# Intel - EPF10K50VRC240-3N 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 | Obsolete |
| Number of LABs/CLBs | 360 |
| Number of Logic Elements/Cells | 2880 |
| Total RAM Bits | 20480 |
| Number of I/O | 189 |
| Number of Gates | 116000 |
| Voltage - Supply | 3V ~ 3.6V |
| Mounting Type | Surface Mount |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Package / Case | 240-BFQFP Exposed Pad |
| Supplier Device Package | 240-RQFP (32x32) |
| Purchase URL | https://www.e-xfl.com/product-detail/intel/epf10k50vrc240-3n |

*Table 4. FLEX 10K Package Options & I/O Pin Count* *Note (1)*

| Device | 84-Pin PLCC | 100-Pin TQFP | 144-Pin TQFP | 208-Pin PQFP RQFP | 240-Pin PQFP RQFP |
|---|---|---|---|---|---|
| EPF10K10 | 59 | | 102 | 134 | |
| EPF10K10A | | 66 | 102 | 134 | |
| EPF10K20 | | | 102 | 147 | 189 |
| EPF10K30 | | | | 147 | 189 |
| EPF10K30A | | | 102 | 147 | 189 |
| EPF10K40 | | | | 147 | 189 |
| EPF10K50 | | | | | 189 |
| EPF10K50V | | | | | 189 |
| EPF10K70 | | | | | 189 |
| EPF10K100 | | | | | |
| EPF10K100A | | | | | 189 |
| EPF10K130V | | | | | |
| EPF10K250A | | | | | |

*Table 5. FLEX 10K Package Options & I/O Pin Count (Continued)* *Note (1)*

| Device | 503-Pin PGA | 599-Pin PGA | 256-Pin FineLine BGA | 356-Pin BGA | 484-Pin FineLine BGA | 600-Pin BGA | 403-Pin PGA |
|---|---|---|---|---|---|---|---|
| EPF10K10 | | | | | | | |
| EPF10K10A | | | 150 | | 150 *(2)* | | |
| EPF10K20 | | | | | | | |
| EPF10K30 | | | | 246 | | | |
| EPF10K30A | | | 191 | 246 | 246 | | |
| EPF10K40 | | | | | | | |
| EPF10K50 | | | | 274 | | | 310 |
| EPF10K50V | | | | 274 | | | |
| EPF10K70 | 358 | | | | | | |
| EPF10K100 | 406 | | | | | | |
| EPF10K100A | | | | 274 | 369 | 406 | |
| EPF10K130V | | 470 | | | | 470 | |
| EPF10K250A | | 470 | | | | 470 | |

*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.

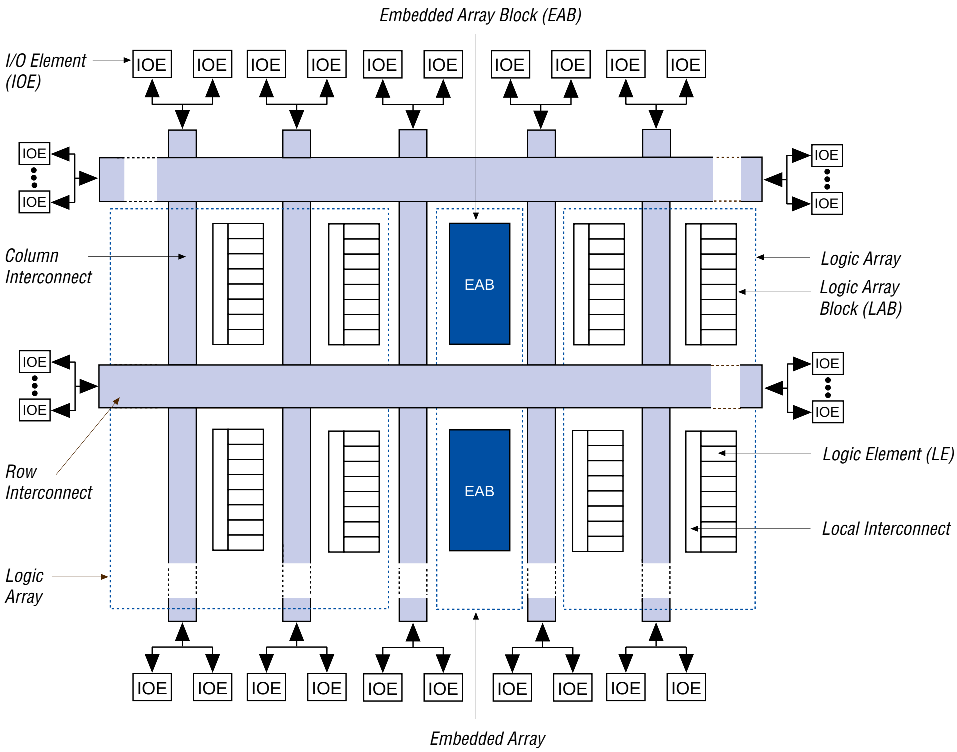
The logic array consists of logic array blocks (LABs). Each LAB contains eight LEs and a local interconnect. An LE consists of a 4-input look-up table (LUT), a programmable flipflop, and dedicated signal paths for carry and cascade functions. The eight LEs can be used to create medium-sized blocks of logic—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 10K devices and to and from device pins are provided by the FastTrack Interconnect, 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. 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 1.6 ns and hold times of 0 ns; as outputs, these registers provide clock-to-output times as low as 5.3 ns. IOEs provide a variety of features, such as JTAG BST support, slew-rate control, tri-state buffers, and open-drain outputs.

Figure 1 shows a block diagram of the FLEX 10K architecture. Each group of LEs is combined into an LAB; LABs are arranged into rows and columns. Each row also contains a single EAB. The LABs and EABs are interconnected by the FastTrack Interconnect. IOEs are located at the end of each row and column of the FastTrack Interconnect.

***Figure 1. FLEX 10K Device Block Diagram***

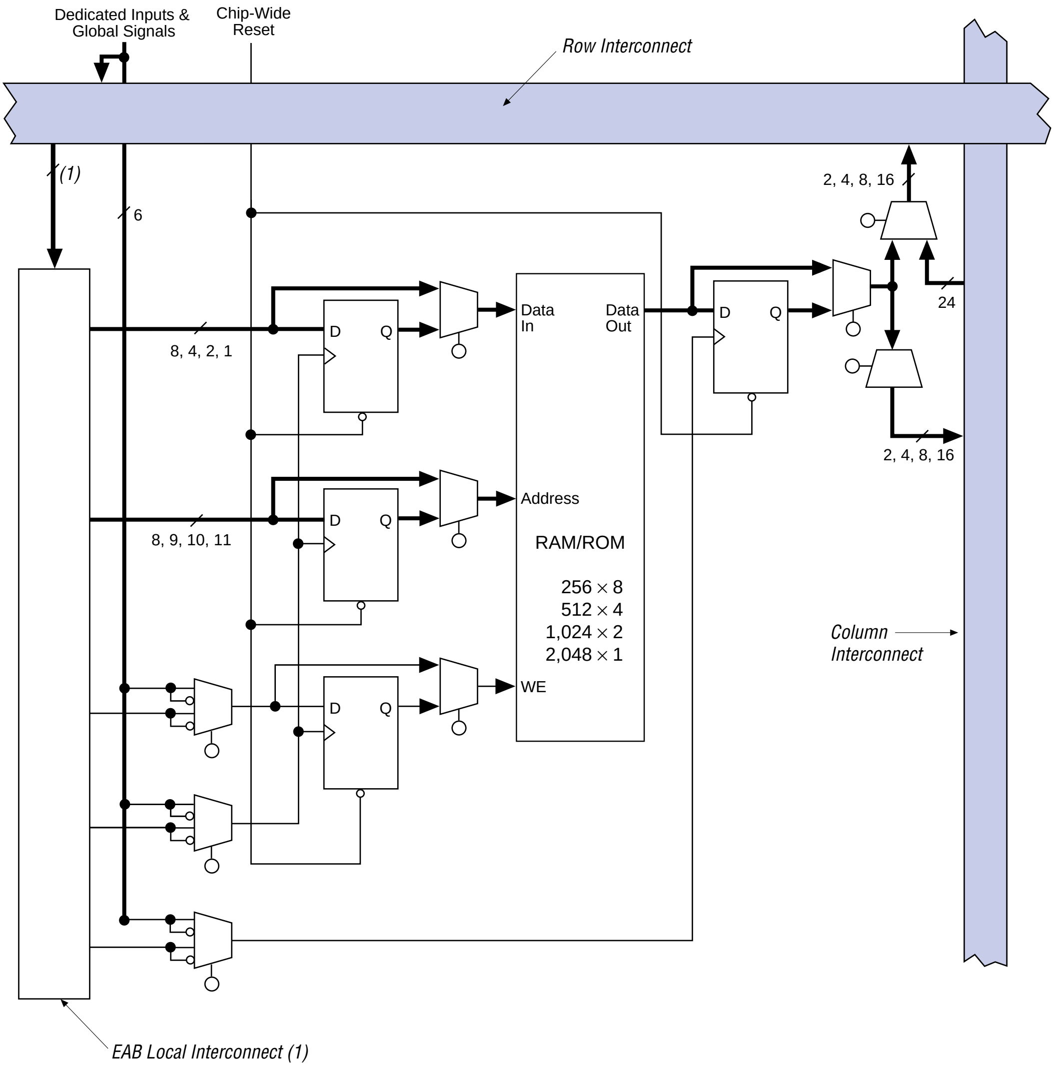



FLEX 10K devices provide six dedicated inputs that drive the flipflops' control inputs to ensure the efficient distribution of high-speed, low-skew (less than 1.5 ns) control signals. These signals use dedicated routing channels that provide shorter delays and lower skews than the FastTrack Interconnect. Four of the dedicated inputs drive four global signals. These four global signals can also be driven by internal logic, providing an ideal solution for a clock divider or an internally generated asynchronous clear signal that clears many registers in the device.

## Embedded Array Block

The EAB is a flexible block of RAM with registers on the input and output ports, and is used to implement common gate array megafunctions. The EAB is also suitable for functions such as multipliers, vector scalars, and error correction circuits, because it is large and flexible. These functions can be combined in applications such as digital filters and microcontrollers.

*Figure 4. FLEX 10K Embedded Array Block*




*Note:*

(1) EPF10K10, EPF10K10A, EPF10K20, EPF10K30, EPF10K30A, EPF10K40, EPF10K50, and EPF10K50V devices have 22 EAB local interconnect channels; EPF10K70, EPF10K100, EPF10K100A, EPF10K130V, and EPF10K250A devices have 26.

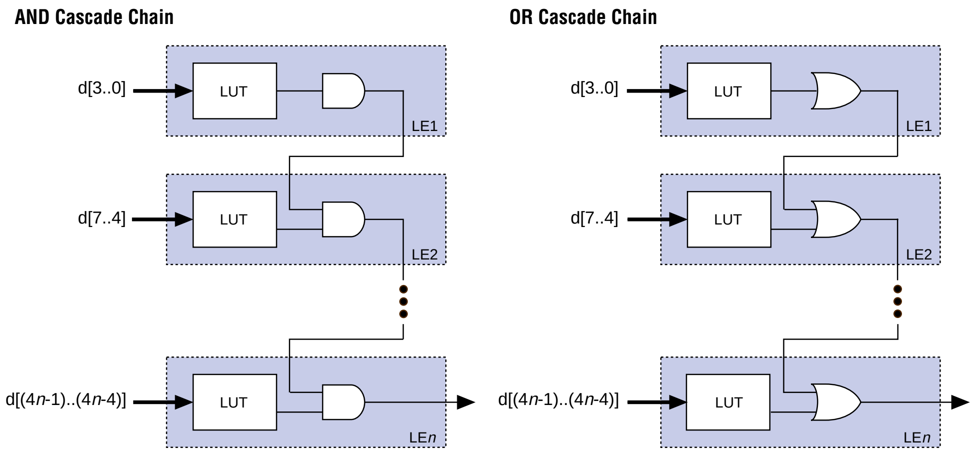
### *Cascade Chain*

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 4*n* 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***



#### 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 `AND`ed 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.

## I/O Element

An I/O element (IOE) contains a bidirectional I/O buffer and a register that can be used either as an input register for external data that requires a fast setup time, or as an output register for data that requires fast clock-to-output performance. In some cases, using an LE register for an input register will result in a faster setup time than using an IOE register. IOEs can be used as input, output, or bidirectional pins. For bidirectional registered I/O implementation, the output register should be in the IOE and, the data input and output enable register should be LE registers placed adjacent to the bidirectional pin. The Compiler uses the programmable inversion option to invert signals from the row and column interconnect automatically where appropriate. Figure 13 shows the bidirectional I/O registers.

Table 12 describes the FLEX 10K device supply voltages and MultiVolt I/O support levels.

| *Table 12. Supply Voltages & MultiVolt I/O Support Levels* | | | | |
|---|---|---|---|---|
| **Devices** | **Supply Voltage (V)** | | **MultiVolt I/O Support Levels (V)** | |
| | **$V_{CCINT}$** | **$V_{CCIO}$** | **Input** | **Output** |
| FLEX 10K *(1)* | 5.0 | 5.0 | 3.3 or 5.0 | 5.0 |
| | 5.0 | 3.3 | 3.3 or 5.0 | 3.3 or 5.0 |
| EPF10K50V *(1)* | 3.3 | 3.3 | 3.3 or 5.0 | 3.3 or 5.0 |
| EPF10K130V | 3.3 | 3.3 | 3.3 or 5.0 | 3.3 or 5.0 |
| FLEX 10KA *(1)* | 3.3 | 3.3 | 2.5, 3.3, or 5.0 | 3.3 or 5.0 |
| | 3.3 | 2.5 | 2.5, 3.3, or 5.0 | 2.5 |

*Note*

(1) 240-pin QFP packages do not support the MultiVolt I/O features, so they do not have separate $V_{CCIO}$ pins.

## Power Sequencing & Hot-Socketing

Because FLEX 10K devices can be used in a multi-voltage environment, they have been designed specifically to tolerate any possible power-up sequence. The $V_{CCIO}$ and $V_{CCINT}$ power supplies can be powered in any order.

Signals can be driven into FLEX 10KA devices before and during power up without damaging the device. Additionally, FLEX 10KA devices do not drive out during power up. Once operating conditions are reached, FLEX 10KA devices operate as specified by the user.

## IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All FLEX 10K devices provide JTAG BST circuitry that complies with the IEEE Std. 1149.1-1990 specification. All FLEX 10K devices can also be configured using the JTAG pins through the BitBlaster serial download cable, or ByteBlasterMV parallel port download cable, or via hardware that uses the Jam™ programming and test language. JTAG BST can be performed before or after configuration, but not during configuration. FLEX 10K devices support the JTAG instructions shown in Table 13.

*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 45. EPF10K10 & EPF10K20 Device External Timing Parameters* *Note (1)*

| Symbol | -3 Speed Grade | | -4 Speed Grade | | Unit |
|---|---|---|---|---|---|
| | Min | Max | Min | Max | |
| $t_{DRR}$ | | 16.1 | | 20.0 | ns |
| $t_{INSU}$ (2), (3) | 5.5 | | 6.0 | | ns |
| $t_{INH}$ (3) | 0.0 | | 0.0 | | ns |
| $t_{OUTCO}$ (3) | 2.0 | 6.7 | 2.0 | 8.4 | ns |

*Table 46. EPF10K10 Device External Bidirectional Timing Parameters* *Note (1)*

| Symbol | -3 Speed Grade | | -4 Speed Grade | | Unit |
|---|---|---|---|---|---|
| | Min | Max | Min | Max | |
| $t_{INSUBIDIR}$ | 4.5 | | 5.6 | | ns |
| $t_{INHBIDIR}$ | 0.0 | | 0.0 | | ns |
| $t_{OUTCOBIDIR}$ | 2.0 | 6.7 | 2.0 | 8.4 | ns |
| $t_{XZBIDIR}$ | | 10.5 | | 13.4 | ns |
| $t_{ZXBIDIR}$ | | 10.5 | | 13.4 | ns |

*Table 47. EPF10K20 Device External Bidirectional Timing Parameters* *Note (1)*

| Symbol | -3 Speed Grade | | -4 Speed Grade | | Unit |
|---|---|---|---|---|---|
| | Min | Max | Min | Max | |
| $t_{INSUBIDIR}$ | 4.6 | | 5.7 | | ns |
| $t_{INHBIDIR}$ | 0.0 | | 0.0 | | ns |
| $t_{OUTCOBIDIR}$ | 2.0 | 6.7 | 2.0 | 8.4 | ns |
| $t_{XZBIDIR}$ | | 10.5 | | 13.4 | ns |
| $t_{ZXBIDIR}$ | | 10.5 | | 13.4 | 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.

*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 |

| Table 65. EPF10K100 Device IOE Timing Microparameters *Note (1)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Symbol | -3DX Speed Grade | | -3 Speed Grade | | -4 Speed Grade | | Unit |
| | Min | Max | Min | Max | Min | Max | |
| $t_{IOD}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{IOC}$ | | 0.5 | | 0.5 | | 0.7 | ns |
| $t_{IOCO}$ | | 0.4 | | 0.4 | | 0.9 | ns |
| $t_{IOCOMB}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{IOSU}$ | 5.5 | | 5.5 | | 6.7 | | ns |
| $t_{IOH}$ | 0.5 | | 0.5 | | 0.7 | | ns |
| $t_{IOCLR}$ | | 0.7 | | 0.7 | | 1.6 | ns |
| $t_{OD1}$ | | 4.0 | | 4.0 | | 5.0 | ns |
| $t_{OD2}$ | | 6.3 | | 6.3 | | 7.3 | ns |
| $t_{OD3}$ | | 7.7 | | 7.7 | | 8.7 | ns |
| $t_{XZ}$ | | 6.2 | | 6.2 | | 6.8 | ns |
| $t_{ZX1}$ | | 6.2 | | 6.2 | | 6.8 | ns |
| $t_{ZX2}$ | | 8.5 | | 8.5 | | 9.1 | ns |
| $t_{ZX3}$ | | 9.9 | | 9.9 | | 10.5 | ns |
| $t_{INREG}$ without ClockLock or ClockBoost circuitry | | 9.0 | | 9.0 | | 10.5 | ns |
| $t_{INREG}$ with ClockLock or ClockBoost circuitry | | 3.0 | | – | | – | ns |
| $t_{IOFD}$ | | 8.1 | | 8.1 | | 10.3 | ns |
| $t_{INCOMB}$ | | 8.1 | | 8.1 | | 10.3 | ns |

**Table 69. EPF10K100 Device External Timing Parameters** *Note (1)*

| Symbol | -3DX Speed Grade | | -3 Speed Grade | | -4 Speed Grade | | Unit |
|---|---|---|---|---|---|---|---|
| | Min | Max | Min | Max | Min | Max | |
| $t_{DRR}$ | | 19.1 | | 19.1 | | 24.2 | ns |
| $t_{INSU}$ *(2), (3), (4)* | 7.8 | | 7.8 | | 8.5 | | ns |
| $t_{OUTCO}$ *(3), (4)* | 2.0 | 11.1 | 2.0 | 11.1 | 2.0 | 14.3 | ns |
| $t_{INH}$ *(3)* | 0.0 | | 0.0 | | 0.0 | | ns |
| $t_{INSU}$ *(2), (3), (5)* | 6.2 | | – | | – | | ns |
| $t_{OUTCO}$ *(3), (5)* | 2.0 | 6.7 | | – | | – | ns |

**Table 70. EPF10K100 Device External Bidirectional Timing Parameters** *Note (1)*

| Symbol | -3DX Speed Grade | | -3 Speed Grade | | -4 Speed Grade | | Unit |
|---|---|---|---|---|---|---|---|
| | Min | Max | Min | Max | Min | Max | |
| $t_{INSUBIDIR}$ *(4)* | 8.1 | | 8.1 | | 10.4 | | ns |
| $t_{INHBIDIR}$ *(4)* | 0.0 | | 0.0 | | 0.0 | | ns |
| $t_{OUTCOBIDIR}$ *(4)* | 2.0 | 11.1 | 2.0 | 11.1 | 2.0 | 14.3 | ns |
| $t_{XZBIDIR}$ *(4)* | | 15.3 | | 15.3 | | 18.4 | ns |
| $t_{ZXBIDIR}$ *(4)* | | 15.3 | | 15.3 | | 18.4 | ns |
| $t_{INSUBIDIR}$ *(5)* | 9.1 | | – | | – | | ns |
| $t_{INHBIDIR}$ *(5)* | 0.0 | | – | | – | | ns |
| $t_{OUTCOBIDIR}$ *(5)* | 2.0 | 7.2 | – | – | – | – | ns |
| $t_{XZBIDIR}$ *(5)* | | 14.3 | | – | | – | ns |
| $t_{ZXBIDIR}$ *(5)* | | 14.3 | | – | | – | 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.
(4) This parameter is measured without the use of the ClockLock or ClockBoost circuits.
(5) This parameter is measured with the use of the ClockLock or ClockBoost circuits.

| 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 |

| *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 |
| $t_{EABBYPASS}$ | | 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 |

*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.*

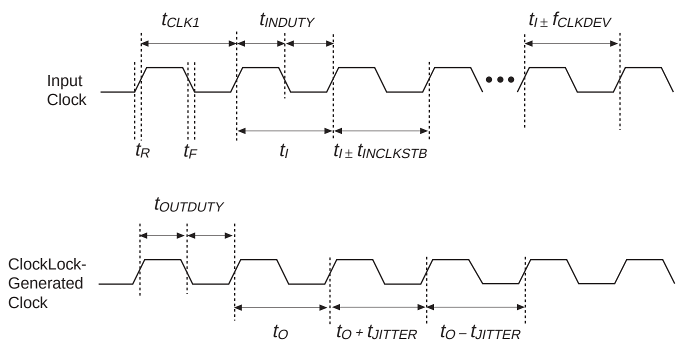



Table 113 summarizes the ClockLock and ClockBoost parameters.

**Table 113. ClockLock & ClockBoost Parameters (Part 1 of 2)**

| 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 |

$f_{MAX}$ = 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.