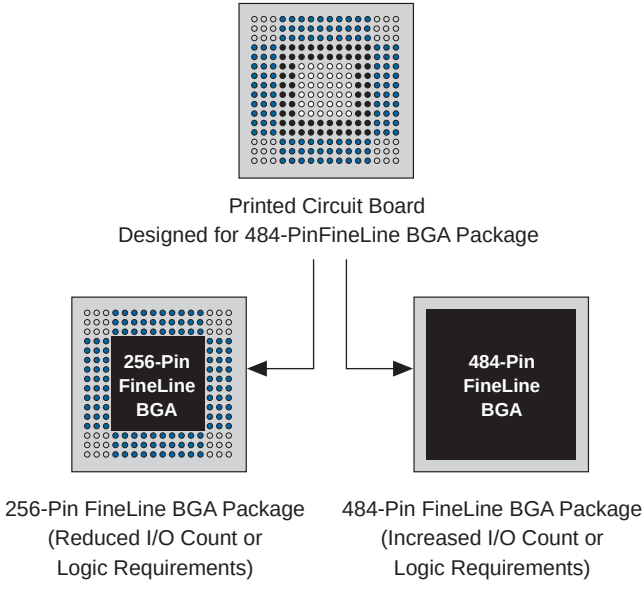


SameFrame Pin-Outs

FLEX 10KE devices support the SameFrame pin-out feature for FineLine BGA packages. The SameFrame pin-out feature is the arrangement of balls on FineLine BGA packages such that the lower-ball-count packages form a subset of the higher-ball-count packages. SameFrame pin-outs provide the flexibility to migrate not only from device to device within the same package, but also from one package to another. A given printed circuit board (PCB) layout can support multiple device density/package combinations. For example, a single board layout can support a range of devices from an EPF10K10A device in a 256-pin FineLine BGA package to an EPF10K100A device in a 484-pin FineLine BGA package.

The Altera software provides support to design PCBs with SameFrame pin-out devices. Devices can be defined for present and future use. The Altera software generates pin-outs describing how to lay out a board to take advantage of this migration (see Figure 16).

Figure 16. SameFrame Pin-Out Example



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.

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 28. FLEX 10KA 3.3-V Device DC Operating Conditions *Notes (6), (7)*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	High-level input voltage		1.7 or $0.5 \times V_{CCINT}$, whichever is lower		5.75	V
V_{IL}	Low-level input voltage		–0.5		$0.3 \times V_{CCINT}$	V
V_{OH}	3.3-V high-level TTL output voltage	I_{OH} = –11 mA DC, V_{CCIO} = 3.00 V *(8)*	2.4			V
	3.3-V high-level CMOS output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 3.00 V *(8)*	V_{CCIO} – 0.2			V
	3.3-V high-level PCI output voltage	I_{OH} = –0.5 mA DC, V_{CCIO} = 3.00 to 3.60 V *(8)*	$0.9 \times V_{CCIO}$			V
	2.5-V high-level output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 2.30 V *(8)*	2.1			V
		I_{OH} = –1 mA DC, V_{CCIO} = 2.30 V *(8)*	2.0			V
		I_{OH} = –2 mA DC, V_{CCIO} = 2.30 V *(8)*	1.7			V
V_{OL}	3.3-V low-level TTL output voltage	I_{OL} = 9 mA DC, V_{CCIO} = 3.00 V *(9)*			0.45	V
	3.3-V low-level CMOS output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 3.00 V *(9)*			0.2	V
	3.3-V low-level PCI output voltage	I_{OL} = 1.5 mA DC, V_{CCIO} = 3.00 to 3.60 V *(9)*			$0.1 \times V_{CCIO}$	V
	2.5-V low-level output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 2.30 V *(9)*			0.2	V
		I_{OL} = 1 mA DC, V_{CCIO} = 2.30 V *(9)*			0.4	V
		I_{OL} = 2 mA DC, V_{CCIO} = 2.30 V *(9)*			0.7	V
I_I	Input pin leakage current	V_I = 5.3 V to –0.3 V *(10)*	–10		10	µA
I_{OZ}	Tri-stated I/O pin leakage current	V_O = 5.3 V to –0.3 V *(10)*	–10		10	µA
I_{CC0}	V_{CC} supply current (standby)	V_I = ground, no load		0.3	10	mA
		V_I = ground, no load *(11)*		10		mA

Table 52. EPF10K30 Device Interconnect Timing Microparameters *Note (1)*

Symbol	-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	
$t_{DIN2IOE}$		6.9		8.7	ns
t_{DIN2LE}		3.6		4.8	ns
$t_{DIN2DATA}$		5.5		7.2	ns
$t_{DCLK2IOE}$		4.6		6.2	ns
$t_{DCLK2LE}$		3.6		4.8	ns
$t_{SAMELAB}$		0.3		0.3	ns
$t_{SAMEROW}$		3.3		3.7	ns
$t_{SAMECOLUMN}$		2.5		2.7	ns
$t_{DIFFROW}$		5.8		6.4	ns
$t_{TWOROWS}$		9.1		10.1	ns
$t_{LEPERIPH}$		6.2		7.1	ns
$t_{LABCARRY}$		0.4		0.6	ns
$t_{LABCASC}$		2.4		3.0	ns

Table 53. EPF10K40 Device Interconnect Timing Microparameters *Note (1)*

Symbol	-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	
$t_{DIN2IOE}$		7.6		9.4	ns
t_{DIN2LE}		3.6		4.8	ns
$t_{DIN2DATA}$		5.5		7.2	ns
$t_{DCLK2IOE}$		4.6		6.2	ns
$t_{DCLK2LE}$		3.6		4.8	ns
$t_{SAMELAB}$		0.3		0.3	ns
$t_{SAMEROW}$		3.3		3.7	ns
$t_{SAMECOLUMN}$		3.1		3.2	ns
$t_{DIFFROW}$		6.4		6.4	ns
$t_{TWOROWS}$		9.7		10.6	ns
$t_{LEPERIPH}$		6.4		7.1	ns
$t_{LABCARRY}$		0.4		0.6	ns
$t_{LABCASC}$		2.4		3.0	ns

Table 68. EPF10K100 Device Interconnect Timing Microparameters *Note (1)*

Symbol	-3DX Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		10.3		10.3		12.2	ns
t_{DIN2LE}		4.8		4.8		6.0	ns
$t_{DIN2DATA}$		7.3		7.3		11.0	ns
$t_{DCLK2IOE}$ without ClockLock or ClockBoost circuitry		6.2		6.2		7.7	ns
$t_{DCLK2IOE}$ with ClockLock or ClockBoost circuitry		2.3		–		–	ns
$t_{DCLK2LE}$ without ClockLock or ClockBoost circuitry		4.8		4.8		6.0	ns
$t_{DCLK2LE}$ with ClockLock or ClockBoost circuitry		2.3		–		–	ns
$t_{SAMELAB}$		0.4		0.4		0.5	ns
$t_{SAMEROW}$		4.9		4.9		5.5	ns
$t_{SAMECOLUMN}$		5.1		5.1		5.4	ns
$t_{DIFFROW}$		10.0		10.0		10.9	ns
$t_{TWOROWS}$		14.9		14.9		16.4	ns
$t_{LEPERIPH}$		6.9		6.9		8.1	ns
$t_{LABCARRY}$		0.9		0.9		1.1	ns
$t_{LABCASC}$		3.0		3.0		3.2	ns

Table 74. EPF10K50V Device EAB Internal Timing Macroparameters *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_{EABAA}		9.5		13.6		16.5		20.8	ns
$t_{EABRCCOMB}$	9.5		13.6		16.5		20.8		ns
$t_{EABRCREG}$	6.1		8.8		10.8		13.4		ns
t_{EABWP}	6.0		4.9		6.0		7.4		ns
$t_{EABWCCOMB}$	6.2		6.1		7.5		9.2		ns
$t_{EABWCREG}$	12.0		11.6		14.2		17.4		ns
t_{EABDD}		6.8		9.7		11.8		14.9	ns
$t_{EABDATACO}$		1.0		1.4		1.8		2.2	ns
$t_{EABDATASU}$	5.3		4.6		5.6		6.9		ns
$t_{EABDATAH}$	0.0		0.0		0.0		0.0		ns
$t_{EABWESU}$	4.4		4.8		5.8		7.2		ns
t_{EABWEH}	0.0		0.0		0.0		0.0		ns
$t_{EABWDSU}$	1.8		1.1		1.4		2.1		ns
t_{EABWDH}	0.0		0.0		0.0		0.0		ns
$t_{EABWASU}$	4.5		4.6		5.6		7.4		ns
t_{EABWAH}	0.0		0.0		0.0		0.0		ns
t_{EABWO}		5.1		9.4		11.4		14.0	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 78 through 84 show EPF10K130V device internal and external timing parameters.

Table 78. EPF10K130V 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		1.8		2.3	ns
t_{CLUT}		0.5		0.7		0.9	ns
t_{RLUT}		1.2		1.7		2.2	ns
t_{PACKED}		0.5		0.6		0.7	ns
t_{EN}		0.6		0.8		1.0	ns
t_{CICO}		0.2		0.3		0.4	ns
t_{CGEN}		0.3		0.4		0.5	ns
t_{CGENR}		0.7		1.0		1.3	ns
t_{CASC}		0.9		1.2		1.5	ns
t_{C}		1.9		2.4		3.0	ns
t_{CO}		0.6		0.9		1.1	ns
t_{COMB}		0.5		0.7		0.9	ns
t_{SU}	0.2		0.2		0.3		ns
t_{H}	0.0		0.0		0.0		ns
t_{PRE}		2.4		3.1		3.9	ns
t_{CLR}		2.4		3.1		3.9	ns
t_{CH}	4.0		4.0		4.0		ns
t_{CL}	4.0		4.0		4.0		ns

Table 79. EPF10K130V Device IOE Timing Microparameters *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{IOD}		1.3		1.6		2.0	ns
t_{IOC}		0.4		0.5		0.7	ns
t_{IOCO}		0.3		0.4		0.5	ns
t_{IOCOMB}		0.0		0.0		0.0	ns
t_{IOSU}	2.6		3.3		3.8		ns
t_{IOH}	0.0		0.0		0.0		ns
t_{IOCLR}		1.7		2.2		2.7	ns
t_{OD1}		3.5		4.4		5.0	ns
t_{OD2}		–		–		–	ns
t_{OD3}		8.2		8.1		9.7	ns
t_{XZ}		4.9		6.3		7.4	ns
t_{ZX1}		4.9		6.3		7.4	ns
t_{ZX2}		–		–		–	ns
t_{ZX3}		9.6		10.0		12.1	ns
t_{INREG}		7.9		10.0		12.6	ns
t_{IOFD}		6.2		7.9		9.9	ns
t_{INCOMB}		6.2		7.9		9.9	ns

Table 89. EPF10K10A Device Interconnect Timing Microparameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		4.2		5.0		6.5	ns
t_{DIN2LE}		2.2		2.6		3.4	ns
$t_{DIN2DATA}$		4.3		5.2		7.1	ns
$t_{DCLK2IOE}$		4.2		4.9		6.6	ns
$t_{DCLK2LE}$		2.2		2.6		3.4	ns
$t_{SAMELAB}$		0.1		0.1		0.2	ns
$t_{SAMEROW}$		2.2		2.4		2.9	ns
$t_{SAMECOLUMN}$		0.8		1.0		1.4	ns
$t_{DIFFROW}$		3.0		3.4		4.3	ns
$t_{TWOROWS}$		5.2		5.8		7.2	ns
$t_{LEPERIPH}$		1.8		2.2		2.8	ns
$t_{LABCARRY}$		0.5		0.5		0.7	ns
$t_{LABCASC}$		0.9		1.0		1.5	ns

Table 90. EPF10K10A External Reference Timing Parameters *Note (1)*

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{DRR}		10.0		12.0		16.0	ns
t_{INSU} *(2)*, *(3)*	1.6		2.1		2.8		ns
t_{INH} *(3)*	0.0		0.0		0.0		ns
t_{OUTCO} *(3)*	2.0	5.8	2.0	6.9	2.0	9.2	ns

Table 91. EPF10K10A Device External Bidirectional Timing Parameters *Note (1)*

Symbol	-2 Speed Grade		-3 Speed Grade		-4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	2.4		3.3		4.5		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$	2.0	5.8	2.0	6.9	2.0	9.2	ns
$t_{XZBIDIR}$		6.3		7.5		9.9	ns
$t_{ZXBIDIR}$		6.3		7.5		9.9	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 92 through 98 show EPF10K30A device internal and external timing parameters.

Table 92. EPF10K30A 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}		0.8		1.1		1.5	ns
t_{CLUT}		0.6		0.7		1.0	ns
t_{RLUT}		1.2		1.5		2.0	ns
t_{PACKED}		0.6		0.6		1.0	ns
t_{EN}		1.3		1.5		2.0	ns
t_{CICO}		0.2		0.3		0.4	ns
t_{CGEN}		0.8		1.0		1.3	ns
t_{CGENR}		0.6		0.8		1.0	ns
t_{CASC}		0.9		1.1		1.4	ns
t_{C}		1.1		1.3		1.7	ns
t_{CO}		0.4		0.6		0.7	ns
t_{COMB}		0.6		0.7		0.9	ns
t_{SU}	0.9		0.9		1.4		ns
t_{H}	1.1		1.3		1.7		ns
t_{PRE}		0.5		0.6		0.8	ns
t_{CLR}		0.5		0.6		0.8	ns
t_{CH}	3.0		3.5		4.0		ns
t_{CL}	3.0		3.5		4.0		ns

Table 93. EPF10K30A Device IOE Timing Microparameters *Note (1)* **(Part 1 of 2)**

Symbol	-1 Speed Grade		-2 Speed Grade		-3 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{IOD}		2.2		2.6		3.4	ns
t_{IOC}		0.3		0.3		0.5	ns
t_{IOCO}		0.2		0.2		0.3	ns
t_{IOCOMB}		0.5		0.6		0.8	ns
t_{IOSU}	1.4		1.7		2.2		ns

Table 95. EPF10K30A 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}		9.7		11.6		16.2	ns
$t_{EABRCCOMB}$	9.7		11.6		16.2		ns
$t_{EABRCREG}$	5.9		7.1		9.7		ns
t_{EABWP}	3.8		4.5		5.9		ns
$t_{EABWCCOMB}$	4.0		4.7		6.3		ns
$t_{EABWCREG}$	9.8		11.6		16.6		ns
t_{EABDD}		9.2		11.0		16.1	ns
$t_{EABDATACO}$		1.7		2.1		3.4	ns
$t_{EABDATASU}$	2.3		2.7		3.5		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	3.3		3.9		4.9		ns
t_{EABWEH}	0.0		0.0		0.0		ns
$t_{EABWDSU}$	3.2		3.8		5.0		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	3.7		4.4		5.1		ns
t_{EABWAH}	0.0		0.0		0.0		ns
t_{EABWO}		6.1		7.3		11.3	ns

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.

Notes: