
**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 | 216 |
| Number of Logic Elements/Cells | 1728 |
| Total RAM Bits | 24576 |
| Number of I/O | 147 |
| Number of Gates | 119000 |
| Voltage - Supply | 2.375V ~ 2.625V |
| Mounting Type | Surface Mount |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Package / Case | 208-BFQFP |
| Supplier Device Package | 208-PQFP (28x28) |
| Purchase URL | https://www.e-xfl.com/product-detail/intel/ep1k30qc208-3 |

Email: info@E-XFL.COM Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

- Software design support and automatic place-and-route provided by Altera development systems for Windows-based PCs and Sun SPARCstation, and HP 9000 Series 700/800 workstations
- Flexible package options are available in 100 to 484 pins, including the innovative FineLine BGA™ packages (see Tables 2 and 3)
- Additional design entry and simulation support provided by EDIF 2 0 0 and 3 0 0 netlist files, library of parameterized modules (LPM), DesignWare components, Verilog HDL, VHDL, and other interfaces to popular EDA tools from manufacturers such as Cadence, Exemplar Logic, Mentor Graphics, OrCAD, Synopsys, Synplicity, VeriBest, and Viewlogic

***Table 2. ACEX 1K Package Options & I/O Pin Count*** *Notes (1), (2)*

| Device | 100-Pin TQFP | 144-Pin TQFP | 208-Pin PQFP | 256-Pin FineLine BGA | 484-Pin FineLine BGA |
|---|---|---|---|---|---|
| EP1K10 | 66 | 92 | 120 | 136 | 136 *(3)* |
| EP1K30 | | 102 | 147 | 171 | 171 *(3)* |
| EP1K50 | | 102 | 147 | 186 | 249 |
| EP1K100 | | | 147 | 186 | 333 |

***Notes:***

(1) ACEX 1K device package types include thin quad flat pack (TQFP), plastic quad flat pack (PQFP), and FineLine BGA packages.

(2) Devices in the same package are pin-compatible, although some devices have more I/O pins than others. When planning device migration, use the I/O pins that are common to all devices.

(3) 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 256-pin and 484-pin FineLine BGA packages.

***Table 3. ACEX 1K Package Sizes***

| Device | 100-Pin TQFP | 144-Pin TQFP | 208-Pin PQFP | 256-Pin FineLine BGA | 484-Pin FineLine BGA |
|---|---|---|---|---|---|
| Pitch (mm) | 0.50 | 0.50 | 0.50 | 1.0 | 1.0 |
| Area ($mm^2$) | 256 | 484 | 936 | 289 | 529 |
| Length × width (mm × mm) | 16 × 16 | 22 × 22 | 30.6 × 30.6 | 17 × 17 | 23 × 23 |

Table 5 shows ACEX 1K device performance for more complex designs. These designs are available as Altera MegaCore™ functions.

*Table 5. ACEX 1K Device Performance for Complex Designs*

| Application | LEs Used | Performance | | | |
|---|---|---|---|---|---|
| | | Speed Grade | | | Units |
| | | -1 | -2 | -3 | |
| 16-bit, 8-tap parallel finite impulse response (FIR) filter | 597 | 192 | 156 | 116 | MSPS |
| 8-bit, 512-point Fast Fourier transform (FFT) function | 1,854 | 23.4 | 28.7 | 38.9 | µs |
| | | 113 | 92 | 68 | MHz |
| `a16450` universal asynchronous receiver/transmitter (UART) | 342 | 36 | 28 | 20.5 | MHz |

Each ACEX 1K device contains an embedded array and a logic array. The embedded array is used to implement a variety of memory functions or complex logic functions, such as digital signal processing (DSP), wide data-path manipulation, microcontroller applications, and data-transformation functions. The logic array performs the same function as the sea-of-gates in the gate array and is used to implement general logic such as counters, adders, state machines, and multiplexers. The combination of embedded and logic arrays provides the high performance and high density of embedded gate arrays, enabling designers to implement an entire system on a single device.

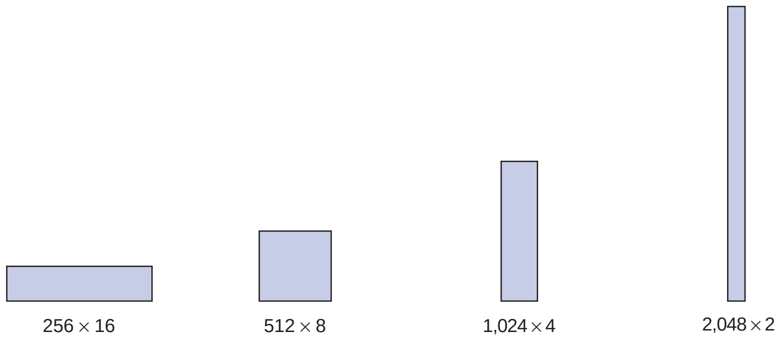
ACEX 1K devices are configured at system power-up with data stored in an Altera serial configuration device or provided by a system controller. Altera offers EPC16, EPC2, EPC1, and EPC1441 configuration devices, which configure ACEX 1K devices via a serial data stream. Configuration data can also be downloaded from system RAM or via the Altera MasterBlaster™, ByteBlasterMV™, or BitBlaster™ download cables. After an ACEX 1K device has been configured, it can be reconfigured in-circuit by resetting the device and loading new data. Because reconfiguration requires less than 40 ms, real-time changes can be made during system operation.

ACEX 1K devices contain an interface that permits microprocessors to configure ACEX 1K devices serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat an ACEX 1K device as memory and configure it by writing to a virtual memory location, simplifying device reconfiguration.

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.

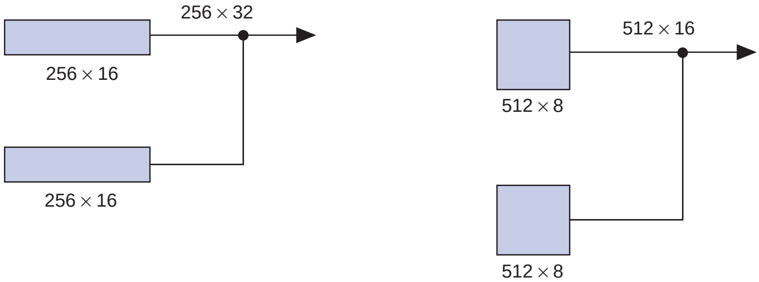
When used as RAM, each EAB can be configured in any of the following sizes: 256 × 16; 512 × 8; 1,024 × 4; or 2,048 × 2. Figure 5 shows the ACEX 1K EAB memory configurations.

***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.

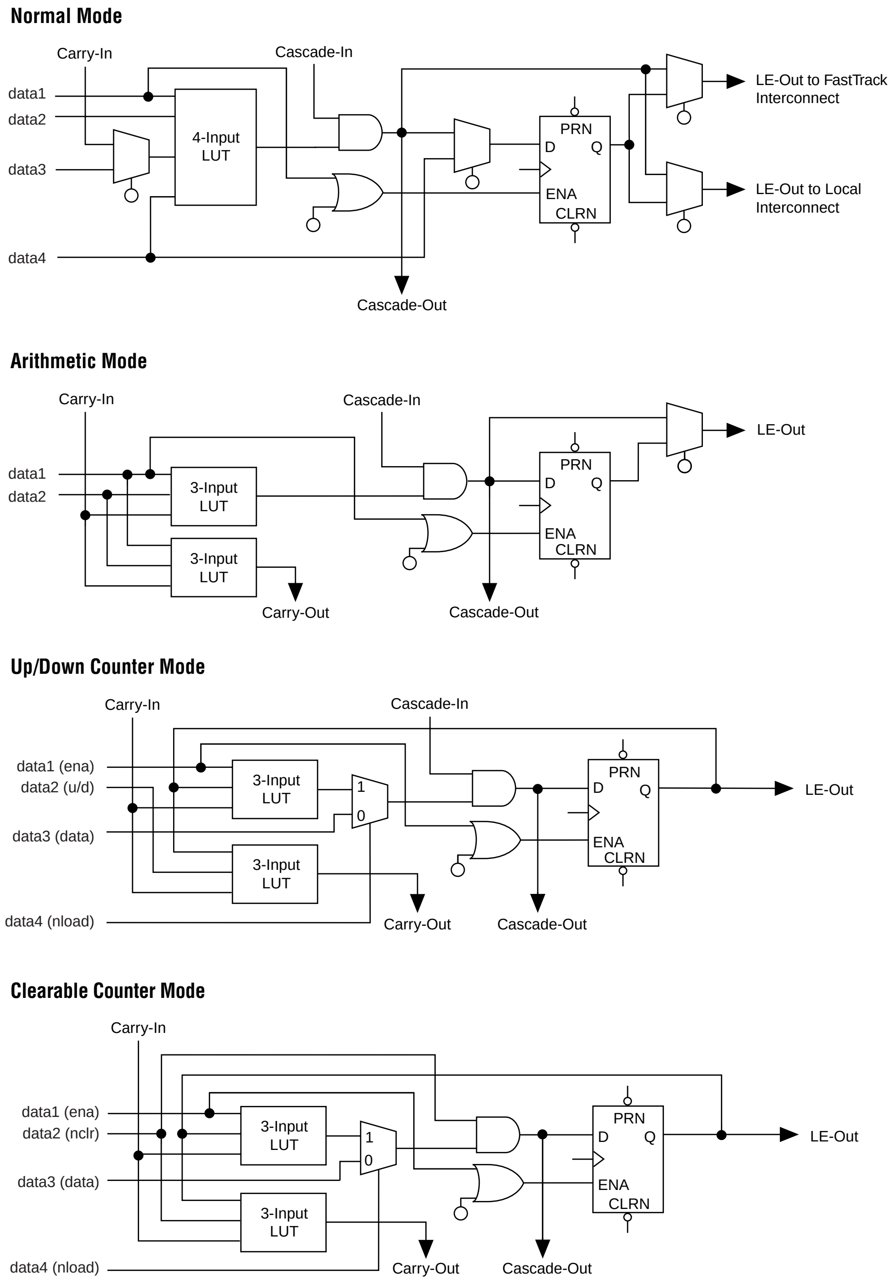
### *Carry Chain*

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.

*Figure 11. ACEX 1K LE Operating Modes*



For more information, search for "SameFrame" in MAX+PLUS II Help.

*Table 10. ACEX 1K SameFrame Pin-Out Support*

| Device | 256-Pin FineLine BGA | 484-Pin FineLine BGA |
|---|---|---|
| EP1K10 | ✓ | *(1)* |
| EP1K30 | ✓ | *(1)* |
| EP1K50 | ✓ | ✓ |
| EP1K100 | ✓ | ✓ |

***Note:***
(1) This option is supported with a 256-pin FineLine BGA package and SameFrame migration.

## ClockLock & ClockBoost Features

To support high-speed designs, -1 and -2 speed grade ACEX 1K devices offer 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 ACEX 1K 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 lock onto the rising edge of the incoming clock. The circuit output can drive the clock inputs of registers only; the generated clock cannot be gated or inverted.

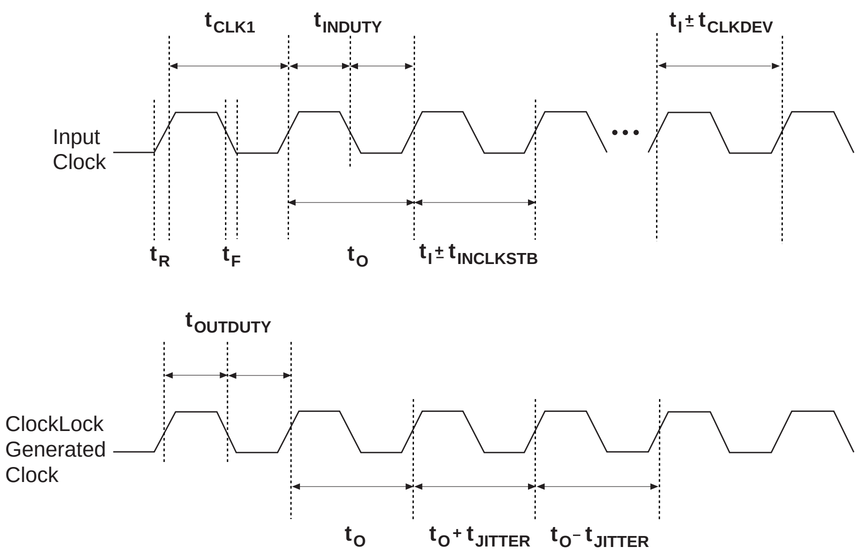
The dedicated clock pin (`GCLK1`) supplies the clock to the ClockLock and ClockBoost circuitry. When the dedicated clock pin is driving the ClockLock or ClockBoost circuitry, it cannot drive elsewhere in the device.

For designs that require both a multiplied and non-multiplied clock, the clock trace on the board can be connected to the `GCLK1` pin. In the Altera software, the `GCLK1` pin can feed both the ClockLock and ClockBoost circuitry in the ACEX 1K device. However, when both circuits are used, the other clock pin cannot be used.

## 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 19 shows the incoming and generated clock specifications.

***Figure 19. Specifications for the Incoming & Generated Clocks*** *Note (1)*




***Note:***

(1) The $\mathbf{t_I}$ parameter refers to the nominal input clock period; the $\mathbf{t_O}$ parameter refers to the nominal output clock period.

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

All ACEX 1K devices provide JTAG BST circuitry that complies with the IEEE Std. 1149.1-1990 specification. ACEX 1K devices can also be configured using the JTAG pins through the ByteBlasterMV or BitBlaster download cable, or via hardware that uses the Jam™ Standard Test and Programming Language (STAPL), JEDEC standard JESD-71. JTAG boundary-scan testing can be performed before or after configuration, but not during configuration. ACEX 1K devices support the JTAG instructions shown in Table 14.

*Table 14. ACEX 1K JTAG Instructions*

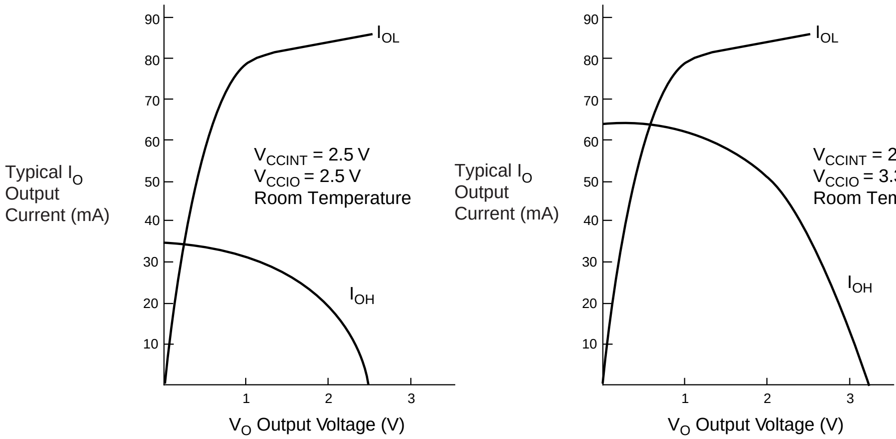
| JTAG Instruction | Description |
|---|---|
| SAMPLE/PRELOAD | Allows a snapshot of signals at the device pins to be captured and examined during normal device operation and permits an initial data pattern to be output at the device pins. |
| EXTEST | Allows the external circuitry and board-level interconnections to be tested by forcing a test pattern at the output pins and capturing test results at the input pins. |
| BYPASS | Places the 1-bit bypass register between the `TDI` and `TDO` pins, allowing the BST data to pass synchronously through a selected device to adjacent devices during normal operation. |
| USERCODE | Selects the user electronic signature (USERCODE) register and places it between the `TDI` and `TDO` pins, allowing the USERCODE to be serially shifted out of `TDO`. |
| IDCODE | Selects the IDCODE register and places it between `TDI` and `TDO`, allowing the IDCODE to be serially shifted out of `TDO`. |
| ICR Instructions | These instructions are used when configuring an ACEX 1K device via JTAG ports using a MasterBlaster, ByteBlasterMV, or BitBlaster download cable, or a Jam File (**.jam**) or Jam Byte-Code File (**.jbc**) via an embedded processor. |

The instruction register length of ACEX 1K devices is 10 bits. The USERCODE register length in ACEX 1K devices is 32 bits; 7 bits are determined by the user, and 25 bits are pre-determined. Tables 15 and 16 show the boundary-scan register length and device IDCODE information for ACEX 1K devices.

*Table 15. ACEX 1K Boundary-Scan Register Length*

| Device | Boundary-Scan Register Length |
|---|---|
| EP1K10 | 438 |
| EP1K30 | 690 |
| EP1K50 | 798 |
| EP1K100 | 1,050 |

*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:

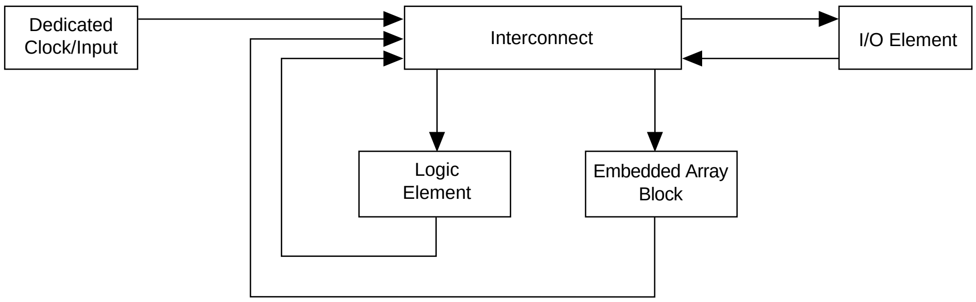
- LE register clock-to-output delay ($t_{CO}$)
- Interconnect delay ($t_{SAMEROW}$)
- LE look-up table delay ($t_{LUT}$)
- LE register setup time ($t_{SU}$)

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.

Figure 24 shows the overall timing model, which maps the possible paths to and from the various elements of the ACEX 1K device.

***Figure 24. ACEX 1K Device Timing Model***




Figures 25 through 28 show the delays that correspond to various paths and functions within the LE, IOE, EAB, and bidirectional timing models.

***Figure 25. ACEX 1K Device LE Timing Model***

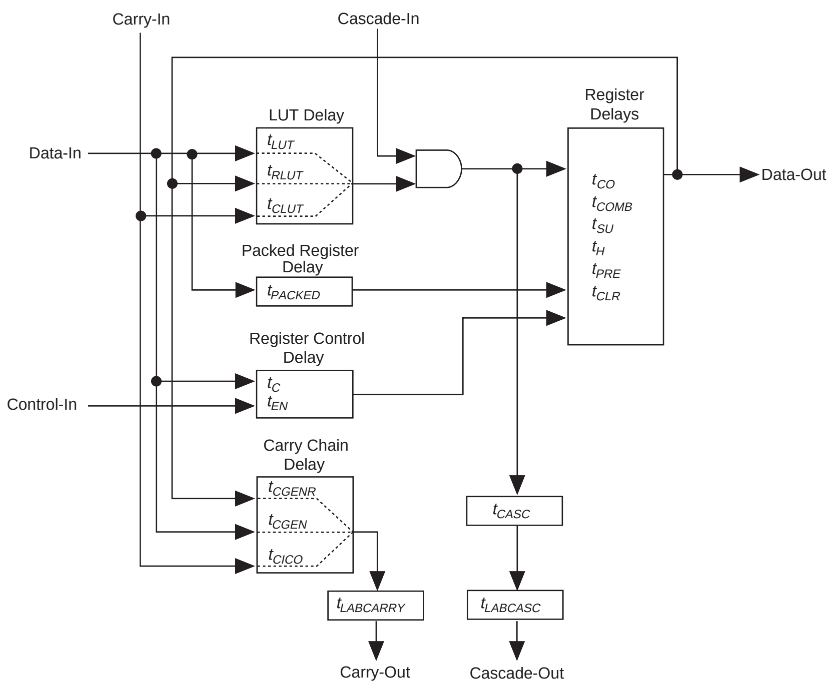
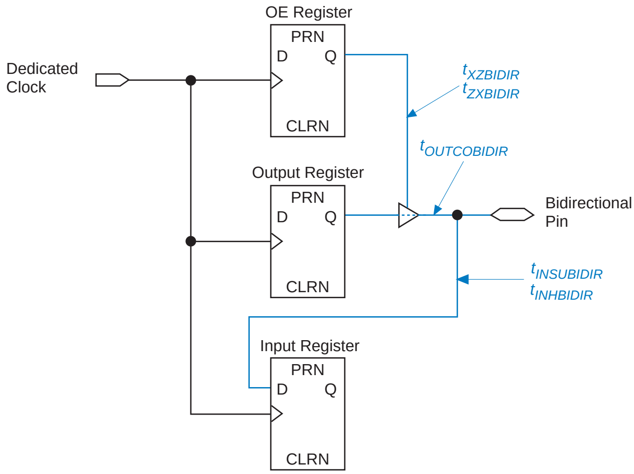
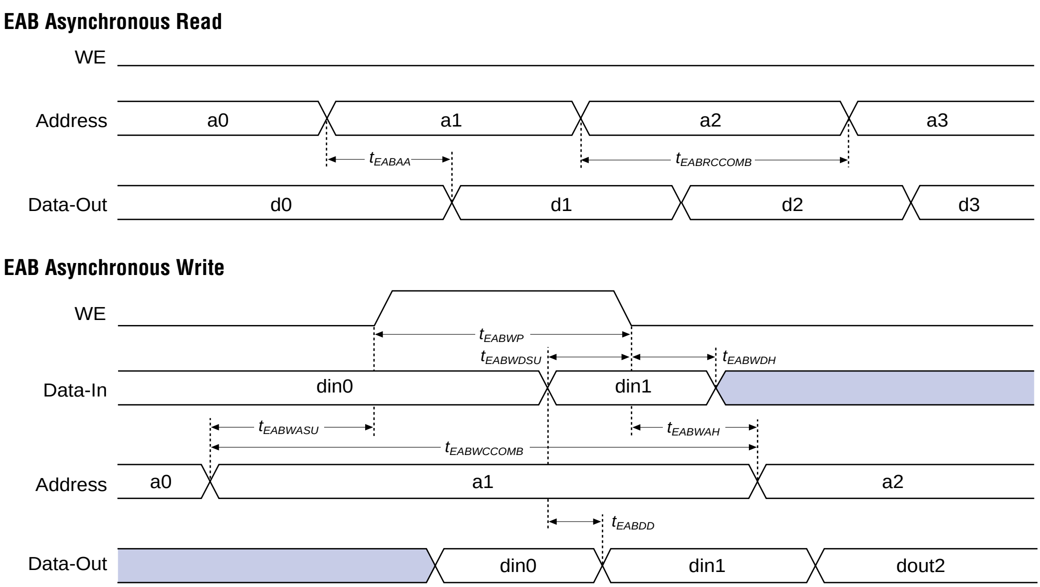


**Figure 28. Synchronous Bidirectional Pin External Timing Model**




Tables 29 and 30 show the asynchronous and synchronous timing waveforms, respectively, for the EAB macroparameters in Table 24.

**Figure 29. EAB Asynchronous Timing Waveforms**



| ***Table 25. EAB Timing Macroparameters*** *Notes (1), (6)* | | |
|---|---|---|
| **Symbol** | **Parameter** | **Conditions** |
| $t_{EABAA}$ | EAB address access delay | |
| $t_{EABRCCOMB}$ | EAB asynchronous read cycle time | |
| $t_{EABRCREG}$ | EAB synchronous read cycle time | |
| $t_{EABWP}$ | EAB write pulse width | |
| $t_{EABWCCOMB}$ | EAB asynchronous write cycle time | |
| $t_{EABWCREG}$ | EAB synchronous write cycle time | |
| $t_{EABDD}$ | EAB data-in to data-out valid delay | |
| $t_{EABDATACO}$ | EAB clock-to-output delay when using output registers | |
| $t_{EABDATASU}$ | EAB data/address setup time before clock when using input register | |
| $t_{EABDATAH}$ | EAB data/address hold time after clock when using input register | |
| $t_{EABWESU}$ | EAB WE setup time before clock when using input register | |
| $t_{EABWEH}$ | EAB WE hold time after clock when using input register | |
| $t_{EABWDSU}$ | EAB data setup time before falling edge of write pulse when not using input registers | |
| $t_{EABWDH}$ | EAB data hold time after falling edge of write pulse when not using input registers | |
| $t_{EABWASU}$ | EAB address setup time before rising edge of write pulse when not using input registers | |
| $t_{EABWAH}$ | EAB address hold time after falling edge of write pulse when not using input registers | |
| $t_{EABWO}$ | EAB write enable to data output valid delay | |

**Table 37. EP1K30 Device LE Timing Microparameters (Part 2 of 2)** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{COMB}$ | | 0.4 | | 0.4 | | 0.6 | ns |
| $t_{SU}$ | 0.4 | | 0.6 | | 0.6 | | ns |
| $t_{H}$ | 0.7 | | 1.0 | | 1.3 | | ns |
| $t_{PRE}$ | | 0.8 | | 0.9 | | 1.2 | ns |
| $t_{CLR}$ | | 0.8 | | 0.9 | | 1.2 | ns |
| $t_{CH}$ | 2.0 | | 2.5 | | 2.5 | | ns |
| $t_{CL}$ | 2.0 | | 2.5 | | 2.5 | | ns |

**Table 38. EP1K30 Device IOE Timing Microparameters** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{IOD}$ | | 2.4 | | 2.8 | | 3.8 | ns |
| $t_{IOC}$ | | 0.3 | | 0.4 | | 0.5 | ns |
| $t_{IOCO}$ | | 1.0 | | 1.1 | | 1.6 | ns |
| $t_{IOCOMB}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{IOSU}$ | 1.2 | | 1.4 | | 1.9 | | ns |
| $t_{IOH}$ | 0.3 | | 0.4 | | 0.5 | | ns |
| $t_{IOCLR}$ | | 1.0 | | 1.1 | | 1.6 | ns |
| $t_{OD1}$ | | 1.9 | | 2.3 | | 3.0 | ns |
| $t_{OD2}$ | | 1.4 | | 1.8 | | 2.5 | ns |
| $t_{OD3}$ | | 4.4 | | 5.2 | | 7.0 | ns |
| $t_{XZ}$ | | 2.7 | | 3.1 | | 4.3 | ns |
| $t_{ZX1}$ | | 2.7 | | 3.1 | | 4.3 | ns |
| $t_{ZX2}$ | | 2.2 | | 2.6 | | 3.8 | ns |
| $t_{ZX3}$ | | 5.2 | | 6.0 | | 8.3 | ns |
| $t_{INREG}$ | | 3.4 | | 4.1 | | 5.5 | ns |
| $t_{IOFD}$ | | 0.8 | | 1.3 | | 2.4 | ns |
| $t_{INCOMB}$ | | 0.8 | | 1.3 | | 2.4 | ns |

**Table 39. EP1K30 Device EAB Internal Microparameters** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{EABDATA1}$ | | 1.7 | | 2.0 | | 2.3 | ns |
| $t_{EABDATA1}$ | | 0.6 | | 0.7 | | 0.8 | ns |
| $t_{EABWE1}$ | | 1.1 | | 1.3 | | 1.4 | ns |
| $t_{EABWE2}$ | | 0.4 | | 0.4 | | 0.5 | ns |
| $t_{EABRE1}$ | | 0.8 | | 0.9 | | 1.0 | ns |
| $t_{EABRE2}$ | | 0.4 | | 0.4 | | 0.5 | ns |
| $t_{EABCLK}$ | | 0.0 | | 0.0 | | 0.0 | ns |
| $t_{EABCO}$ | | 0.3 | | 0.3 | | 0.4 | ns |
| $t_{EABBYPASS}$ | | 0.5 | | 0.6 | | 0.7 | ns |
| $t_{EABSU}$ | 0.9 | | 1.0 | | 1.2 | | ns |
| $t_{EABH}$ | 0.4 | | 0.4 | | 0.5 | | ns |
| $t_{EABCLR}$ | 0.3 | | 0.3 | | 0.3 | | ns |
| $t_{AA}$ | | 3.2 | | 3.8 | | 4.4 | ns |
| $t_{WP}$ | 2.5 | | 2.9 | | 3.3 | | ns |
| $t_{RP}$ | 0.9 | | 1.1 | | 1.2 | | ns |
| $t_{WDSU}$ | 0.9 | | 1.0 | | 1.1 | | ns |
| $t_{WDH}$ | 0.1 | | 0.1 | | 0.1 | | ns |
| $t_{WASU}$ | 1.7 | | 2.0 | | 2.3 | | ns |
| $t_{WAH}$ | 1.8 | | 2.1 | | 2.4 | | ns |
| $t_{RASU}$ | 3.1 | | 3.7 | | 4.2 | | ns |
| $t_{RAH}$ | 0.2 | | 0.2 | | 0.2 | | ns |
| $t_{WO}$ | | 2.5 | | 2.9 | | 3.3 | ns |
| $t_{DD}$ | | 2.5 | | 2.9 | | 3.3 | ns |
| $t_{EABOUT}$ | | 0.5 | | 0.6 | | 0.7 | ns |
| $t_{EABCH}$ | 1.5 | | 2.0 | | 2.3 | | ns |
| $t_{EABCL}$ | 2.5 | | 2.9 | | 3.3 | | ns |

**Table 40. EP1K30 Device EAB Internal Timing Macroparameters** *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | 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 48. EP1K50 Device Interconnect Timing Microparameters* *Note (1)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Symbol** | **Speed Grade** | | | | | | **Unit** |
| | **-1** | | **-2** | | **-3** | | |
| | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | |
| $t_{DIN2IOE}$ | | 3.1 | | 3.7 | | 4.6 | ns |
| $t_{DIN2LE}$ | | 1.7 | | 2.1 | | 2.7 | ns |
| $t_{DIN2DATA}$ | | 2.7 | | 3.1 | | 5.1 | ns |
| $t_{DCLK2IOE}$ | | 1.6 | | 1.9 | | 2.6 | ns |
| $t_{DCLK2LE}$ | | 1.7 | | 2.1 | | 2.7 | ns |
| $t_{SAMELAB}$ | | 0.1 | | 0.1 | | 0.2 | ns |
| $t_{SAMEROW}$ | | 1.5 | | 1.7 | | 2.4 | ns |
| $t_{SAMECOLUMN}$ | | 1.0 | | 1.3 | | 2.1 | ns |
| $t_{DIFFROW}$ | | 2.5 | | 3.0 | | 4.5 | ns |
| $t_{TWOROWS}$ | | 4.0 | | 4.7 | | 6.9 | ns |
| $t_{LEPERIPH}$ | | 2.6 | | 2.9 | | 3.4 | ns |
| $t_{LABCARRY}$ | | 0.1 | | 0.2 | | 0.2 | ns |
| $t_{LABCASC}$ | | 0.8 | | 1.0 | | 1.3 | ns |

| *Table 49. EP1K50 External Timing Parameters* *Note (1)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Symbol** | **Speed Grade** | | | | | | **Unit** |
| | **-1** | | **-2** | | **-3** | | |
| | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | |
| $t_{DRR}$ | | 8.0 | | 9.5 | | 12.5 | ns |
| $t_{INSU}$ *(2)* | 2.4 | | 2.9 | | 3.9 | | ns |
| $t_{INH}$ *(2)* | 0.0 | | 0.0 | | 0.0 | | ns |
| $t_{OUTCO}$ *(2)* | 2.0 | 4.3 | 2.0 | 5.2 | 2.0 | 7.3 | ns |
| $t_{INSU}$ *(3)* | 2.4 | | 2.9 | | – | | ns |
| $t_{INH}$ *(3)* | 0.0 | | 0.0 | | – | | ns |
| $t_{OUTCO}$ *(3)* | 0.5 | 3.3 | 0.5 | 4.1 | – | – | ns |
| $t_{PCISU}$ | 2.4 | | 2.9 | | – | | ns |
| $t_{PCIH}$ | 0.0 | | 0.0 | | – | | ns |
| $t_{PCICO}$ | 2.0 | 6.0 | 2.0 | 7.7 | – | – | ns |

Tables 51 through 57 show EP1K100 device internal and external timing parameters.

*Table 51. EP1K100 Device LE Timing Microparameters* *Note (1)*

| Symbol | Speed Grade | | | | | | Unit |
|---|---|---|---|---|---|---|---|
| | -1 | | -2 | | -3 | | |
| | Min | Max | Min | Max | Min | Max | |
| $t_{LUT}$ | | 0.7 | | 1.0 | | 1.5 | ns |
| $t_{CLUT}$ | | 0.5 | | 0.7 | | 0.9 | ns |
| $t_{RLUT}$ | | 0.6 | | 0.8 | | 1.1 | ns |
| $t_{PACKED}$ | | 0.3 | | 0.4 | | 0.5 | ns |
| $t_{EN}$ | | 0.2 | | 0.3 | | 0.3 | 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 |
| $t_{CASC}$ | | 0.6 | | 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 |

The $I_{CCACTIVE}$ value can be calculated with the following equation:

$I_{CCACTIVE} = K \times \mathbf{f_{MAX}} \times N \times \mathbf{tog_{LC}}$ (µA)

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 58 provides the constant (K) values for ACEX 1K devices.

***Table 58. ACEX 1K Constant Values***

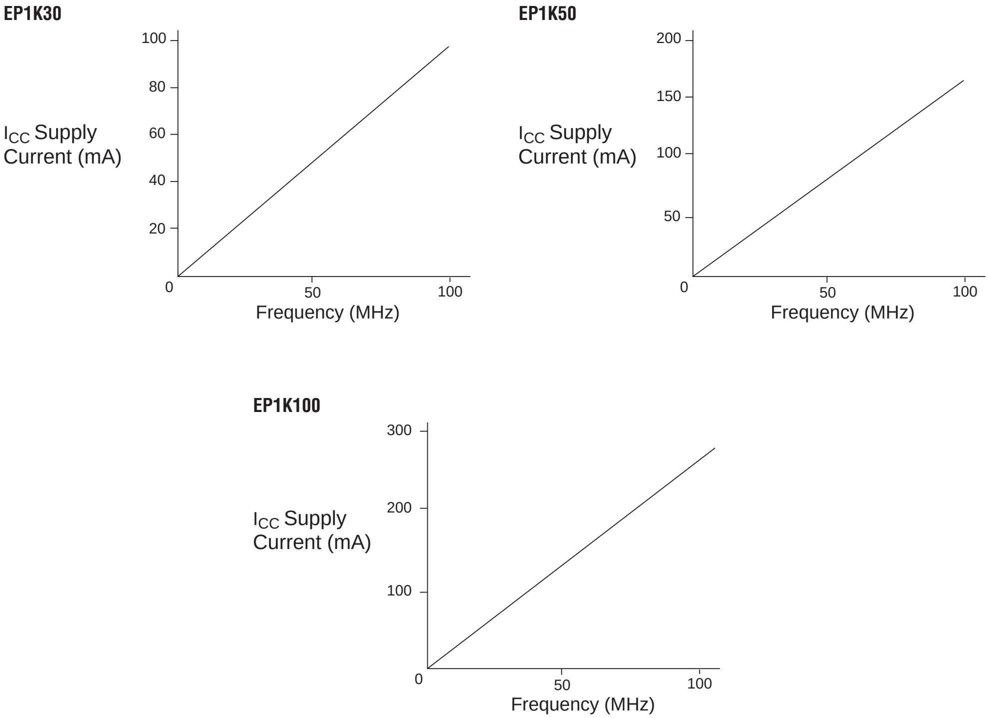
| Device | K Value |
|---|---|
| EP1K10 | 4.5 |
| EP1K30 | 4.5 |
| EP1K50 | 4.5 |
| EP1K100 | 4.5 |

This supply power 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 ACEX 1K devices assumes that LEs drive FastTrack Interconnect channels. In contrast, the power model of segmented FPGAs assumes that all LEs drive only one short interconnect segment. This assumption may lead to inaccurate results when compared to measured power consumption for actual designs in segmented FPGAs.

Figure 31 shows the relationship between the current and operating frequency of ACEX 1K devices. For information on other ACEX 1K devices, contact Altera Applications at (800) 800-EPLD.

*Figure 31. ACEX 1K $I_{CCACTIVE}$ vs. Operating Frequency*




# Configuration & Operation

The ACEX 1K architecture supports several configuration schemes. This section summarizes the device operating modes and available device configuration schemes.

## Operating Modes

The ACEX 1K 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 ACEX 1K POR time does not exceed 50 µs.

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