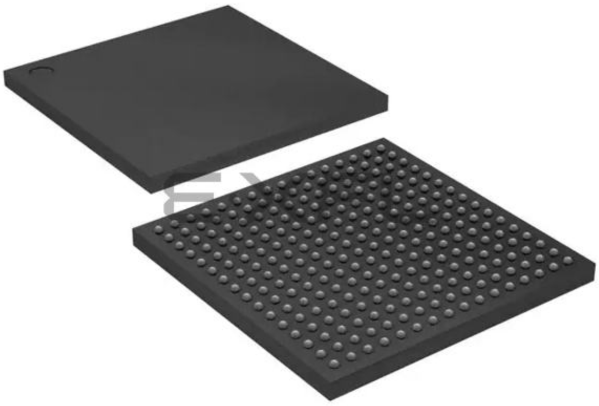


Welcome to E-XFL.COM

## 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

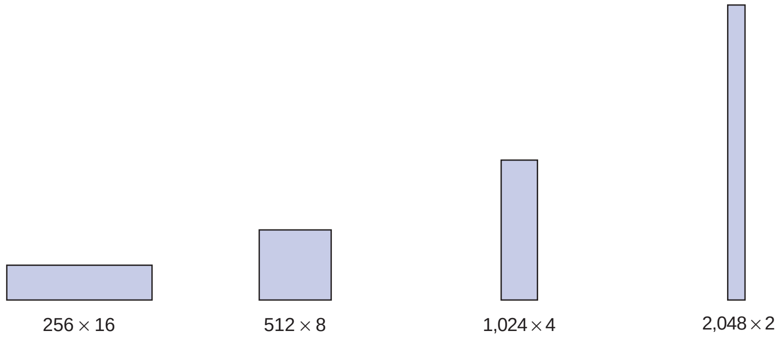
The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

| Details | |
|---|---|
| Product Status | Obsolete |
| Number of LABs/CLBs | 72 |
| Number of Logic Elements/Cells | 576 |
| Total RAM Bits | 12288 |
| Number of I/O | 136 |
| Number of Gates | 56000 |
| Voltage - Supply | 2.375V ~ 2.625V |
| Mounting Type | Surface Mount |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Package / Case | 256-BGA |
| Supplier Device Package | 256-FBGA (17x17) |
| Purchase URL | https://www.e-xfl.com/product-detail/intel/ep1k10fi256-2n |

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 signal, while ensuring that its data and address signals meet setup and hold time specifications relative to the write enable signal. In contrast, the EAB's synchronous RAM generates its own write enable signal and is self-timed with respect to the input or write clock. A circuit using the EAB's self-timed RAM must only meet the setup and hold time specifications of the global clock.

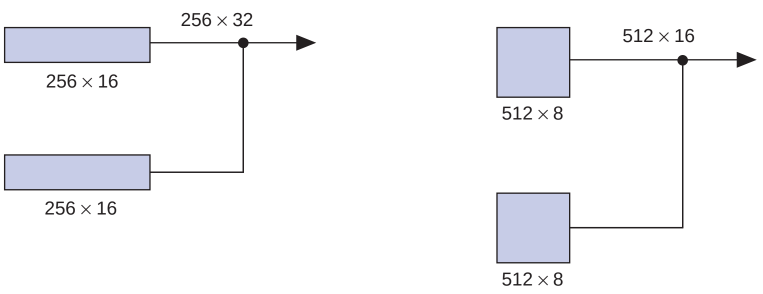
When used as RAM, each EAB can be configured in any of the following sizes: 256 × 16; 512 × 8; 1,024 × 4; or 2,048 × 2. Figure 5 shows the ACEX 1K EAB memory configurations.

***Figure 5. ACEX 1K EAB Memory Configurations***




Larger blocks of RAM are created by combining multiple EABs. For example, two 256 × 16 RAM blocks can be combined to form a 256 × 32 block, and two 512 × 8 RAM blocks can be combined to form a 512 × 16 block. Figure 6 shows examples of multiple EAB combination.

***Figure 6. Examples of Combining ACEX 1K EABs***



If necessary, all EABs in a device can be cascaded to form a single RAM block. EABs can be cascaded to form RAM blocks of up to 2,048 words without impacting timing. Altera software automatically combines EABs to meet a designer's RAM specifications.

EABs provide flexible options for driving and controlling clock signals. Different clocks and clock enables can be used for reading and writing to the EAB. Registers can be independently inserted on the data input, EAB output, write address, write enable signals, read address, and read enable signals. The global signals and the EAB local interconnect can drive write-enable, read-enable, and clock-enable signals. The global signals, dedicated clock pins, and EAB local interconnect can drive the EAB clock signals. Because the LEs drive the EAB local interconnect, the LEs can control write-enable, read-enable, clear, clock, and clock-enable signals.

An EAB is fed by a row interconnect and can drive out to row and column interconnects. Each EAB output can drive up to two row channels and up to two column channels; the unused row channel can be driven by other LEs. This feature increases the routing resources available for EAB outputs (see Figures 2 and 4). The column interconnect, which is adjacent to the EAB, has twice as many channels as other columns in the device.

## Logic Array Block

An 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 ACEX 1K architecture, facilitating efficient routing with optimum device utilization and high performance. Figure 7 shows the ACEX 1K LAB.

### *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 ACEX 1K architecture to efficiently implement high-speed counters, adders, and comparators of arbitrary width. Carry chain logic can be created automatically by the 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 EP1K50 device, the carry chain stops at the eighteenth LAB, and a new carry chain begins at the nineteenth LAB.

Figure 9 shows how an $n$-bit full adder can be implemented in $n + 1$ LEs with the carry chain. One portion of the LUT generates the sum of two bits using the input signals and the carry-in signal; the sum is routed to the output of the LE. The register can be bypassed for simple adders or used for an accumulator function. Another portion of the LUT and the carry chain logic generates the carry-out signal, which is routed directly to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it can be used as a general-purpose signal.

### Clearable Counter Mode

The clearable counter mode is similar to the up/down counter mode, but it 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. Two 3-input LUTs are used; 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-states 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

On all ACEX 1K devices, the input path from the I/O pad to the FastTrack Interconnect has a programmable delay element that can be used to guarantee a zero hold time. Depending on the placement of the IOE relative to what it is driving, the designer may choose to turn on the programmable delay to ensure a zero hold time or turn it off to minimize setup time. This feature is used to reduce setup time for complex pin-to-register paths (e.g., PCI designs).

Each IOE selects the clock, clear, clock enable, and output enable controls from a network of I/O control signals called the peripheral control bus. The peripheral control bus uses high-speed drivers to minimize signal skew across devices and provides up to 12 peripheral control signals that can be allocated as follows:

- Up to eight output enable signals
- Up to six clock enable signals
- Up to two clock signals
- Up to two clear signals

If more than six clock-enable or eight output-enable signals are required, each IOE on the device can be controlled by clock enable and output enable signals driven by specific LEs. In addition to the two clock signals available on the peripheral control bus, each IOE can use one of two dedicated clock pins. Each peripheral control signal can be driven by any of the dedicated input pins or the first LE of each LAB in a particular row. In addition, a LE in a different row can drive a column interconnect, which causes a row interconnect to drive the peripheral control signal. The chip-wide reset signal resets all IOE registers, overriding any other control signals.

When a dedicated clock pin drives IOE registers, it can be inverted for all IOEs in the device. All IOEs must use the same sense of the clock. For example, if any IOE uses the inverted clock, all IOEs must use the inverted clock, and no IOE can use the non-inverted clock. However, LEs can still use the true or complement of the clock on an LAB-by-LAB basis.

The incoming signal may be inverted at the dedicated clock pin and will drive all IOEs. For the true and complement of a clock to be used to drive IOEs, drive it into both global clock pins. One global clock pin will supply the true, and the other will supply the complement.

When the true and complement of a dedicated input drives IOE clocks, two signals on the peripheral control bus are consumed, one for each sense of the clock.

**Table 16. 32-Bit IDCODE for ACEX 1K Devices** *Note (1)*

| Device | IDCODE (32 Bits) | | | |
|---|---|---|---|---|
| | Version (4 Bits) | Part Number (16 Bits) | Manufacturer's Identity (11 Bits) | 1 (1 Bit) *(2)* |
| EP1K10 | 0001 | 0001 0000 0001 0000 | 00001101110 | 1 |
| EP1K30 | 0001 | 0001 0000 0011 0000 | 00001101110 | 1 |
| EP1K50 | 0001 | 0001 0000 0101 0000 | 00001101110 | 1 |
| EP1K100 | 0010 | 0000 0001 0000 0000 | 00001101110 | 1 |

***Notes to tables:***
(1) The most significant bit (MSB) is on the left.
(2) The least significant bit (LSB) for all JTAG IDCODEs is 1.

ACEX 1K devices include weak pull-up resistors on the JTAG pins.

For more information, see the following documents:

- *Application Note 39 (IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices)*
- *ByteBlasterMV Parallel Port Download Cable Data Sheet*
- *BitBlaster Serial Download Cable Data Sheet*
- *Jam Programming & Test Language Specification*

Figure 20 shows the timing requirements for the JTAG signals.

*Figure 20. ACEX 1K JTAG Waveforms*

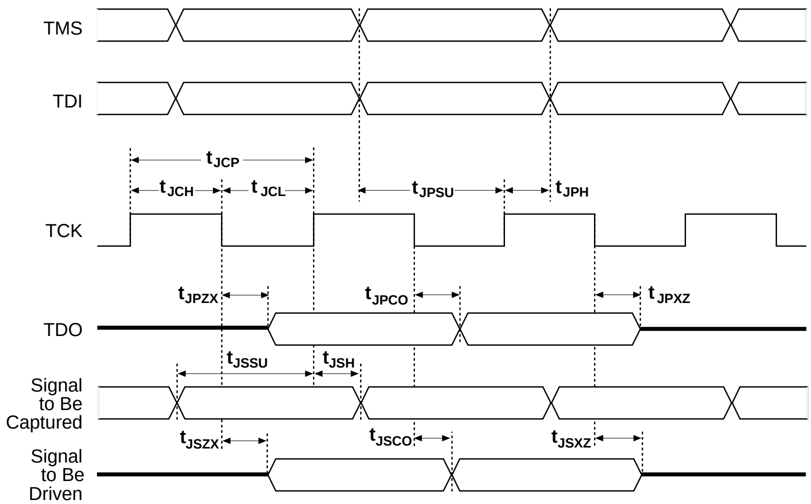



Table 17 shows the timing parameters and values for ACEX 1K devices.

*Table 17. ACEX 1K JTAG Timing Parameters & Values*

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{JCP}$ | TCK clock period | 100 | | ns |
| $t_{JCH}$ | TCK clock high time | 50 | | ns |
| $t_{JCL}$ | TCK clock low time | 50 | | ns |
| $t_{JPSU}$ | JTAG port setup time | 20 | | ns |
| $t_{JPH}$ | JTAG port hold time | 45 | | ns |
| $t_{JPCO}$ | JTAG port clock to output | | 25 | ns |
| $t_{JPZX}$ | JTAG port high impedance to valid output | | 25 | ns |
| $t_{JPXZ}$ | JTAG port valid output to high impedance | | 25 | ns |
| $t_{JSSU}$ | Capture register setup time | 20 | | ns |
| $t_{JSH}$ | Capture register hold time | 45 | | ns |
| $t_{JSCO}$ | Update register clock to output | | 35 | ns |
| $t_{JSZX}$ | Update register high impedance to valid output | | 35 | ns |
| $t_{JSXZ}$ | Update register valid output to high impedance | | 35 | ns |

| Table 19. ACEX 1K 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 | (2), (5) | –0.5 | 5.75 | V |
| $V_O$ | Output voltage | | 0 | $V_{CCIO}$ | V |
| $T_A$ | Ambient temperature | Commercial range | 0 | 70 | ° C |
| | | Industrial range | –40 | 85 | ° C |
| $T_J$ | Junction temperature | Commercial range | 0 | 85 | ° C |
| | | Industrial range | –40 | 100 | ° C |
| | | Extended range | –40 | 125 | ° C |
| $t_R$ | Input rise time | | | 40 | ns |
| $t_F$ | Input fall time | | | 40 | ns |

| Table 20. ACEX 1K Device DC Operating Conditions (Part 1 of 2) Notes (6), (7) | | | | | | |
|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Conditions** | **Min** | **Typ** | **Max** | **Unit** |
| $V_{IH}$ | High-level input voltage | | 1.7, $0.5 \times V_{CCIO}$ (8) | | 5.75 | V |
| $V_{IL}$ | Low-level input voltage | | –0.5 | | 0.8, $0.3 \times V_{CCIO}$ (8) | V |
| $V_{OH}$ | 3.3-V high-level TTL output voltage | $I_{OH}$ = –8 mA DC, $V_{CCIO}$ = 3.00 V (9) | 2.4 | | | V |
| | 3.3-V high-level CMOS output voltage | $I_{OH}$ = –0.1 mA DC, $V_{CCIO}$ = 3.00 V (9) | $V_{CCIO} - 0.2$ | | | V |
| | 3.3-V high-level PCI output voltage | $I_{OH}$ = –0.5 mA DC, $V_{CCIO}$ = 3.00 to 3.60 V (9) | 0.9 ×†$V_{CCIO}$ | | | V |
| | 2.5-V high-level output voltage | $I_{OH}$ = –0.1 mA DC, $V_{CCIO}$ = 2.375 V (9) | 2.1 | | | V |
| | | $I_{OH}$ = –1 mA DC, $V_{CCIO}$ = 2.375 V (9) | 2.0 | | | V |
| | | $I_{OH}$ = –2 mA DC, $V_{CCIO}$ = 2.375 V (9) | 1.7 | | | V |

| *Table 21. ACEX 1K Device Capacitance* *Note (14)* | | | | | |
|---|---|---|---|---|---|
| **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 | | 12 | pF |
| $C_{OUT}$ | Output capacitance | $V_{OUT}$ = 0 V, f = 1.0 MHz | | 10 | pF |

***Notes to tables:***

(1) See the *Operating Requirements for Altera Devices Data Sheet*.
(2) Minimum DC input voltage is –0.5 V. During transitions, the inputs may undershoot to –2.0 V for input currents less than 100 mA and periods shorter than 20 ns.
(3) Numbers in parentheses are for industrial- and extended-temperature-range devices.
(4) Maximum $V_{CC}$ rise time is 100 ms, and $V_{CC}$ must rise monotonically.
(5) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before $V_{CCINT}$ and $V_{CCIO}$ are powered.
(6) Typical values are for $T_A$ = 25° C, $V_{CCINT}$ = 2.5 V, and $V_{CCIO}$ = 2.5 V or 3.3 V.
(7) These values are specified under the ACEX 1K Recommended Operating Conditions shown in Table 19 on page 46.
(8) The ACEX 1K input buffers are compatible with 2.5-V, 3.3-V (LVTTL and LVCMOS), and 5.0-V TTL and CMOS signals. Additionally, the input buffers are 3.3-V PCI compliant when $V_{CCIO}$ and $V_{CCINT}$ meet the relationship shown in Figure 22.
(9) The $I_{OH}$ parameter refers to high-level TTL, PCI, or CMOS output current.
(10) The $I_{OL}$ parameter refers to low-level TTL, PCI, or CMOS output current. This parameter applies to open-drain pins as well as output pins.
(11) This value is specified for normal device operation. The value may vary during power-up.
(12) This parameter applies to -1 speed grade commercial temperature devices and -2 speed grade industrial and extended temperature devices.
(13) Pin pull-up resistance values will be lower if the pin is driven higher than $V_{CCIO}$ by an external source.
(14) Capacitance is sample-tested only.

Figure 22 shows the required relationship between $V_{CCIO}$ and $V_{CCINT}$ to satisfy 3.3-V PCI compliance.

***Figure 22. Relationship between $V_{CCIO}$ & $V_{CCINT}$ for 3.3-V PCI Compliance***

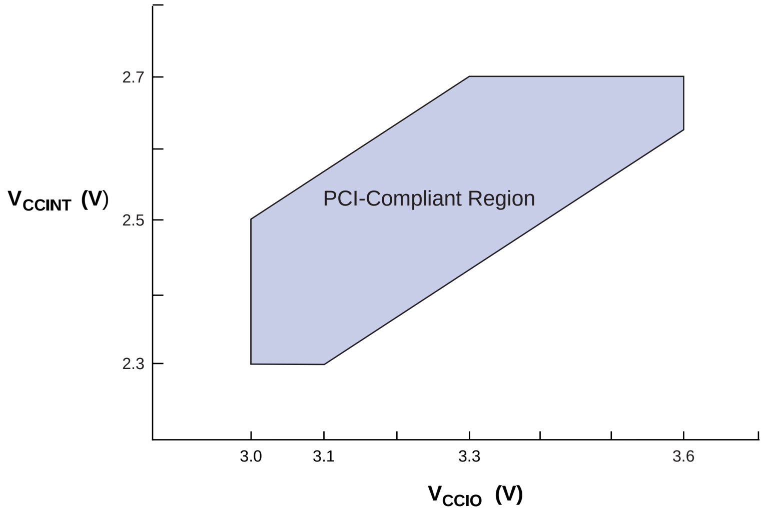
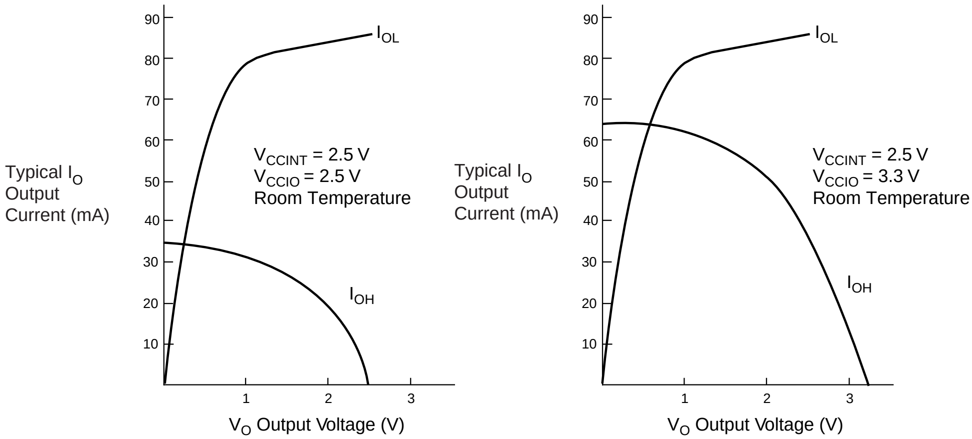



Figure 23 shows the typical output drive characteristics of ACEX 1K devices with 3.3-V and 2.5-V $V_{CCIO}$. The output driver is compliant to the 3.3-V ***PCI Local Bus Specification***, ***Revision 2.2*** (when `VCCIO` pins are connected to 3.3 V). ACEX 1K devices with a -1 speed grade also comply with the drive strength requirements of the ***PCI Local Bus Specification***, ***Revision 2.2*** (when `VCCINT` pins are powered with a minimum supply of 2.375 V, and `VCCIO` pins are connected to 3.3 V). Therefore, these devices can be used in open 5.0-V PCI systems.

***Figure 23. Output Drive Characteristics of ACEX 1K Devices***




## Timing Model

The continuous, high-performance FastTrack Interconnect routing resources ensure accurate simulation and timing analysis as well as predictable performance. This predictable performance contrasts with that of FPGAs, which use a segmented connection scheme and, therefore, have an 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.

Timing simulation and delay prediction are available with the 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.

| ***Table 24. EAB Timing Microparameters*** *Note (1)* | | |
|---|---|---|
| **Symbol** | **Parameter** | **Conditions** |
| $t_{EABDATA1}$ | Data or address delay to EAB for combinatorial input | |
| $t_{EABDATA2}$ | Data or address delay to EAB for registered input | |
| $t_{EABWE1}$ | Write enable delay to EAB for combinatorial input | |
| $t_{EABWE2}$ | Write enable delay to EAB for registered input | |
| $t_{EABRE1}$ | Read enable delay to EAB for combinatorial input | |
| $t_{EABRE2}$ | Read enable delay to EAB for registered input | |
| $t_{EABCLK}$ | EAB register clock delay | |
| $t_{EABCO}$ | EAB register clock-to-output delay | |
| $t_{EABBYPASS}$ | Bypass register delay | |
| $t_{EABSU}$ | EAB register setup time before clock | |
| $t_{EABH}$ | EAB register hold time after clock | |
| $t_{EABCLR}$ | EAB register asynchronous clear time to output delay | |
| $t_{AA}$ | Address access delay (including the read enable to output delay) | |
| $t_{WP}$ | Write pulse width | |
| $t_{RP}$ | Read pulse width | |
| $t_{WDSU}$ | Data setup time before falling edge of write pulse | *(5)* |
| $t_{WDH}$ | Data hold time after falling edge of write pulse | *(5)* |
| $t_{WASU}$ | Address setup time before rising edge of write pulse | *(5)* |
| $t_{WAH}$ | Address hold time after falling edge of write pulse | *(5)* |
| $t_{RASU}$ | Address setup time before rising edge of read pulse | |
| $t_{RAH}$ | Address hold time after falling edge of read pulse | |
| $t_{WO}$ | Write enable to data output valid delay | |
| $t_{DD}$ | Data-in to data-out valid delay | |
| $t_{EABOUT}$ | Data-out delay | |
| $t_{EABCH}$ | Clock high time | |
| $t_{EABCL}$ | Clock low time | |

Tables 30 through 36 show EP1K10 device internal and external timing parameters.

**Table 30. EP1K10 Device LE Timing Microparameters** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{LUT}$ | | 0.7 | | 0.8 | | 1.1 | ns |
| $t_{CLUT}$ | | 0.5 | | 0.6 | | 0.8 | ns |
| $t_{RLUT}$ | | 0.6 | | 0.7 | | 1.0 | ns |
| $t_{PACKED}$ | | 0.4 | | 0.4 | | 0.5 | ns |
| $t_{EN}$ | | 0.9 | | 1.0 | | 1.3 | ns |
| $t_{CICO}$ | | 0.1 | | 0.1 | | 0.2 | ns |
| $t_{CGEN}$ | | 0.4 | | 0.5 | | 0.7 | ns |
| $t_{CGENR}$ | | 0.1 | | 0.1 | | 0.2 | ns |
| $t_{CASC}$ | | 0.7 | | 0.9 | | 1.1 | ns |
| $t_C$ | | 1.1 | | 1.3 | | 1.7 | ns |
| $t_{CO}$ | | 0.5 | | 0.7 | | 0.9 | ns |
| $t_{COMB}$ | | 0.4 | | 0.5 | | 0.7 | ns |
| $t_{SU}$ | 0.7 | | 0.8 | | 1.0 | | ns |
| $t_H$ | 0.9 | | 1.0 | | 1.1 | | ns |
| $t_{PRE}$ | | 0.8 | | 1.0 | | 1.4 | ns |
| $t_{CLR}$ | | 0.9 | | 1.0 | | 1.4 | ns |
| $t_{CH}$ | 2.0 | | 2.5 | | 2.5 | | ns |
| $t_{CL}$ | 2.0 | | 2.5 | | 2.5 | | ns |

**Table 32. EP1K10 Device EAB Internal Microparameters** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{EABDATA1}$ | | 1.8 | | 1.9 | | 1.9 | ns |
| $t_{EABDATA2}$ | | 0.6 | | 0.7 | | 0.7 | ns |
| $t_{EABWE1}$ | | 1.2 | | 1.2 | | 1.2 | ns |
| $t_{EABWE2}$ | | 0.4 | | 0.4 | | 0.4 | ns |
| $t_{EABRE1}$ | | 0.9 | | 0.9 | | 0.9 | ns |
| $t_{EABRE2}$ | | 0.4 | | 0.4 | | 0.4 | ns |
| $t_{EABCLK}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABCO}$ | | 0.3 | | 0.3 | | 0.3 | ns |
| $t_{EABBYPASS}$ | | 0.5 | | 0.6 | | 0.6 | ns |
| $t_{EABSU}$ | 1.0 | | 1.0 | | 1.0 | | ns |
| $t_{EABH}$ | 0.5 | | 0.4 | | 0.4 | | ns |
| $t_{EABCLR}$ | 0.3 | | 0.3 | | 0.3 | | ns |
| $t_{AA}$ | | 3.4 | | 3.6 | | 3.6 | ns |
| $t_{WP}$ | 2.7 | | 2.8 | | 2.8 | | ns |
| $t_{RP}$ | 1.0 | | 1.0 | | 1.0 | | ns |
| $t_{WDSU}$ | 1.0 | | 1.0 | | 1.0 | | ns |
| $t_{WDH}$ | 0.1 | | 0.1 | | 0.1 | | ns |
| $t_{WASU}$ | 1.8 | | 1.9 | | 1.9 | | ns |
| $t_{WAH}$ | 1.9 | | 2.0 | | 2.0 | | ns |
| $t_{RASU}$ | 3.1 | | 3.5 | | 3.5 | | ns |
| $t_{RAH}$ | 0.2 | | 0.2 | | 0.2 | | ns |
| $t_{WO}$ | | 2.7 | | 2.8 | | 2.8 | ns |
| $t_{DD}$ | | 2.7 | | 2.8 | | 2.8 | ns |
| $t_{EABOUT}$ | | 0.5 | | 0.6 | | 0.6 | ns |
| $t_{EABCH}$ | 1.5 | | 2.0 | | 2.0 | | ns |
| $t_{EABCL}$ | 2.7 | | 2.8 | | 2.8 | | ns |

**Table 33. EP1K10 Device EAB Internal Timing Macroparameters** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{EABAA}$ | | 6.7 | | 7.3 | | 7.3 | ns |
| $t_{EABRCCOMB}$ | 6.7 | | 7.3 | | 7.3 | | ns |
| $t_{EABRCREG}$ | 4.7 | | 4.9 | | 4.9 | | ns |
| $t_{EABWP}$ | 2.7 | | 2.8 | | 2.8 | | ns |
| $t_{EABWCCOMB}$ | 6.4 | | 6.7 | | 6.7 | | ns |
| $t_{EABWCREG}$ | 7.4 | | 7.6 | | 7.6 | | ns |
| $t_{EABDD}$ | | 6.0 | | 6.5 | | 6.5 | ns |
| $t_{EABDATACO}$ | | 0.8 | | 0.9 | | 0.9 | ns |
| $t_{EABDATASU}$ | 1.6 | | 1.7 | | 1.7 | | ns |
| $t_{EABDATAH}$ | 0.0 | | 0.0 | | 0.0 | | ns |
| $t_{EABWESU}$ | 1.4 | | 1.4 | | 1.4 | | ns |
| $t_{EABWEH}$ | 0.1 | | 0.0 | | 0.0 | | ns |
| $t_{EABWDSU}$ | 1.6 | | 1.7 | | 1.7 | | ns |
| $t_{EABWDH}$ | 0.0 | | 0.0 | | 0.0 | | ns |
| $t_{EABWASU}$ | 3.1 | | 3.4 | | 3.4 | | ns |
| $t_{EABWAH}$ | 0.6 | | 0.5 | | 0.5 | | ns |
| $t_{EABWO}$ | | 5.4 | | 5.8 | | 5.8 | ns |

**Table 46. EP1K50 Device EAB Internal Microparameters** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{EABDATA1}$ | | 1.7 | | 2.4 | | 3.2 | ns |
| $t_{EABDATA2}$ | | 0.4 | | 0.6 | | 0.8 | ns |
| $t_{EABWE1}$ | | 1.0 | | 1.4 | | 1.9 | ns |
| $t_{EABWE2}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABRE1}$ | | 0.0 | | 0.0 | | 0.0 | |
| $t_{EABRE2}$ | | 0.4 | | 0.6 | | 0.8 | |
| $t_{EABCLK}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABCO}$ | | 0.8 | | 1.1 | | 1.5 | ns |
| $t_{EABBYPASS}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABSU}$ | 0.7 | | 1.0 | | 1.3 | | ns |
| $t_{EABH}$ | 0.4 | | 0.6 | | 0.8 | | ns |
| $t_{EABCLR}$ | 0.8 | | 1.1 | | 1.5 | | |
| $t_{AA}$ | | 2.0 | | 2.8 | | 3.8 | ns |
| $t_{WP}$ | 2.0 | | 2.8 | | 3.8 | | ns |
| $t_{RP}$ | 1.0 | | 1.4 | | 1.9 | | |
| $t_{WDSU}$ | 0.5 | | 0.7 | | 0.9 | | ns |
| $t_{WDH}$ | 0.1 | | 0.1 | | 0.2 | | ns |
| $t_{WASU}$ | 1.0 | | 1.4 | | 1.9 | | ns |
| $t_{WAH}$ | 1.5 | | 2.1 | | 2.9 | | ns |
| $t_{RASU}$ | 1.5 | | 2.1 | | 2.8 | | |
| $t_{RAH}$ | 0.1 | | 0.1 | | 0.2 | | |
| $t_{WO}$ | | 2.1 | | 2.9 | | 4.0 | ns |
| $t_{DD}$ | | 2.1 | | 2.9 | | 4.0 | ns |
| $t_{EABOUT}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABCH}$ | 1.5 | | 2.0 | | 2.5 | | ns |
| $t_{EABCL}$ | 1.5 | | 2.0 | | 2.5 | | ns |

*Table 47. EP1K50 Device EAB Internal Timing Macroparameters* *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{EABAA}$ | | 3.7 | | 5.2 | | 7.0 | ns |
| $t_{EABRCCOMB}$ | 3.7 | | 5.2 | | 7.0 | | ns |
| $t_{EABRCREG}$ | 3.5 | | 4.9 | | 6.6 | | ns |
| $t_{EABWP}$ | 2.0 | | 2.8 | | 3.8 | | ns |
| $t_{EABWCCOMB}$ | 4.5 | | 6.3 | | 8.6 | | ns |
| $t_{EABWCREG}$ | 5.6 | | 7.8 | | 10.6 | | ns |
| $t_{EABDD}$ | | 3.8 | | 5.3 | | 7.2 | ns |
| $t_{EABDATACO}$ | | 0.8 | | 1.1 | | 1.5 | ns |
| $t_{EABDATASU}$ | 1.1 | | 1.6 | | 2.1 | | ns |
| $t_{EABDATAH}$ | 0.0 | | 0.0 | | 0.0 | | ns |
| $t_{EABWESU}$ | 0.7 | | 1.0 | | 1.3 | | ns |
| $t_{EABWEH}$ | 0.4 | | 0.6 | | 0.8 | | ns |
| $t_{EABWDSU}$ | 1.2 | | 1.7 | | 2.2 | | ns |
| $t_{EABWDH}$ | 0.0 | | 0.0 | | 0.0 | | ns |
| $t_{EABWASU}$ | 1.6 | | 2.3 | | 3.0 | | ns |
| $t_{EABWAH}$ | 0.9 | | 1.2 | | 1.8 | | ns |
| $t_{EABWO}$ | | 3.1 | | 4.3 | | 5.9 | ns |

| *Table 53. EP1K100 Device EAB Internal Microparameters* *Note (1)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Symbol** | **Speed Grade** | | | | | | **Unit** |
| | **-1** | | **-2** | | **-3** | | |
| | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | |
| $t_{EABDATA1}$ | | 1.5 | | 2.0 | | 2.6 | ns |
| $t_{EABDATA1}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABWE1}$ | | 1.5 | | 2.0 | | 2.6 | ns |
| $t_{EABWE2}$ | | 0.3 | | 0.4 | | 0.5 | ns |
| $t_{EABRE1}$ | | 0.3 | | 0.4 | | 0.5 | ns |
| $t_{EABRE2}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABCLK}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABCO}$ | | 0.3 | | 0.4 | | 0.5 | ns |
| $t_{EABBYPASS}$ | | 0.1 | | 0.1 | | 0.2 | ns |
| $t_{EABSU}$ | 0.8 | | 1.0 | | 1.4 | | ns |
| $t_{EABH}$ | 0.1 | | 0.1 | | 0.2 | | ns |
| $t_{EABCLR}$ | 0.3 | | 0.4 | | 0.5 | | ns |
| $t_{AA}$ | | 4.0 | | 5.1 | | 6.6 | ns |
| $t_{WP}$ | 2.7 | | 3.5 | | 4.7 | | ns |
| $t_{RP}$ | 1.0 | | 1.3 | | 1.7 | | ns |
| $t_{WDSU}$ | 1.0 | | 1.3 | | 1.7 | | ns |
| $t_{WDH}$ | 0.2 | | 0.2 | | 0.3 | | ns |
| $t_{WASU}$ | 1.6 | | 2.1 | | 2.8 | | ns |
| $t_{WAH}$ | 1.6 | | 2.1 | | 2.8 | | ns |
| $t_{RASU}$ | 3.0 | | 3.9 | | 5.2 | | ns |
| $t_{RAH}$ | 0.1 | | 0.1 | | 0.2 | | ns |
| $t_{WO}$ | | 1.5 | | 2.0 | | 2.6 | ns |
| $t_{DD}$ | | 1.5 | | 2.0 | | 2.6 | ns |
| $t_{EABOUT}$ | | 0.2 | | 0.3 | | 0.3 | ns |
| $t_{EABCH}$ | 1.5 | | 2.0 | | 2.5 | | ns |
| $t_{EABCL}$ | 2.7 | | 3.5 | | 4.7 | | ns |

During initialization, which occurs immediately after configuration, the device resets registers, enables I/O pins, and begins to operate as a logic device. Before and during configuration, all I/O pins (except dedicated inputs, clock, or configuration pins) are pulled high by a weak pull-up resistor. Together, the configuration and initialization processes are called *command mode*; normal device operation is called *user mode*.

SRAM configuration elements allow ACEX 1K 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, re-initializing the device, and resuming user-mode operation. The entire reconfiguration process requires less than 40 ms and can be used to reconfigure an entire system dynamically. In-field upgrades can be performed by distributing new configuration files.

### Configuration Schemes

The configuration data for an ACEX 1K device can be loaded with one of five configuration schemes (see Table 59), chosen on the basis of the target application. An EPC16, EPC2, EPC1, or EPC1441 configuration device, intelligent controller, or the JTAG port can be used to control the configuration of a ACEX 1K device, allowing automatic configuration on system power-up.

Multiple ACEX 1K devices can be configured in any of the five configuration schemes by connecting the configuration enable (nCE) and configuration enable output (nCEO) pins on each device. Additional APEX 20K, APEX 20KE, FLEX 10K, FLEX 10KA, FLEX 10KE, ACEX 1K, and FLEX 6000 devices can be configured in the same serial chain.

***Table 59. Data Sources for ACEX 1K Configuration***

| Configuration Scheme | Data Source |
|---|---|
| Configuration device | EPC16, EPC2, EPC1, or EPC1441 configuration device |
| Passive serial (PS) | BitBlaster or ByteBlasterMV download cables, or serial data source |
| Passive parallel asynchronous (PPA) | Parallel data source |
| Passive parallel synchronous (PPS) | Parallel data source |
| JTAG | BitBlaster or ByteBlasterMV download cables, or microprocessor with a Jam STAPL File or JBC File |

## Device Pin-Outs

See the Altera web site (**http://www.altera.com**) or the ***Altera Documentation Library*** for pin-out information.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com





Printed on Recycled Paper.