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	-
Number of Logic Elements/Cells	-
Total RAM Bits	-
Number of I/O	-
Number of Gates	-
Voltage - Supply	-
Mounting Type	Through Hole
Operating Temperature	-
Package / Case	599-BCPGA
Supplier Device Package	599-PGA (62.5x62.5)
Purchase URL	https://www.e-xfl.com/product-detail/intel/epf10k250egc599-2

Table 6 shows FLEX 10KE performance for more complex designs. These designs are available as Altera MegaCore™ functions.

Table 6. FLEX 10KE Performance for Complex Designs

Application	LEs Used	Performance			Units
		-1 Speed Grade	-2 Speed Grade	-3 Speed Grade	
16-bit, 8-tap parallel finite impulse response (FIR) filter	592	138	128	103	MSPS
8-bit, 512-point fast Fourier transform (FFT) function	1,845	60	72	83	µs
		76	66	56	MHz
a16450 universal asynchronous receiver/transmitter (UART)	479	42	39	34	MHz

The FLEX 10KE architecture is similar to that of embedded gate arrays, which is the fastest-growing segment of the gate array market. As with standard gate arrays, embedded gate arrays implement general logic in a conventional "sea-of-gates" architecture. Additionally, embedded gate arrays have dedicated die areas for implementing large, specialized functions. By embedding functions in silicon, embedded gate arrays provide reduced die area and increased speed compared to standard gate arrays. While embedded megafunctions typically cannot be customized, FLEX 10KE devices are programmable, providing the designer with full control over embedded megafunctions and general logic while facilitating iterative design changes during debugging.

Each FLEX 10KE 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), microcontroller, wide-data-path manipulation, and data-transformation functions. The logic array performs the same function as the sea-of-gates in the gate array: it 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.

FLEX 10KE devices are configured at system power-up with data stored in an Altera serial Configuration EPROM device or provided by a system controller. Altera offers the EPC1, EPC2, and EPC1441 Configuration EPROMs, which configure FLEX 10KE devices via a serial data stream. Configuration data can also be downloaded from system RAM or from the Altera BitBlaster™ serial download cable, ByteBlaster™ parallel port download cable, or ByteBlasterMV™ parallel port download cable. After a FLEX 10KE device has been configured, it can be reconfigured in-circuit by resetting the device and loading new data. Because reconfiguration requires less than 330 ms, real-time changes can be made during system operation.

FLEX 10KE devices contain an optimized interface that permits microprocessors to configure FLEX 10KE devices serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat a FLEX 10KE device as memory and configure the device by writing to a virtual memory location, making it very easy for the designer to reconfigure the device.

For more information, go to the following documents:

- *Configuration EPROMs for FLEX Devices Data Sheet*
- *BitBlaster Serial Download Cable Data Sheet*
- *ByteBlaster Parallel Port Download Cable Data Sheet*
- *ByteBlasterMV Parallel Port Download Cable Data Sheet*

FLEX 10KE devices are supported by the Altera MAX+PLUS II development system, a single, integrated package that offers schematic, text—including AHDL—and waveform design entry; compilation and logic synthesis; full simulation and worst-case timing analysis; and device configuration. The MAX+PLUS II software provides EDIF 2 0 0 and 3 0 0, LPM, VHDL, Verilog HDL, and other interfaces for additional design entry and simulation support from other industry-standard PC- and UNIX workstation-based EDA tools.

The MAX+PLUS II software works easily with common gate array EDA tools for synthesis and simulation. For example, the MAX+PLUS II software can generate Verilog HDL files for simulation with tools such as Cadence Verilog-XL. Additionally, the MAX+PLUS II software contains EDA libraries that use device-specific features such as carry chains, which are used for fast counter and arithmetic functions. For instance, the Synopsys Design Compiler library supplied with the MAX+PLUS II development system includes DesignWare functions that are optimized for the FLEX 10KE architecture.

The MAX+PLUS II software runs on 486- and Pentium-based PCs, Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations.

Go to the *MAX+PLUS II Programmable Logic Development System & Software Data Sheet* in the ***1998 Data Book*** for more information.

Functional Description

Each FLEX 10KE device contains an enhanced embedded array to implement memory and specialized logic functions, and a logic array to implement general logic.

The embedded array consists of a series of EABs. When implementing memory functions, each EAB provides 4,096 bits, which can be used to create RAM, ROM, dual-port RAM, or first-in first-out (FIFO) functions. When implementing logic, each EAB can contribute 100 to 600 gates towards complex logic functions, such as multipliers, microcontrollers, state machines, and DSP functions. EABs can be used independently, or multiple EABs can be combined to implement larger functions.

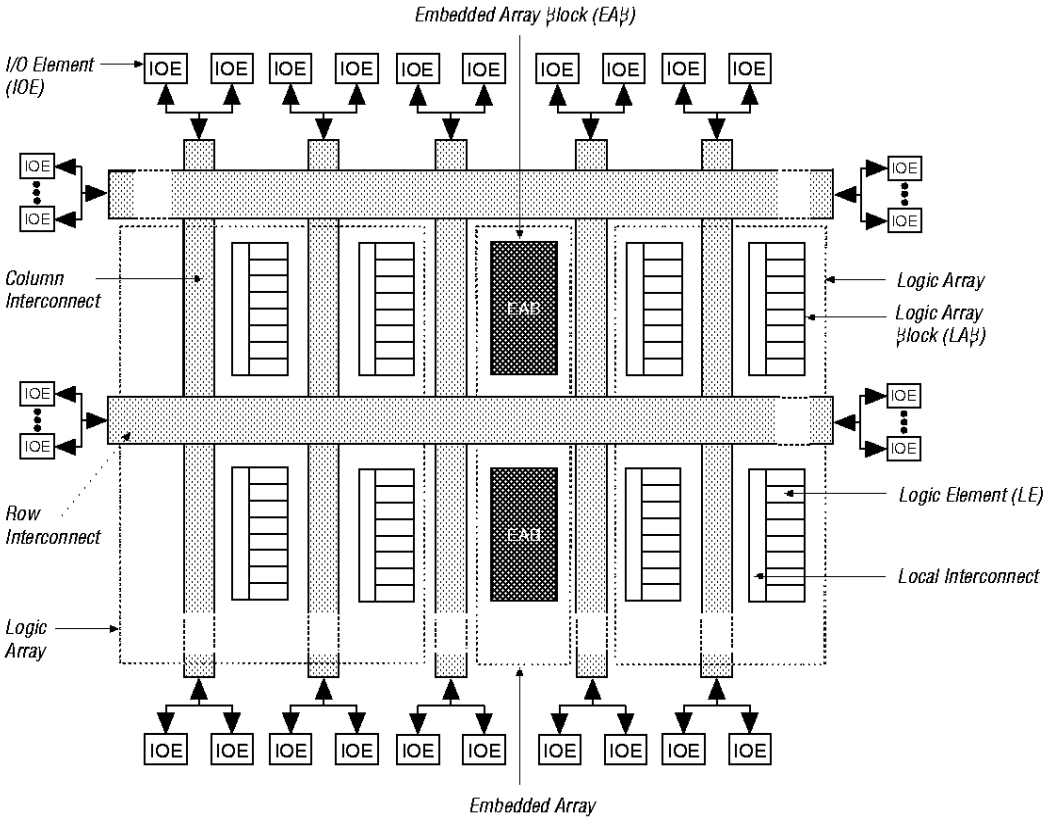
The logic array consists of logic array blocks (LABs). Each LAB contains eight LEs and a local interconnect. An LE consists of a 4-input look-up table (LUT), a programmable flipflop, and dedicated signal paths for carry and cascade functions. The eight LEs can be used to create medium-sized blocks of logic—such as 8-bit counters, address decoders, or state machines—or combined across LABs to create larger logic blocks. Each LAB represents about 96 usable gates of logic.

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

Each I/O pin is fed by an I/O element (IOE) located at the end of each row and column of the FastTrack Interconnect routing structure. Each IOE contains a bidirectional I/O buffer and a flipflop that can be used as either an output or input register to feed input, output, or bidirectional signals. When used with a dedicated clock pin, these registers provide exceptional performance. As inputs, they provide setup times as low as 2.9 ns and hold times of 0 ns; as outputs, these registers provide clock-to-output times as low as 4.3 ns. IOEs provide a variety of features, such as JTAG BST support, slew-rate control, tri-state buffers, and open-drain outputs.

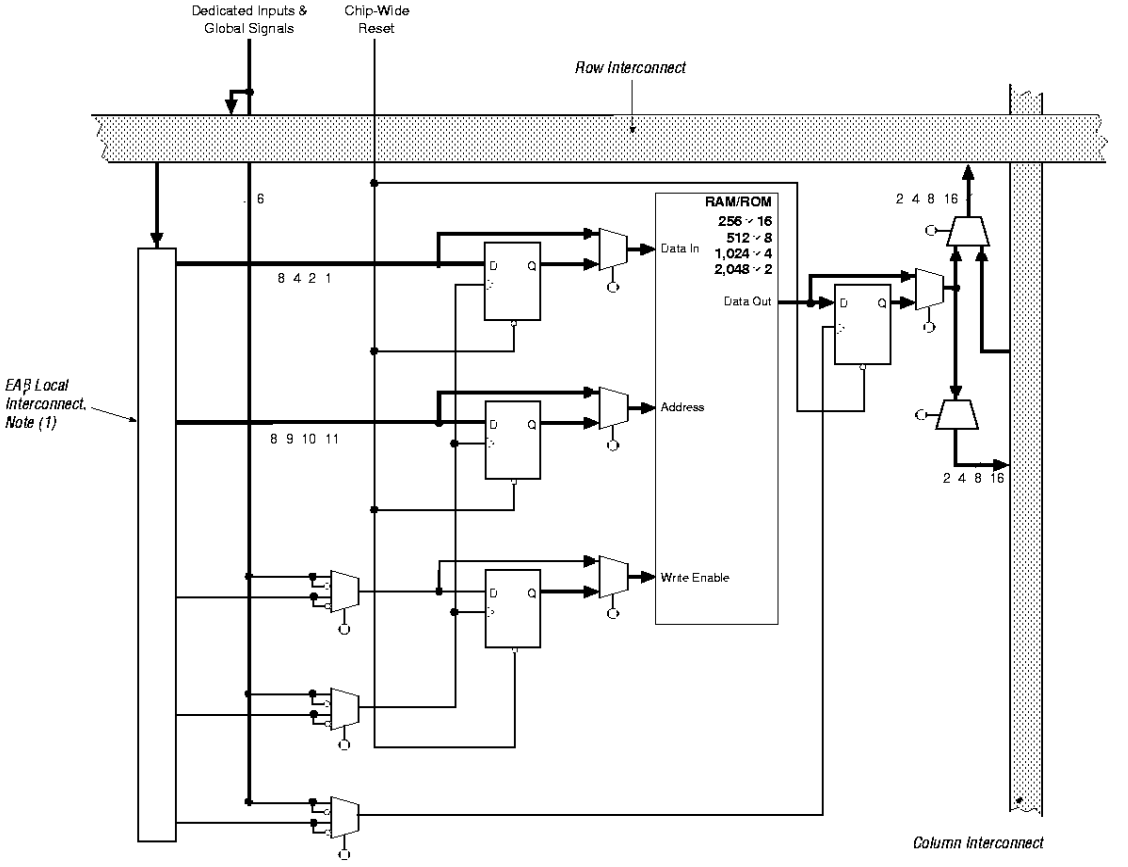
Figure 1 shows a block diagram of the FLEX 10KE architecture. Each group of LEs is combined into an LAB; LABs are arranged into rows and columns. Each row also contains a single EAB. The LABs and EABs are interconnected by the FastTrack Interconnect routing structure. IOEs are located at the end of each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

Figure 1. FLEX 10KE Device Block Diagram



FLEX 10KE devices provide six dedicated inputs that drive the control inputs of the flipflops 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 routing structure. 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.

Figure 3. FLEX 10KE Device in Single-Port EAB



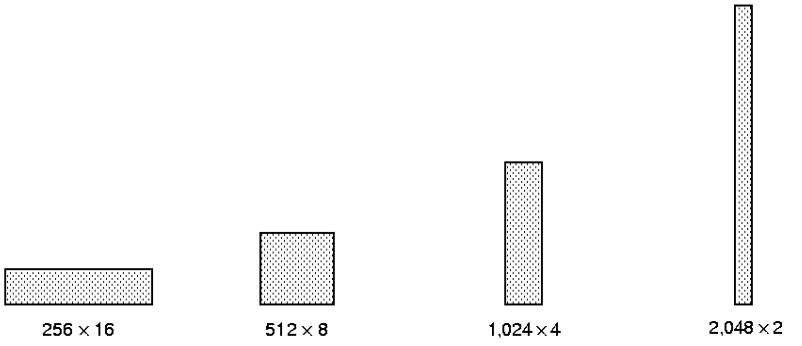
Note:

(1) EPF10K30E and EPF10K50E devices have 88 EAB local interconnect channels. EPF10K100E, EPF10K100B, EPF10K130E, EPF10K200E, and EPF10K250E devices have 104 EAB local interconnect channels.

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 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.

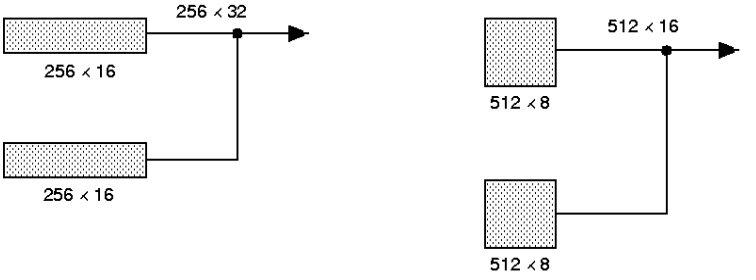
When used as RAM, each EAB can be configured in any of the following sizes: 256 × 16, 512 × 8, 1,024 × 4, or 2,048 × 2 (see Figure 4).

Figure 4. FLEX 10KE EAβ 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; two 512 × 8 RAM blocks can be combined to form a 512 × 16 block (see Figure 5).

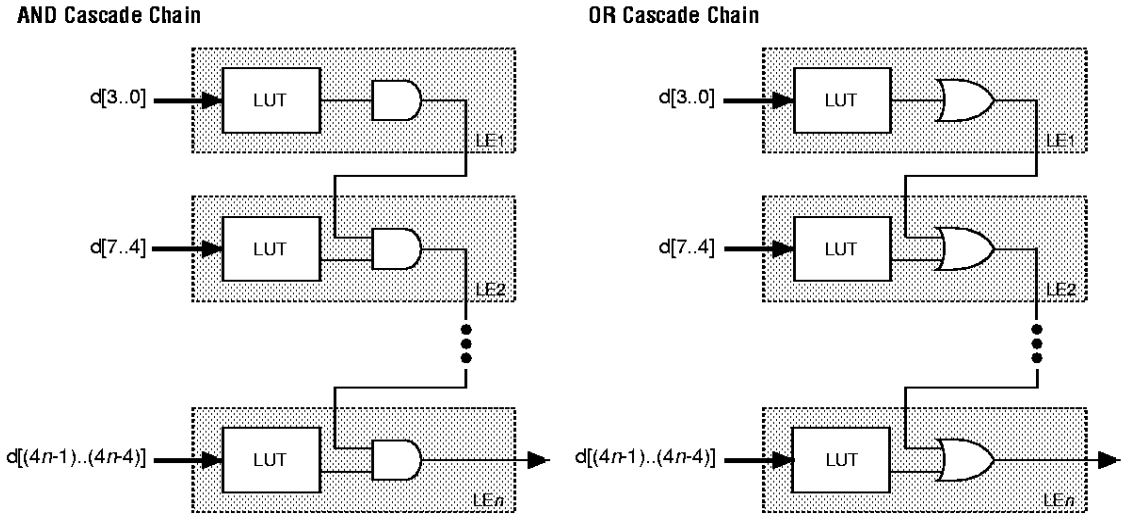
Figure 5. Examples of Combining FLEX 10KE EABs



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. The Altera MAX+PLUS II 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, WE signals, read address, and RE signals. The global signals and the EAB local interconnect can drive WE, RE, 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 WE, RE, clear, clock, and clock enable signals.

Figure 9. FLEX 10KE Cascade Chain Operation



LE Operating Modes

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

- Normal mode
- Arithmetic mode
- Up/down counter mode
- Clearable counter mode

Each of these modes uses LE resources differently. In each mode, seven available inputs to the LE—the four data inputs from the LAB local interconnect, the feedback from the programmable register, and the carry-in and cascade-in from the previous LE—are directed to different destinations to implement the desired logic function. Three inputs to the LE provide clock, clear, and preset control for the register. The MAX+PLUS II software, in conjunction with parameterized functions such as LPM and DesignWare functions, automatically chooses the appropriate mode for common functions such as counters, adders, and multipliers. If required, the designer can also create special-purpose functions to use an LE operating mode for optimal performance.

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

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. 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 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 MAX+PLUS II 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 MAX+PLUS II 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

Asynchronous Clear

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

Asynchronous Preset

An asynchronous preset is implemented as an asynchronous load, or with an asynchronous clear. If DATA3 is tied to VCC, asserting LABCTRL1 asynchronously loads a one into the register. Alternatively, the MAX+PLUS II software can provide preset control by using the clear and inverting the input and output of the register. Inversion control is available for the inputs to both LEs and IOEs. Therefore, if a register is preset by only one of the two LABCTRL signals, the DATA3 input is not needed and can be used for one of the LE operating modes.

Asynchronous Preset & Clear

When implementing asynchronous clear and preset, LABCTRL1 controls the preset and LABCTRL2 controls the clear. DATA3 is tied to VCC, so that asserting LABCTRL1 asynchronously loads a one into the register, effectively presetting the register. Asserting LABCTRL2 clears the register.

Asynchronous Load with Clear

When implementing an asynchronous load in conjunction with the clear, LABCTRL1 implements the asynchronous load of DATA3 by controlling the register preset and clear. LABCTRL2 implements the clear by controlling the register clear; LABCTRL2 does not have to feed the preset circuits.

Asynchronous Load with Preset

When implementing an asynchronous load in conjunction with preset, the MAX+PLUS II 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 MAX+PLUS II 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.

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 a LAB-by-LAB basis.

The incoming signal may be inverted at the dedicated clock pin and will drive all IOEs. To use the true and complement of a clock 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, then two signals on the peripheral control bus are consumed, one for each sense of the clock.

When dedicated inputs drive non-inverted and inverted peripheral clears, clock enables, and output enables, two signals on the peripheral control bus will be used.

Tables 8 and 9 list the sources for each peripheral control signal, and show how the output enable, clock enable, clock, and clear signals share 12 peripheral control signals. The tables also show the rows that can drive global signals.

Table 8. EPF10K30E & EPF10K50E Peripheral Bus Sources

Peripheral Control Signal	EPF10K30E	EPF10K50E
OE0	Row A	Row A
OE1	Row B	Row B
OE2	Row C	Row D
OE3	Row D	Row F
OE4	Row E	Row H
OE5	Row F	Row J
CLKENA0/CLK0/GLOBAL0	Row A	Row A
CLKENA1/OE6/GLOBAL1	Row B	Row C
CLKENA2/CLR0	Row C	Row E
CLKENA3/OE7/GLOBAL2	Row D	Row G
CLKENA4/CLR1	Row E	Row I
CLKENA5/CLK1/GLOBAL3	Row F	Row J

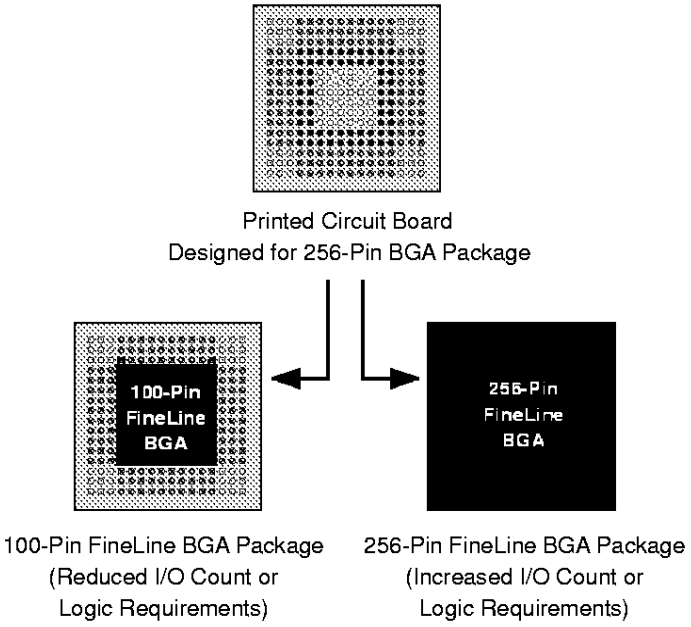
Table 9. EPF10K100β, EPF10K100E, EPF10K130E, EPF10K200E & EPF10K250E Peripheral βus Sources

Peripheral Control Signal	EPF10K100β EPF10K100E	EPF10K130E	EPF10K200E	EPF10K250E
OE0	Row A	Row C	Row G	Row E
OE1	Row C	Row E	Row I	Row G
OE2	Row E	Row G	Row K	Row I
OE3	Row L	Row N	Row R	Row P
OE4	Row I	Row K	Row O	Row M
OE5	Row K	Row M	Row Q	Row O
CLKENA0/CLK0/GLOBAL0	Row F	Row H	Row L	Row J
CLKENA1/OE6/GLOBAL1	Row D	Row F	Row J	Row H
CLKENA2/CLR0	Row B	Row D	Row H	Row F
CLKENA3/OE7/GLOBAL2	Row H	Row J	Row N	Row L
CLKENA4/CLR1	Row J	Row L	Row P	Row N
CLKENA5/CLK1/GLOBAL3	Row G	Row I	Row M	Row K

Signals on the peripheral control bus can also drive the four global signals, referred to as GLOBAL0 through GLOBAL3 in Tables 8 and 9. An internally generated signal can drive a global signal, providing the same low-skew, low-delay characteristics as a signal driven by an input pin. An LE drives the global signal by driving a row line that drives the peripheral bus, which then drives the global signal. This feature is ideal for internally generated clear or clock signals with high fan-out. However, internally driven global signals offer no advantage over the general-purpose interconnect for routing data signals.

The chip-wide output enable pin is an active-low pin that can be used to tri-state all pins on the device. This option can be set in the MAX+PLUS II software. The registers in the IOE can also be reset by the chip-wide reset pin.

Figure 17. SameFrame Pin-Out Example



Output Configuration

This section discusses the peripheral component interconnect (PCI) pull-up clamping diode option, slew-rate control, open-drain output option, and MultiVolt I/O interface for FLEX 10KE devices. The PCI pull-up clamping diode, slew-rate control, and open-drain output options are controlled pin-by-pin via MAX+PLUS II logic options. The MultiVolt I/O interface is controlled by connecting VCCIO to a different voltage. Its effect can be simulated in the MAX+PLUS II software via a **Global Project Device Options** command (Assign menu).

PCI Pull-up Clamping Diode Option

FLEX 10KE devices have a pull-up clamping diode on every I/O, dedicated input, and dedicated clock pin. PCI clamping diodes clamp the signal to the V_{CCIO} value and are required for 3.3-V PCI compliance. Clamping diodes can also be used to limit overshoot in other systems.

Clamping diodes are controlled pin-by-pin. When V_{CCIO} is 3.3 V, a pin that has the clamping diode option turned on can be driven by a 2.5-V or 3.3-V signal, but not a 5.0-V signal. When V_{CCIO} is 2.5 V, a pin that has the clamping diode option turned on can be driven by a 2.5-V signal, but not a 3.3-V or 5.0-V signal. Additionally, a clamping diode can be activated for a subset of pins, which would allow a device to bridge between a 3.3-V PCI bus and a 5.0-V device.

FLEX 10KE devices are generally pin-compatible with the equivalent FLEX 10KA device. In some cases, FLEX 10KE devices have fewer I/O pins than FLEX 10KA devices. Table 17 shows which FLEX 10KE devices have fewer I/Os. However, power, ground, JTAG, and configuration pins are the same on FLEX 10KA and FLEX 10KE devices, enabling migration from a FLEX 10KA design to a FLEX 10KE design.

Table 17. I/O Count on FLEX 10KA & FLEX 10KE Devices

FLEX 10KA		FLEX 10KE	
Device	**I/O Count**	**Device**	**I/O Count**
EPF10K30AF256	191	EPF10K30EF256	176
EPF10K30AF484	244	EPF10K30EF484	218
EPF10K30AB356	246	EPF10K30EB356	220
EPF10K50VF484	300	EPF10K50EF484	254
EPF10K50VB356	274	EPF10K50EB356	256
EPF10K100AF484	366	EPF10K100EF484	337
EPF10K130VB600	470	EPF10K130EB600	426

Configuration Schemes

The configuration data for a FLEX 10KE device can be loaded with one of five configuration schemes (see Table 18), chosen on the basis of the target application. An EPC2, EPC1, or EPC1441 Configuration EPROM, intelligent controller, or the JTAG port can be used to control the configuration of a FLEX 10KE device, allowing automatic configuration on system power-up.

Multiple FLEX 10KE 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 FLEX 10K, FLEX 10KA, FLEX 10KE, and FLEX 6000 devices can be configured in the same serial chain.

Table 18. Data Sources for FLEX 10KE Configuration

Configuration Scheme	Data Source
Configuration EPROM	EPC1, EPC2, or EPC1441 Configuration EPROM
Passive serial (PS)	BitBlaster, ByteBlaster, or ByteBlasterMV download cables, or serial data source
Passive parallel asynchronous (PPA)	Parallel data source
Passive parallel synchronous (PPS)	Parallel data source
JTAG	BitBlaster, ByteBlaster, or ByteBlasterMV download cables, or microprocessor with a Jam File

Device Pin-Outs

Table 19 shows the dedicated pin-outs for FLEX 10KE devices in 144-pin TQFP, 208-pin PQFP, 240-pin PQFP, 356-pin BGA, 599-pin PGA, and 600-pin BGA packages.

Table 19. FLEX 10KE Device Pin-Outs (Part 1 of 4) *Note (1)*

Pin Name	144-Pin TQFP EPF10K30E EPF10K50E	208-Pin PQFP EPF10K30E EPF10K50E EPF10K100E EPF10K100β	240-Pin PQFP EPF10K50E EPF10K100E EPF10K100β	240-Pin PQFP EPF10K130E	356-Pin βGA EPF10K100E	599-Pin PGA EPF10K200E EPF10K250E	600-Pin βGA EPF10K200E EPF10K250E
MSEL0 *(2)*	77	108	124	124	D4	F6	F5
MSEL1 *(2)*	76	107	123	123	D3	C3	C1
nSTATUS *(2)*	35	52	60	60	D24	E43	D32
nCONFIG *(2)*	74	105	121	121	D2	B4	D4
DCLK *(2)*	107	155	179	179	AC5	BE5	AP1
CONF_DONE *(2)*	2	2	2	2	AC24	BC43	AM32
INIT_DONE *(3)*	14	19	26	26	T24	AM40	AE32
nCE *(2)*	106	154	178	178	AC2	BB6	AN2
nCEO *(2)*	3	3	3	3	AC22	BF44	AP35
nWS *(4)*	142	206	238	238	AE24	BB40	AR29
nRS *(4)*	141	204	236	236	AE23	BA37	AM28

Table 19. FLEX 10KE Device Pin-Outs (Part 4 of 4) *Note (1)*

Pin Name	144-Pin TQFP EPF10K30E EPF10K50E	208-Pin PQFP EPF10K30E EPF10K50E EPF10K100E EPF10K100β	240-Pin PQFP EPF10K50E EPF10K100E EPF10K100β	240-Pin PQFP EPF10K130E	356-Pin βGA EPF10K100E	599-Pin PGA EPF10K200E EPF10K250E	600-Pin βGA EPF10K200E EPF10K250E
GNDIO	15, 40, 50, 66, 85, 104, 129, 139	20, 32, 48, 59, 72, 91, 124, 130, 152, 171, 188, 201	–	–	–	D24, E9, E15, E21, E27, E33, E39, G7, G41, J5, J43, R5, R43, AA5, AA43, AD4, AD44, AG5, AG43, AN5, AN43, AW5, AW43, BA7, BA41, BC9, BC15, BC21, BC27, BC33, BC39, BD24	C8, E12, C15, A20, C23, A27, AM26, AR23, AM19, AN15, AL12, AN8, C2, C3, C4, D5, E5, C33, C32, D31, E31, AL5, AM5, AN4, AN3, AM31, AN32, AN33, AP34
Total User I/O Pins *(7)*	102	147	189	186	274	470	470

Notes:
(1) All pins that are not listed are user I/O pins.
(2) This pin is a dedicated pin; it is not available as a user I/O pin.
(3) This pin can be used as a user I/O pin if it is not used for its device-wide or configuration function.
(4) This pin can be used as a user I/O pin after configuration.
(5) This pin is tri-stated in user mode.
(6) The optional JTAG pin TRST is not used in the 144-pin TQFP package.
(7) The user I/O pin count includes dedicated input pins, dedicated clock pins, and all I/O pins.

Table 20. FLEX 10KE FineLine βGA Device Pin-Outs (Part 2 of 4) *Note (1)*

Pin Name	256-Pin FineLine βGA EPF10K30E	256-Pin FineLine βGA EPF10K50E EPF10K100E EPF10K100β	672-Pin FineLine βGA EPF10K130E	672-Pin FineLine βGA EPF10K200E EPF10K250E
DEV_CLRn *(3)*	D8	D8	J13	J13
DEV_OE *(3)*	C9	C9	H14	H14
VCCINT (2.5 V)	E11, F5, F7, F9, F12, H6, H7, H10, J7, J10, J11, K9, L5, L7, L9, L12, M11, R2	E11, F5, F7, F9, F12, H6, H7, H10, J7, J10, J11, K9, L5, L7, L9, L12, M11, R2	E13, E17, H2, H25, K16, L10, L12, L14, L17, M2, M25, N11, N12, N15, P12, P15, P16, R14, T2, T10,T12, T14, T17, T25, U16, Y7, AA23, AB10, AC14	E13, E17, H2, H25, K16, L10, L12, L14, L17, M2, M25, N11, N12, N15, P12, P15, P16, R14, T2, T10,T12, T14, T17, T25, U16, Y7, AA23, AB10, AC14
VCCIO (2.5 or 3.3 V)	D12, E6, F8, F10, G6, G8, G11, H11, J6, K6, K8, K11, L10, M6, N12	D12, E6, F8, F10, G6, G8, G11, H11, J6, K6, K8, K11, L10, M6, N12	C8, C15, D7, G3, J3, J17, K11, K22, L13, L15, M11, M13, M16, M22, N16, P11, R5, R11, R13, R16, R22, T15, U3, U11, V5, V17, V24, Y2, Y24, AA26, AD15	C8, C15, D7, G3, J3, J17, K11, K22, L13, L15, M11, M13, M16, M22, N16, P11, R5, R11, R13, R16, R22, T15, U3, U11, V5, V17, V24, Y2, Y24, AA26, AD15
GND	E5, E12, F6, F11, G7, G9, G10, H8, H9, J8, J9, K7, K10, L6, L11, M5, M12, T8	E5, E12, F6, F11, G7, G9, G10, H8, H9, J8, J9, K7, K10, L6, L11, M5, M12, T8	A2, A25, B2, B25, C3, C10, C24, D3, D4, D19, D23, D24, E4, E23, G23, J5, J23, K4, K10, K17, L11, L16, L22, M5, M12, M14, M15, N13, N14, P13, P14, P22, R12, R15, T11, T16, U10, U17, U24, V3, Y5, AA13, AA22, AB3, AB4, AB5, AB23, AB24, AC3, AC8, AC24, AD13, AD18, AE2, AE25, AF2, AF25	A2, A25, B2, B25, C3, C10, C24, D3, D4, D19, D23, D24, E4, E23, G23, J5, J23, K4, K10, K17, L11, L16, L22, M5, M12, M14, M15, N13, N14, P13, P14, P22, R12, R15, T11, T16, U10, U17, U24, V3, Y5, AA13, AA22, AB3, AB4, AB5, AB23, AB24, AC3, AC8, AC24, AD13, AD18, AE2, AE25, AF2, AF25

Table 21. FLEX 10KE FineLine βGA Device Pin-Outs (Part 3 of 5) Note (1)

Pin Name	484-Pin FineLine βGA EPF10K30E	484-Pin FineLine βGA EPF10K50E	484-Pin FineLine βGA EPF10K100E	484-Pin FineLine βGA EPF10K130E
GND	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W11, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W11, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W11, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W11, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16

Table 23 shows the FLEX 10KE device/package combinations that support SameFrame pin-outs. All FineLine BGA packages support SameFrame pin-outs providing the flexibility to migrate not only from device to device within the same package, but also from one package to another. The I/O count will vary, and MAX+PLUS II software versions 9.1 and higher provide features to help use only the common pins.

Table 23. FLEX 10KE SameFrame Pin-Out Support

Device	256-Pin FineLine BGA	484-Pin FineLine BGA	672-Pin FineLine BGA
EPF10K30E	✓	✓	
EPF10K50E	✓	✓	
EPF10K100B	✓		
EPF10K100E	✓	✓	
EPF10K130E		✓	✓
EPF10K200E			✓
EPF10K250E			✓

Revision History

The information contained in the *FLEX 10KE Embedded Programmable Logic Family Data Sheet* version 1.01 supersedes information published in previous versions.

The *FLEX 10KE Embedded Programmable Logic Family Data Sheet* version 1.01 contains the following change:

- Updated Table 15 (FLEX 10KE external timing parameters).

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
(888) 3-ALTERA
lit_req@altera.com