The Altera Advantage News & Views

Newsletter for Altera Customers ◆ Fourth Quarter ◆ December 1997

Faster FLEX 10K Devices

To meet the increasing performance requirements of system designers, Altera recently unveiled plans for the next generation of programmable logic. Altera introduced two additions to the FLEX® 10K family:

- FLEX 10K-1 and FLEX 10K-2 devices add faster system speeds to existing members of the high-density FLEX 10K family.
- FLEX 10KB devices offer high-density, high-performance devices—based on a 0.25-micron, 5-layer-metal process—at a lower cost.

These new devices provide the low-cost solution to your high-density, high-performance programmable logic needs.

Breakthrough Performance

Process improvements combined with enhancements to the MAX+PLUS® II software have resulted in performance increases of up to 40% for FLEX 10K-2 devices (see Table 1 on page 3). The faster FLEX 10K-1 devices provide over 100% performance improvements. The performance increases respond directly to market demand for faster devices. With these increases, you can use these devices for applications that are experiencing a rapid increase in performance requirements, such as 100-Mbit and 1-Gbit Ethernet communications designs.

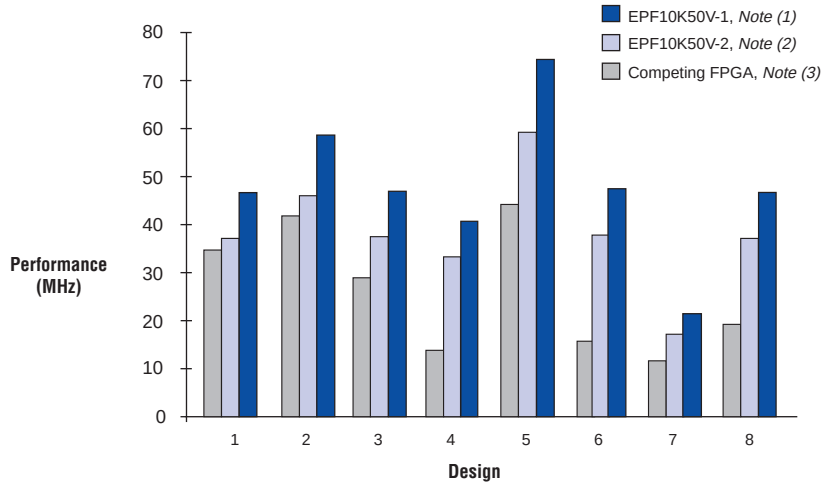
Advanced Process Increases Speed

Altera worked closely with its long-standing partner, TSMC, to improve the die size and performance of FLEX 10K devices. By migrating from a 0.5-micron, three-layer-metal process to a 0.35-micron, four-layer-metal process, Altera can now offer the 3.3-V-based FLEX 10KA devices. This process improvement and smaller die size translate directly to increased performance and lower cost.

FLEX 10K devices outperform competing field-programmable gate arrays (FPGAs) of similar density. In a recent comparison, Altera® Applications implemented 8 customer designs into a FLEX 10K-2 device, a FLEX 10K-1 device, and a competing FPGA device of comparable density. See Figure 1. Designs

continued on page 3

Figure 1. Altera vs. FPGA Performance



Notes:
(1) Performance numbers estimated for the upcomiong release of MAX+PLUS II version 8.2 software.
(2) Performance numbers derived using the MAX+PLUS II version 8.1 software.
(3) These designs were compiled with software from a competing FPGA vendor.

Contents

For information about this newsletter, or to submit questions, contact:

Erica Heidinger, Publisher
Craig Lytle, Technical Editor
101 Innovation Drive
San Jose, CA 95134
Tel: (408) 544-7000
Fax: (408) 544-7809
E-mail: n_v@altera.com

Printed on recycled paper.

I.S. EN ISO 9001

Faster FLEX 10K Devices
continued from page 1

compiled with the MAX+PLUS II software achieved higher performance compared to designs compiled with software from a competing FPGA vendor. In addition, FLEX 10K-2 and FLEX 10K-1 devices achieved a higher average performance:

- FLEX 10K-2 devices averaged 38.36 MHz.
- FLEX 10K-1 devices averaged 47.95 MHz.
- The competing FPGA averaged only 26.44 MHz.

FLEX 10K-1 and FLEX 10K-2 devices operate at a 3.3-V or 5.0-V core voltage and offer the MultiVolt™ I/O interface option, which allows these devices to interface with 2.5-V, 3.3-V, or 5.0-V devices. These devices offer densities of up to 130,000 gates and are available in a variety of package options.

FLEX 10K-1 and FLEX 10K-2 devices are available now and have "-1" and "-2, " respectively, appended to their device ordering codes (for example, EPF10K100ARC240-1). Altera is also shipping -3 and -4 speed grade versions of these devices. Contact Altera Marketing for more information.

MAX+PLUS II Software Improvements

The MAX+PLUS II software offers an architecture-independent development environment, enabling designers to customize their designs for any target device family. The software also provides seamless integration with leading EDA tools, such as tools from Cadence, Exemplar Logic, Mentor Graphics, Synplicity, Synopsys, and Viewlogic, allowing designers to use their preferred design entry methodology.

Improvements to the MAX+PLUS II development system have made a significant contribution to the overall performance increase of FLEX 10K-1 and FLEX 10K-2 devices. Better optimization and routing algorithm improvements have resulted in a 10% to 15% improvement in compilation times for the MAX+PLUS II software. With most compilation times under averaging 30 minutes, you can implement changes and test the results quickly. Moreover, according to benchmark tests performed by Altera Applications, the MAX+PLUS II software compiles designs consistently faster than any other programmable logic development tool. By giving you quick feedback on the utilization and performance of your design, the MAX+PLUS II software helps you bring your product to market faster.

In addition, with MAX+PLUS II version 8.1, designers with a current software maintenance agreement receive upgrades as well as free hardware description language (HDL) synthesis—VHDL or Verilog HDL. For more information, go to "MAX+PLUS II Version 8.1" on page 7.

High-Performance, Low-Cost FLEX 10KB Devices

FLEX 10KB devices will bring higher performance and lower cost to programmable logic. Implemented on a 0.25-micron, five-layer metal process, FLEX 10KB devices offer lower costs, 55- to 75-MHz in-system performance, and densities of up to 250,000 gates. FLEX 10KB devices have a 2.5-V core voltage, resulting in a 75% reduction in power consumption compared to 5.0-V FLEX 10K devices. Further, with the MultiVolt I/O feature, these devices interface with 2.5-V, 3.3-V, or

continued on page 4

Table 1. FLEX 10K Performance Improvements

Device	Speed Grade	Gates	Logic Elements	Embedded RAM	Performance Improvement	Supply Voltage
EPF10K30A	-2	30,000	1,728	12 Kbits	40%, *Note (1)*	3.3 V
EPF10K50V	-1	50,000	2,880	20 Kbits	110%, *Note (2)*	3.3 V
EPF10K50V	-2	50,000	2,880	20 Kbits	40%, *Note (1)*	3.3 V
EPF10K70	-2	70,000	3,744	18 Kbits	22%, *Note (1)*	5.0 V
EPF10K100A	-1	100,000	4,992	24 Kbits	107%, *Note (2)*	3.3 V
EPF10K100A	-2	100,000	4,992	24 Kbits	35%, *Note (1)*	3.3 V
EPF10K130V	-2	130,000	6,656	32 Kbits	38%, *Note (1)*	3.3 V

Notes:
(1) Estimated performance with -2 speed grade using MAX+PLUS II version 8.1 compared to -3 speed grade using MAX+PLUS II version 8.0.
(2) Estimated performance with -1 speed grade using the upcoming release of MAX+PLUS II version 8.2 compared to -3 speed grade using MAX+PLUS II version 8.0.

Faster FLEX 10K Devices
continued from page 3

5.0-V devices. These new devices provide fast time-to-market at unprecedented performance levels and low cost for a broad range of system-level designs.

The FLEX 10KB process migration provides a 20% to 30% performance improvement over existing FLEX 10K devices, enabling programmable logic to support system performance up to 75 MHz. See Figure 2. In addition to higher performance, process migration also provides lower cost. The metal-intensive FLEX architecture efficiently maps metal interconnect to additional process metal layers, allowing Altera to offer smaller die sizes at lower cost.

Altera plans to begin shipping FLEX 10KB devices in the second quarter of 1998. Table 2 shows sample projected volume pricing for FLEX 10KB devices. Table 3 summarizes FLEX 10KB device features.

Figure 2. Sample FLEX 10KB Performance

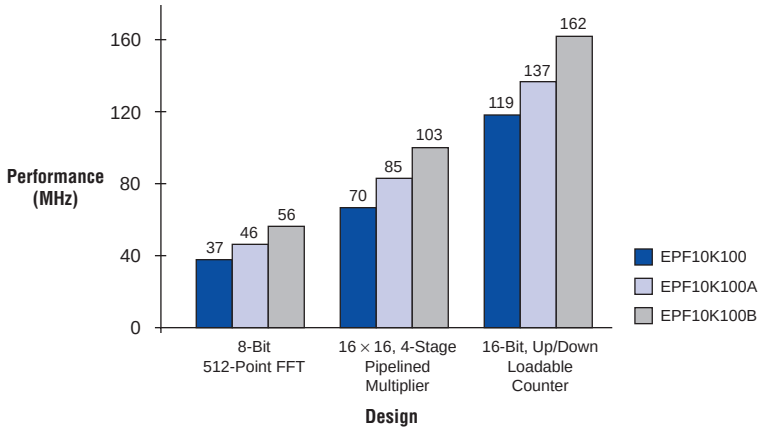


Table 2. Sample FLEX 10KB Volume Price Projections

Device	Gates	Package	Price *Note (1)*
EPF10K30B	30,000	144-Pin TQFP	$8.00
EPF10K50B	50,000	208-Pin PQFP	$10.00
EPF10K100B	100,000	208-Pin PQFP	$20.00

Note:
(1) Projected 50,000-unit volume pricing for the second half of 1999 (North American OEM direct, slowest speed grade).

Table 3. FLEX 10KB Features

Feature	EPF10K30B	EPF10K50B	EPF10K100B	EPF10K130B	EPF10K180B	EPF10K250B
Typical gates	30,000	50,000	100,000	130,000	180,000	250,000
Logic elements	1,728	2,880	4,992	6,656	9,728	12,160
RAM bits	12,288	20,480	24,576	32,768	32,768	40,960
Package options	144-pin TQFP 208-pin QFP 240-pin QFP 356-pin BGA	208-pin QFP 240-pin QFP 356-pin BGA	208-pin QFP 240-pin QFP 356-pin BGA 600-pin BGA	240-pin QFP 356-pin BGA 599-pin PGA 600-pin BGA	240-pin QFP 356-pin BGA 600-pin BGA	356-pin BGA 599-pin PGA 600-pin BGA
Maximum user I/O pins	246	310	406	470	470	470
Availability	Second half 1998	Second half 1998	Second quarter of 1998	Second half 1998	Second half 1998	Second half 1998

Devices & TOOLS

FLEX Update

EPF10K100A Devices Available

In November 1997, Altera began shipping EPF10K100A devices in 240-pin power quad flat pack (RQFP) packages. Altera plans to ship 356-pin ball-grid array (BGA) packages in December 1997 and 600-pin BGA packages in the first quarter of 1998.

Built on a 0.35-mm, quad-layer metal process, EPF10K100A devices are pin-compatible with FLEX 10K devices in 240-pin RQFP and 356-pin BGA packages. For instance, you can migrate from 3.3-V EPF10K50V devices to the higher-density EPF10K100A devices without changing your board layout.

100,000-gate EPF10K100ARC240-3 devices are priced at $165 each for 100-unit quantities. The new EPF10K100ARC240-2 and EPF10K100ARC240-1 are also available; contact your local Altera representative for pricing information.

EPF10K250A Coming in First Quarter 1998

Altera plans to make initial shipments of the 250,000-gate EPF10K250A devices in March 1998. With 12,160 logic elements and 20 embedded array blocks (40,960 memory bits), the EPF10K250A will be the largest programmable logic device available.

EPF10K250A devices are expected to be available in 599-pin ceramic pin-grid-array (PGA) and 600-pin ball-grid array (BGA) packages. It will also be pin-compatible with other FLEX 10K devices in 599-pin PGA and 600-pin BGA packages.

FLEX 6000 Device Availability

The FLEX® 6000 programmable logic device (PLD) family provides a low-cost alternative to high-volume gate array designs and is the industry's most cost-effective PLD family. The FLEX 6000 family offers extensive package options, including 144-pin thin quad flat pack (TQFP), 208- and 240-pin plastic quad flat pack (PQFP), and 256-pin ball-grid array (BGA) packages. Table 1 describes package options, speed grades, and availability for each FLEX 6000 device.

Table 1. FLEX 6000 Availability

Device	Package	Speed Grades	Availability
EPF6016	144-pin TQFP	-2, -3	Now
	208-pin PQFP	-2, -3	Now
	240-pin PQFP	-2, -3	Now
	256-pin BGA	-2, -3	Now
EPF6016A	144-pin TQFP	-2, -3	Q3 1998
	208-pin PQFP	-2, -3	Q3 1998
	240-pin PQFP	-2, -3	Q3 1998
	256-pin BGA	-2, -3	Q3 1998
EPF6024A	208-pin PQFP	-2, -3	Q1 1998
	240-pin PQFP	-2, -3	Q1 1998
	256-pin BGA	-2, -3	Q1 1998

Configuration EPROM Update

The EPC1441 Device: A Low-Cost Configuration EPROM

The EPC1441 device—the newest Configuration EPROM device in the Altera® serial EPROM family—is the most cost-effective serial Configuration EPROM in the industry. EPC1441 devices have a density of 441 Kbits and offer 3.3- or 5.0-V operation. A single EPC1441 device can configure FLEX devices up to 30,000 gates, including:

- All FLEX 6000 devices
- All FLEX 8000 devices
- EPF10K10, EPF10K20, and EPF10K30 FLEX 10K devices

EPC1441 devices are available in 8-pin plastic dual in-line (PDIP), 20-pin plastic J-lead chip carrier (PLCC), and 32-pin thin quad flat pack (TQFP) packages. EPC1441 devices cost $3.50 each for 100-unit quantities.

For more information on EPC1441 devices or other Altera Configuration EPROMs, see the *Configuration EPROMs for FLEX Devices Data Sheet*.

MAX Update

MAX 9000 Update

Altera is performing a process migration on MAX® 9000 devices. The new devices will be fabricated

continued on page 6

Devices & Tools, continued from page 5

on a 0.5-micron, triple-layer-metal process. With propagation delays as fast as 7.5 ns, the new MAX 9000A devices offer a significant performance enhancement over the existing MAX 9000 devices. Production quantities of high-performance MAX 9000A devices will be available beginning in January 1998; samples of 10-ns EPM9560A devices are available now. Table 2 shows MAX 9000 device availability.

Table 2. MAX 9000 Device Availability

Device	t_{PD} (ns)	Availability	84-Pin PLCC	208-Pin RQFP	240-Pin RQFP	356-Pin BGA
EPM9320A	10	February 1998	✓	✓		✓
EPM9480A	10	Second Half 1998		✓	✓	
EPM9560A	10	January 1998		✓	✓	✓

ATE Support Strategy

Devices that support in-system programmability (ISP), such as MAX 9000 and MAX 7000S devices, can be programmed during the final printed circuit board (PCB) testing stage using automated test equipment (ATE). This programming method offers the ability to combine device programming with board-level test methods. To program using ATE, designers must use fixed-algorithm devices, which have an ordering code ending in "F" (e.g., EPM9560RC208-15F). Both MAX 9000 and MAX 7000S fixed-algorithm devices are available today.

Faster MAX 7000S Speed Grade Devices

At any macrocell count, the MAX 7000S family is the fastest product-term-based programmable logic device (PLD) family in the world. The MAX 7000S family now offers even faster performance with devices as fast as 5 ns. The EPM7064S is now shipping with a new 5-ns speed grade. The EPM7128S and EPM7256S devices are available in -6 and -7 speed grades, respectively. Table 3 shows speed grades and availability of MAX 7000S devices.

Table 3. MAX 7000S Device Availability

Device	Package	Speed Grade	Availability
EPM7032S	44-pin PLCC	-6, -7, -10	March 1998
	44-pin TQFP	-6, -7, -10	March 1998
EPM7064S	44-pin PLCC	-5, -6, -7, -10	Now
	44-pin TQFP	-5, -6, -7, -10	Now
	84-pin PLCC	-5, -7, -10	December 1997
	100-pin TQFP	-5, -6, -7, -10	Now
EPM7128S	84-pin PLCC	-6, -7, -10, -15	Now
	100-pin TQFP	-6, -7, -10, -15	Now
	100-pin PQFP	-6, -7, -10, -15	Now
	160-pin PQFP	-6, -7, -10, -15	Now
EPM7160S	160-pin PQFP	-6, -10, -15	May 1998
EPM7192S	160-pin PQFP	-7, -10, -15	Now
EPM7256S	208-pin PQFP	-7, -10, -15	Now

MAX 7000A Family

The MAX 7000A device family (formerly known as Michelangelo), which supports 3.3-V in-system programmability (ISP), will begin shipping in March 1998. Manufactured on a 0.35-micron, quad-layer-metal EEPROM process, MAX 7000A devices are pin-compatible with MAX 7000S devices and have the MultiVolt™ I/O feature, enabling these devices to interface with 5.0-V, 3.3-V, or 2.5-V devices. Table 4 summarizes MAX 7000A device availability.

MAX 7000 Product Transitions

Altera is migrating existing MAX 7000 devices from a 0.65-micron process to a 0.5-micron process. Evaluation packets containing sample devices and documentation are available from your local Altera sales representative. Table 5 on page 7 outlines the process migration schedule.

Table 4. MAX 7000A Device Availability

Device	t_{PD} (ns)	Availability	44-Pin PLCC	44-Pin TQFP	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	208-Pin PQFP	256-Pin BGA
EPM7032A	5.0	Second half 1998	✓	✓					
EPM7064A	5.0	Second half 1998	✓	✓	✓	✓			
EPM7128A	6.0	March 1998			✓	✓	✓		
EPM7256A	6.0	Second quarter 1998				✓	✓	✓	✓
EPM7384A	7.5	1999					✓	✓	✓
EPM7512A	7.5	Second half 1998					✓	✓	✓
EPM71024A	7.5	1999						✓	✓

Table 5. MAX 7000 Migration Schedule Note (1)			
Device	**Reference** *Note (2)*	**Date**	**Process**
EPM7064	PCN9703	Complete	0.5-micron
EPM7064S	ADV9708	Complete	0.5-micron
EPM7128E	PCN9703 ADV9708	January 1, 1998	0.5-micron
EPM7128S	PCN9703 ADV9708	Complete	0.5-micron
EPM7192E EPM7192S	PCN9703 ADV9708	February 1, 1998	0.5-micron
EPM7256S EPM7256E	PCN9703 ADV9708	Complete	0.5-micron

Notes:
(1) This process transition will not result in any changes to data sheet parameters or ordering codes.
(2) Altera provides advisories and process change notices. Go to the Altera world-wide web site at **http://www.altera.com** for these reference documents.

MAX 5000 & Classic Product Transitions

Altera's migration of MAX 5000 and Classic™ devices to a 0.65-micron process is complete. See Table 6 for more details.

Table 6. MAX 5000 & Classic Product Migration Schedule		
Description *(1)*	**Reference** *(2)*	**Device**
MAX 5000 devices fabricated on a 0.65-micron process *Note (3)*	PCN 9407 ADV 9515 ADV 9606	EPM5032 EPM5064 EPM5128 EPM5130 EPM5192
Classic devices fabricated on a 0.65-micron process	PCN 9510 ADV 9607 ADV 9621	EP6*xx* EP9*xx* EP18*xx*

Notes:
(1) Data sheet parameters and ordering codes will not change.
(2) Go to the Altera world-wide web site for advisories and process change notices.
(3) Devices manufactured on the 0.65-micron process must be programmed with new programming adapters.

MAX+PLUS II Updates

Microperipheral MegaCore Library: A Cost-Effective Alternative

Designing with existing intellectual property just became much more cost-effective. Instead of licensing an entire library of functions, Altera microperipheral MegaCore™ functions can be licensed individually and downloaded directly from the Altera world-wide web site at **http://www.altera.com**. Contact your local Altera sales representative to license the following MegaCore functions (ordering codes are shown in parentheses):

- `a8237` programmable DMA controller (PLSM-8237)
- `a8251` programmable communications interface (PLSM-8251)
- `a8255` programmable peripheral interface adapter (PLSM-8255)
- `a6402` universal asynchronous receiver/transmitter (PLSM-6402)
- `a16450` universal asynchronous receiver/transmitter (PLSM-16450)
- `a6850` asynchronous communications interface adapter (PLSM-6850)
- `a8259` programmable interrupt controller (PLSM-8259)

MAX+PLUS II Version 8.1

The Altera® MAX+PLUS® II software, the most powerful programmable logic development system in the industry, is regularly updated to ensure state-of-the-art support for all Altera device families. The Altera maintenance program allows you to automatically benefit from Altera's ongoing product development. With a current software maintenance agreement, you receive support for the latest Altera devices as well as new software features, performance enhancements, and up-to-date online documentation.

Altera has recently added a variety of key features to MAX+PLUS II version 8.1, such as new synthesis options and improved compilation times, to help you be more productive and create more successful designs. For example, Altera now offers HDL synthesis at no additional cost. Starting with MAX+PLUS□II version 8.1, VHDL or Verilog HDL synthesis is

continued on page 8

Devices & Tools, continued from page 7

available as a standard feature for designers with a current software maintenance agreement. VHDL synthesis is available in MAX+PLUS II version 8.1. If you have a current maintenance agreement, you can enable synthesis support by obtaining a new authorization code from the Altera world-wide web (WWW) site at **http://www.altera.com/authcode**.

Verilog HDL synthesis will be available in MAX+PLUS II version 8.2 and will only be available for 32-bit operating systems, such as Windows 95, Windows NT, and UNIX. You can download an authorization code for Verilog HDL today, and Altera will send you the software as soon as it is available. Table 7 shows the free feature upgrades and new standard features available in MAX+PLUS II version 8.1.

Limited-Time Offer on Altera Tools

Altera is offering a world-wide tools promotion for all ES Site License users and MAX+PLUS II design-site users without a current maintenance agreement. The promotion continues through December 19, 1997. When you upgrade your software, you will receive the following features:

- Four quarterly upgrades
- A choice of VHDL or Verilog HDL synthesis
- Functional simulation (for PLS-BASE)
- Extended features and device support for all Altera devices (for PLS-MAGNUM)

ES Site License users—You will receive large discounts when you upgrade to either a PLS-BASE or PLS-MAGNUM design site with one year of software maintenance.

Existing MAX+PLUS II users without a current maintenance agreement—You will receive new software features when you purchase maintenance, depending on your existing design site. See Table 7.

For more information on the worldwide tools promotion, contact your local Altera distributor.

Table 7. Upgrade to MAX+PLUS II Version 8.1 to Receive These Features

If you have…	You can upgrade to…	To receive these standard features..
PLS-ES, PLS-ADV, or PLS-STD	PLS-BASE	■ VHDL or Verilog HDL synthesis ■ Functional simulation
PLS-ES, PLS-ADV, or PLS-STD (with one migration product)	PLS-BASE + your existing migration product	■ VHDL or Verilog HDL synthesis ■ Functional simulation
PLS-ES, PLS-ADV, or PLS-STD (with more than one migration product), *Note (1)*	PLS-MAGNUM	■ VHDL or Verilog HDL synthesis ■ Support for FLEX 6000, FLEX 10K, FLEX 8000, and MAX 9000 device families
PLS-FLEX8 (with no migration products)	PLS-BASE + PLSM-6K/8K	■ VHDL or Verilog HDL synthesis ■ Functional simulation
PLS-FLEX8 (with any migration product) *Note (1)*	PLS-MAGNUM	■ VHDL or Verilog HDL synthesis ■ Support for FLEX 10K and MAX 9000 device families
PLS-MAGNUM	PLS-MAGNUM	■ VHDL or Verilog HDL synthesis
PLS-HPS, *Note (1)*	PLS-MAGNUM	■ VHDL or Verilog HDL synthesis ■ Support for FLEX 6000, FLEX 10K, FLEX 8000, and MAX 9000 device families
PLS-QUARTET, *Note (1)*	PLS-MAGNUM	■ VHDL or Verilog HDL synthesis ■ Support for FLEX 10K and MAX 9000 device families
PLS-WS	PLS-WS	■ VHDL or Verilog HDL synthesis ■ Schematic editor ■ Timing and functional simulation ■ Static timing analysis

Note:
(1) If you have a current software maintenance agreement, you can upgrade to a PLS-MAGNUM system for free with MAX+PLUS II version 8.1.

Synthesis Support

As device density grows, designers are moving towards a high-level design methodology. To support this trend, Altera regularly enhances the MAX+PLUS□II software to provide optimal synthesis solutions for users at all experience levels. For example, to help new users transition to a high-level design flow, Altera is offering a *free* synthesis tool to all users with a current software maintenance agreement. For advanced users, Altera works with major synthesis tool vendors, such as Synopsys, Synplicity, and Exemplar Logic, to ensure that these sophisticated synthesis tools are optimized for Altera device architectures. Table 8 shows Altera's synthesis solution for different experience levels. Evaluation copies of the Synopsys, Synplicity, and Exemplar Logic tools can be downloaded from their respective web sites.

Table 8. Synthesis Solution

Design Expertise	Synthesis Solution	Tool
New synthesis user	HDL synthesis included in all purchased MAX+PLUS II design sites	–
Sophisticated synthesis user	Synopsys	Design Compiler FPGA Compiler FPGA Express
	Synplicity	Synplify
	Exemplar Logic	Galileo Galileo Extreme Leonardo

Entry-Level Software via the Web

With MAX+PLUS II version 8.1, Altera is making it easier for new designers to obtain entry-level software. PLS-WEB, Altera's free entry-level development software, can be downloaded from the Altera world-wide web site. In addition, the ES Site License software, an entry-level feature set of the MAX+PLUS□II software, can be enabled through an authorization code obtained from Altera's WWW site at **http://www.altera.com/authcode**. The PLS-WEB and ES Site License software include the following features:

- Schematic and text-based design entry
- Static timing analysis
- Graphical floorplan editing
- Compilation support for Classic, MAX 5000, MAX□7000, MAX 7000S, EPF8282A, EPF8452A, EPM9320, and EPF10K10 devices

Discontinued Support for Windows 3.1 & Windows for Workgroups

Compilation and synthesis processing requirements for large programmable logic devices (PLDs) continue to demand advanced computer system configurations. The MAX+PLUS II software has kept pace with this demand by providing the fastest compilation times in the industry.

Because 32-bit operating systems, such as Windows□NT, Windows 95, and UNIX, provide the most efficient performance for compilers, Altera is focusing on 32-bit operating systems for MAX+PLUS□II. 16-bit operating systems, such as Windows 3.1 and Windows for Workgroups, do not support 32-bit processing bandwidth or the enhanced Windows user interface. In addition, for 32-bit designs to run on 16-bit operating systems, the design must be processed by an inefficient abstraction layer, which degrades performance.

Therefore, Altera is discontinuing support for Windows 3.1 and Windows for Workgroups with MAX+PLUS□II version 8.3 in 1998. However, the Altera stand-alone programmer, PL-ASAP2, will continue to support Windows 3.1 and Windows for Workgroups through the remainder of 1998.

Technical ARTICLES

Configuring FLEX 10KA Devices

Because Altera's 3.3-V FLEX 10KA devices (including EPF10K50V and EPF10K130V devices) have 5.0-V-tolerant inputs, you can configure the devices using the 5.0-V BitBlaster™ or ByteBlaster™ download cables. These download cables can channel configuration data to FLEX 10KA devices located on system circuit boards that have both 3.3-V and 5.0-V power planes. The download cables use the 5.0-V power supply, while the FLEX 10KA devices use the 3.3-V power supply. By downloading design changes directly to a device, you can easily prototype the devices and accomplish multiple design iterations in quick succession.

This article discusses passive serial (PS) and IEEE Std. 1149.1 (JTAG) configuration of FLEX 10KA devices with the BitBlaster or ByteBlaster download cables, and discusses how a system circuit board should be wired to support these device configurations.

Wiring the Circuit Board

Before configuring a FLEX 10KA device with a BitBlaster or ByteBlaster download cable, the system circuit board should be properly wired to support a PS or JTAG configuration. The BitBlaster and ByteBlaster download cables have 5.0-V TTL input and output pins and drive out 5.0-V TTL compatible signals with a required high-level voltage (V_{OH}) of 2.4 V. FLEX 10KA devices have the MultiVolt™ I/O interface option, which allows I/O pins to accept 5.0-V inputs and drive output voltage levels compatible with a TTL V_{OH} of 2.4 V. Before configuring a FLEX 10KA device with a BitBlaster or ByteBlaster download cable, you should consider the following guidelines:

- Do not pull up FLEX 10KA device outputs to the 5.0-V power supply, and do not pull up BitBlaster or ByteBlaster outputs to the 3.3-V supply. Violating this rule provides a current path between the 5.0-V and 3.3-V power supplies. However, FLEX 10KA open-drain outputs can be pulled up to the 5.0-V or 3.3-V supply.
- You can connect the VCCINT and VCCIO pins of the FLEX 10KA device to the 3.3-V power plane. However, you should not connect the VCCIO pin to a 2.5-V power plane, because 2.5-V output pins cannot drive 5.0-V TTL logic levels.

PS Configuration

FLEX 10KA devices can be configured using a PS configuration scheme. You should use the guidelines above to determine which configuration pins should be pulled up to which power plane. For instance, DCLK, DATA0, and nCONFIG are configuration pins that are driven by BitBlaster or ByteBlaster output pins. Therefore, these pins should be pulled up to the 5.0-V power supply. Because CONF_DONE and nSTATUS pins are bidirectional, open-drain (i.e., they drive low or tri-state) pins driven by a FLEX 10KA device, you should also pull them up to the 5.0-V or 3.3-V supply. See Figure 1.

JTAG Configuration

FLEX 10KA devices can be configured using the industry-standard JTAG interface. You should use the guidelines above to determine which JTAG-dedicated configuration pins to pull up to which power plane. For instance, the ByteBlaster output pins TDI, TCK, and TMS should be pulled up to the 5.0-V power supply. The nCONFIG and TRST pins can be connected to the 5.0-V or 3.3-V supply. Because CONF_DONE and nSTATUS pins are bidirectional, open-drain pins driven by the FLEX 10K device, they should also be pulled up to the 5.0-V or 3.3-V supply. See Figure 2.

For more information on FLEX 10K devices, see the *FLEX 10K Embedded Programmable Logic Family Data Sheet*. For more information on the BitBlaster or ByteBlaster download cables, see the *BitBlaster Serial Download Cable Data Sheet* and the *ByteBlaster Parallel Port Download Cable Data Sheet*, respectively.

Figure 1. PS Configuration Using the ByteBlaster Download Cable

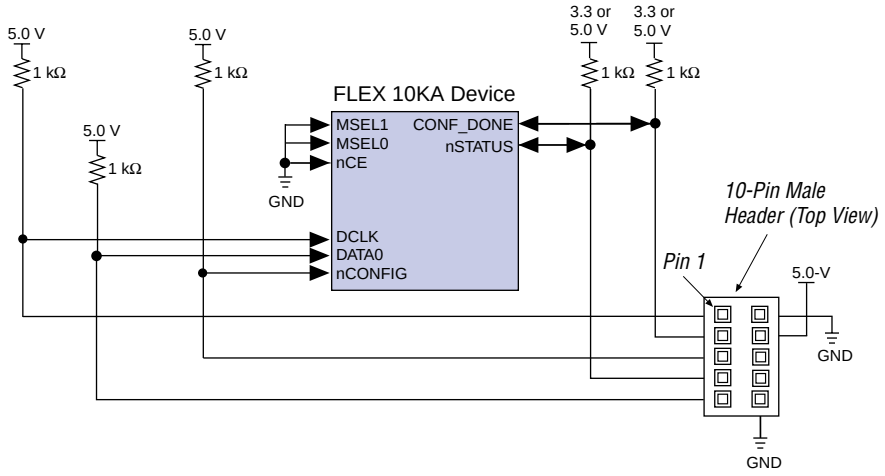
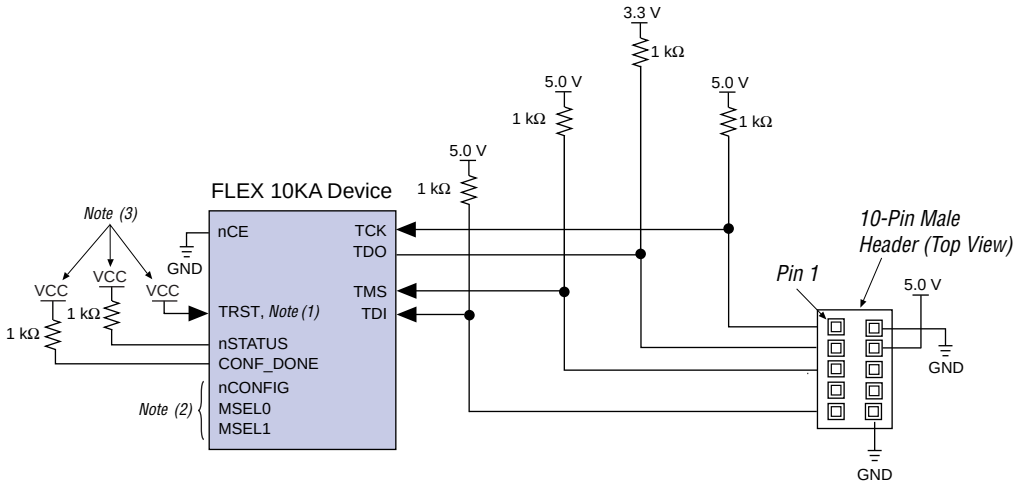


Figure 2. JTAG Configuration Using a ByteBlaster Download Cable



Notes:

(1) Because FLEX 10KA devices in 144-pin TQFP packages do not have a `TRST` pin, you can ignore this connection.

(2) The `nCONFIG`, `MSEL0`, and `MSEL1` pins must be connected to support a FLEX configuration scheme. If only JTAG configuration is used, connect `nCONFIG` to V_{CC}, and `MSEL0` and `MSEL1` to ground.

(3) `TRST` and `nCONFIG` are connected directly to either the 3.3-V or 5.0-V power plane. The `nSTATUS` and `CONF_DONE` pins are pulled up to either the 3.3-V or 5.0-V power plane.

Using the Jam Language to Configure FLEX 10K & FLEX 10KA Devices

Using in-circuit reconfigurability (ICR) via an embedded processor provides designers with greater flexibility during device configuration. The Jam™ programming and test language supports ICR via an embedded processor using the MAX+PLUS® II development system version 8.1 or higher. This article describes how to use the Jam language for ICR via an embedded processor to configure FLEX® 10K and FLEX 10KA devices (including EPF10K50V and EPF10K130V devices).

This article should be used with *AN 88 (Using the Jam Language for ISP via an Embedded Processor)* and the *Jam Programming & Test Language Specification*.

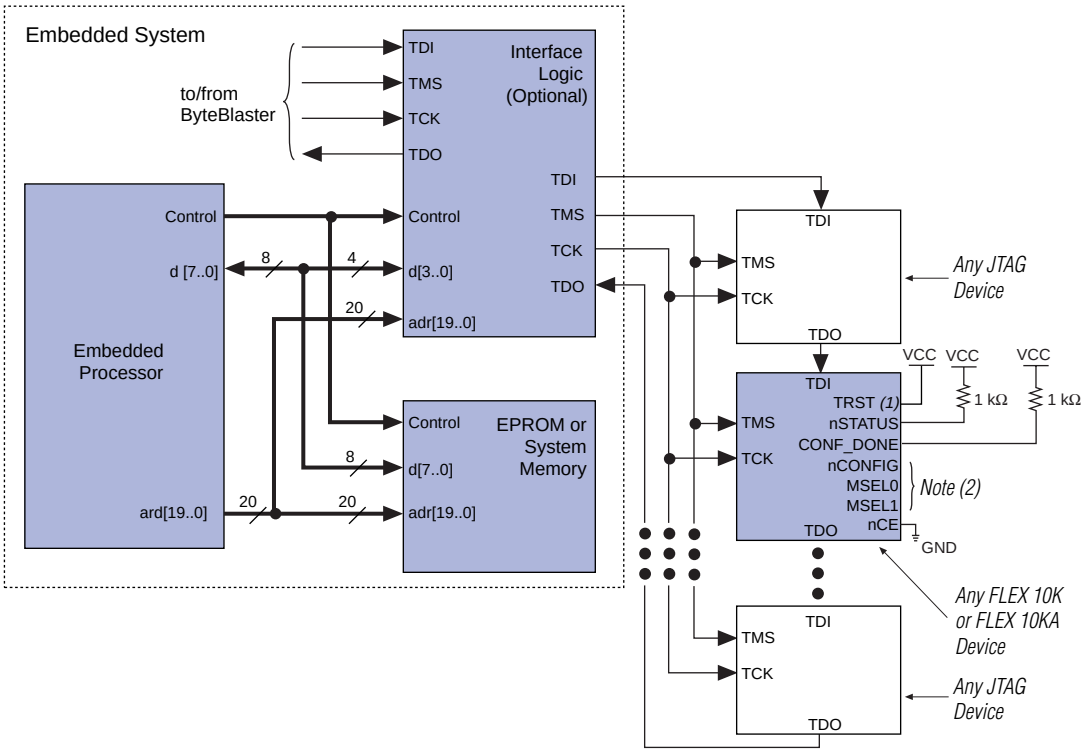
For more information about Altera's support for the Jam language, go to the Altera world-wide web site at **http://www.altera.com/jam-isp**.

The Embedded System Interface

Embedded systems typically consist of an embedded processor, EPROM or system memory, and optional interface logic. Configuration or programming data is stored in system memory (i.e., in EPROM or FLASH memory). For ICR, the embedded processor transfers data from the system memory to a programmable logic device (PLD). Figure 1 shows a block diagram of an embedded system for FLEX 10K or FLEX 10KA devices.

The embedded processor connects to the system memory and PLD that stores the optional interface logic. Although you can connect the JTAG chain directly to four embedded processor data pins, you

Figure 1. FLEX 10K & FLEX 10KA Embedded System Block Diagram



Notes:
(1) Because FLEX 10KA devices in 144-pin TQFP packages do not have a `TRST` pin, you can ignore this connection.
(2) `nCONFIG`, `MSEL0`, and `MSEL1` pins should be connected to support a FLEX configuration scheme. If only JTAG configuration is used, connect `nCONFIG` to V_{CC} and `MSEL0`, and `MSEL1` to ground.

should use the interface logic to save these four pins. Using the interface logic will treat the JTAG chain as an address location on the existing bus. To download JTAG chain configuration data via BitBlaster or ByteBlaster cables, install a 10-pin male header on your circuit board.

Using the Jam Language for Embedded Configuration & Programming

The Jam language is implemented in two steps. First, you must create a Jam File (**.jam**)—using the MAX+PLUS□II software—for configuring your FLEX□10K or FLEX 10KA device(s). You should store the Jam File in system memory. Second, run the Jam Player that resides in the embedded processor. The Jam Player parses the Jam File, interprets the Jam instructions, and reads and writes data to and from the JTAG chain.

Jam File

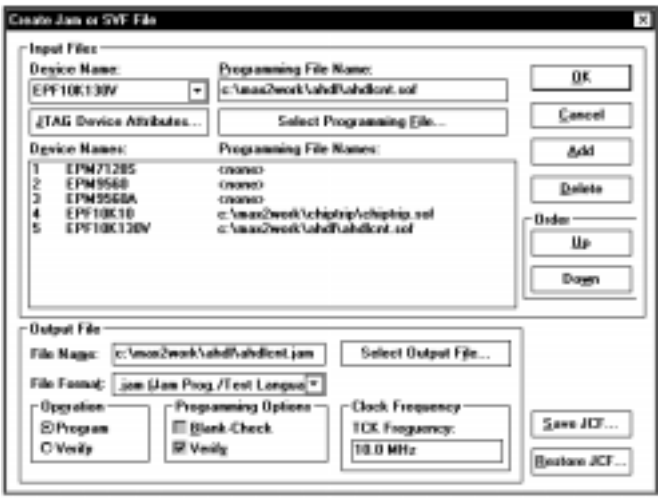
When you use the **Create Jam or SVF File** command (File menu), the MAX+PLUS II Programmer generates Jam Files from SRAM Object Files (**.sof**) or Programmer Object Files (**.pof**).

When creating Jam Files for a JTAG chain containing both FLEX and MAX® devices, you should create separate Jam Files to configure the FLEX and MAX devices independently. Unlike MAX 9000, MAX 7000S, and MAX 7000A devices, FLEX devices require configuration upon power-up. Your FLEX Jam File can bypass the MAX devices in the JTAG chain if you do not want to reprogram the MAX devices in-system. You can use the **Create Jam or SVF File** dialog box in the MAX+PLUS II software to create a Jam File for configuration or programming.

In the sample JTAG chain shown in Figure 2, the EPM7128S device is the first device and the EPF10K130V device is the last device in the JTAG chain. The EPM7128S, EPM9560, and EPM9560A devices are bypassed during configuration.

For complete instructions on how to create and edit Jam Files, search for "Creating Jam or SVF Files" in MAX+PLUS II Help.

Figure 2. Creating a Jam File for Configuration & Programming



Jam Player

The Jam Player has two parts: the main program and I/O functions. The main program performs all basic functions without requiring modification. You will need to modify the I/O functions, which are contained in the **jamstub.c** file, to cusomtize the Jam Player for your application. These I/O functions specify addresses to I/O pins, delay routines, operating system-specific functions, and routines for file I/O.

You can execute the Jam Player on PCs at a DOS or command prompt. The PC processor acts as the embedded processor and a ByteBlaster download cable is the interface to the JTAG chain. You can create the Jam Player using the following command:

```
jam [-h] [-v] [-p<Hexadecimal parallel port
    address>] [-m<Memory size in bytes>]
    -d<Initialization list> <Jam File>
```

Table 1. -d Variables to Configure FLEX 10K Devices

Variable Name	Value	Function
DO_CONFIGURE	0	Do not configure the device.
	1	Configure the device.
READ_UESCODE	0	Do not read the JTAG UES code.
	1	Read and report the UES code.

continued on page 21

Implementing a Counter Using a One-Hot Shift Register

Binary counters are used in many applications because they are fast and use minimal logic. However, for FLEX device applications that have many critical decoding paths, implementing a counter using a one-hot shift register is a better alternative.

Consider a 5-bit binary counter that requires 20 of its 32 values to be decoded. This function would require 5 logic elements (LEs) for the counter and a minimum of 22 additional LEs for the decode logic. The counter registers are likely to be placed in a single contiguous area and the counter's five outputs would need to be routed to wherever the decode logic resides (most likely spread out over the device).

In comparison, the same 32-bit counter implemented using a one-hot shift register uses only one logic cell for each counted value. This counter needs 32 logic cells for implementation and requires no additional decode logic. The one-hot counter has simple logic and can be spread over LEs throughout the device without impacting performance. As a result, one-hot counters are generally best used for highly-decoded counters.

Standard one-hot counters always require one register to be a `1`. Because all registers in Altera devices powerup to `0`, it may be difficult to correctly code a one-hot counter. To resolve this problem, you should place a `NOT` gate before and after the first register. The first bit will be the inverse of the standard one-hot counter. The following table compares the six count values of a standard one-hot counter with an Altera one-hot counter.

Table 1. Counter Values

Value	Standard One Hot Counter	Altera One-Hot Counter
0	100000	000000
1	010000	110000
2	001000	101000
3	000100	100100
4	000010	100010
5	000001	100001

Figure 1 shows a block diagram of an Altera one-hot counter that iterates through these six values.

For one-hot counters of varying lengths, you can use a function from the library of parameterized modules (LPM). Figure 2 shows the design from Figure 1 implemented in VHDL using the function `lpm_shiftreg`. Because `lpm_shiftreg` is a parameterizable function, it can be modified easily for large or small one-hot counters. Because `NOT` gates must be placed before and after the first register, they cannot be included in the LPM shift register. Figure 3 shows a simulation of the VHDL example.

Figure 1. One-Hot Counter Block Diagram

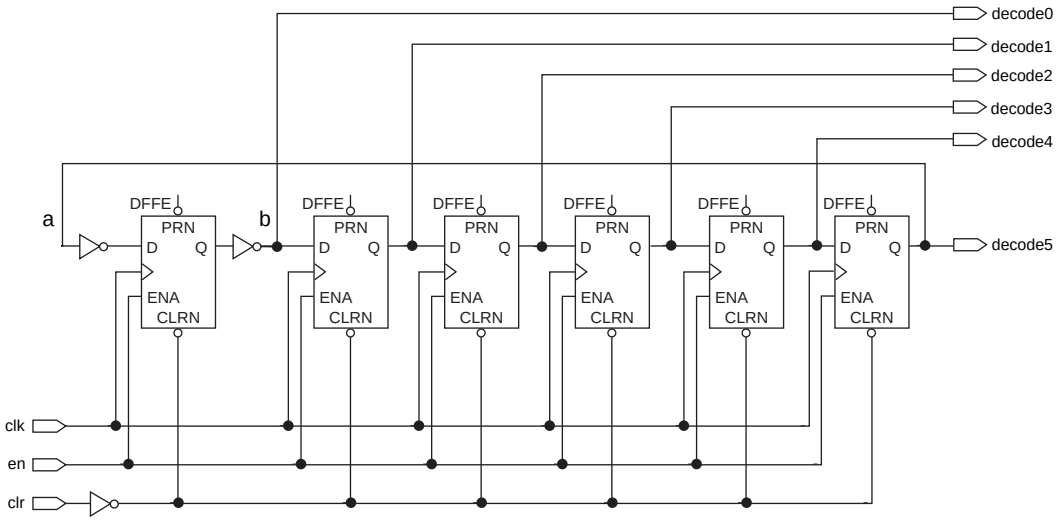
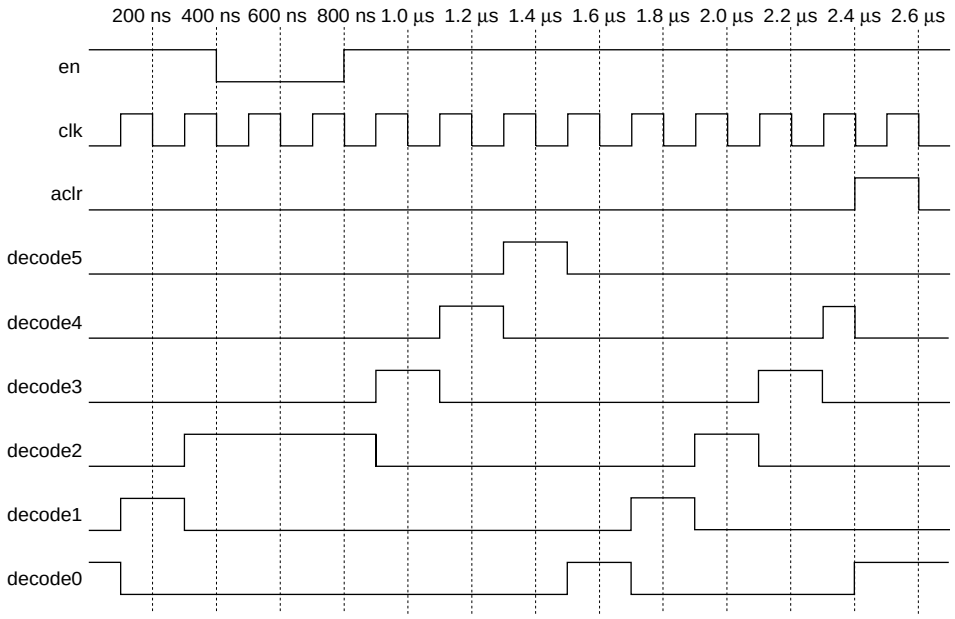


Figure 2. VHDL Counter using lpm_shiftreg

```
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.ALL;

LIBRARY lpm;
USE lpm.lpm_components.ALL;

ENTITY onehot IS

PORT(
    clk         : IN STD_LOGIC;
    en, aclr    : IN STD_LOGIC;
    decode      : OUT STD_LOGIC_VECTOR(5 DOWNTO 0));
END onehot;

ARCHITECTURE altera OF onehot IS

SIGNAL a,b          : STD_LOGIC;
SIGNAL first_reg    : STD_LOGIC;
SIGNAL sr_out       : STD_LOGIC_VECTOR(4 DOWNTO 0);

BEGIN

    U1: lpm_shiftreg
        GENERIC MAP (LPM_WIDTH => 5)
        PORT MAP(
            clock => clk,
            enable => en,
            shiftin => b,
            aclr => aclr,
            q => sr_out,
            shiftout => a);

    PROCESS (clk, aclr)
    BEGIN
        IF aclr = '1' THEN
            first_reg <= '0';
        ELSIF clk'EVENT and clk = '1' THEN
            IF en = '1' THEN
            first_reg <= not a;
            END IF;
        END IF;
    END PROCESS;

    b <= not first_reg;
    decode(5 DOWNTO 1) <= sr_out(4 DOWNTO 0);
    decode(0) <= b;

END altera;
```

Figure 3. One-Hot Counter Simulation of the VHDL Example



Questions & ANSWERS

Q *Can I configure a 3.3-V FLEX® 10K device with the ByteBlaster™ or BitBlaster™ download cable?*

A You can use the 5.0-V BitBlaster or ByteBlaster download cable to configure 3.3-V FLEX 10K devices. The download cables should be connected to a 5.0-V power supply and the 3.3-V FLEX 10K devices should be connected to a 3.3-V power supply. Ensure that the FLEX 10K device is connected to the 3.3-V power supply before attempting configuration with the BitBlaster or ByteBlaster download cables. See "Configuring FLEX 10KA Devices" on page 10 for complete details.

Q *How large are Jam™ Files generated for Altera devices?*

A As one of the PLD vendors supporting Jam, Altera has developed support for ISP via Jam Files for MAX 7000S and MAX 9000 devices. The data provided in the following table correlates to designs that utilize over 90% of MAX 7000S devices, and over 80% of MAX 9000 devices.

Device	Typical Jam File Size (Kbytes)
EPM7064S	22
EPM7128S	29
EPM7192S	35
EPM7256S	41
EPM9320	50
EPM9400	62
EPM9480	67
EPM9560	72

Note:
(1) File size data calculated using MAX+PLUS II version 8.1.

Jam File sizes will vary depending on how much of the device is used. The Jam format compresses the program and verify data. The compression algorithm looks for repetitive strings of data. For a device with high utilization, the probability of the algorithm finding repetitive data will decrease.

For Jam files that target multiple Altera devices, you can estimate the file size using the following equations. These equations assume that all devices in the chain are the same. For example, a Jam File targeting three EPM7064S devices would be 22 + (2 × 1) = 24 Kbytes.

EPM7064S: 22 Kbytes + [(*N* – 1) × 1] Kbytes
EPM7128S: 29 Kbytes + [(*N* – 1) × 5] Kbytes
EPM7192S: 35 Kbytes + [(*N* – 1) × 5] Kbytes
EPM7256S: 41 Kbytes + [(*N* – 1) × 7] Kbytes
EPM9320: 50 Kbytes + [(*N* – 1) × 13] Kbytes
EPM9400: 62 Kbytes + [(*N* – 1) × 16] Kbytes
EPM9480: 67 Kbytes + [(*N* – 1) × 20] Kbytes
EPM9560: 72 Kbytes + [(*N* – 1) × 25] Kbytes

Where *N* is the total number of devices programmed.

Q *How do I activate a MegaCore™ function?*

A The following steps explain how to activate any MegaCore function with the MAX+PLUS II software.

1. In MAX+PLUS II version 8.1, choose **Authorization Code** (Options menu).
2. Choose the **MegaCore/AMPP Licenses** button.
3. In the **MegaCore/AMPP Licenses** dialog box, enter the appropriate megafunction ID code in the *Megafunction ID* box. (See the following table).

Ordering Code	Megafunction ID Code
PLSM-MICROLIB	6AF8-1
PLSM-6402	6AF8-8
PLSM-6850	6AF8-9
PLSM-16450	6AF8-A
PLSM-8237	6AF8-B
PLSM-8251	6AF8-C
PLSM-8255	6AF8-D
PLSM-8259	6AF8-E
PLSM-FFT	6AF8-2
PLSM-CSC	6AF8-3
PLSM-PCI/A	6AF8-4
PLSM-CRC	6AF8-5

4. In the *License Authorization Code* box, enter the license authorization code. You will be given an authorization code when you license the function from Altera. If you are test-driving the megafunction using the OpenCore™ feature, you can skip this step. However, you cannot generate programming or configuration files for a design using the megafunction unless you have licensed the function.
5. Choose **Add** and then choose **OK** to enable the function.

Q *How do I generate a pin-out and a Programmer Object File (**.pof**) for the EPF6016BC256 device in MAX+PLUS II version 8.1?*

A EPF6016BC256 devices will be fully supported in MAX+PLUS II software version 8.2 and higher. To

generate a pin-out and POF for EPF6016BC256 devices in MAX+PLUS II software version 8.1, you must enter a password by performing the following steps:

1. In the MAX+PLUS II Programmer, choose **Device** (Assign menu).
2. In the **Device** dialog box, choose *FLEX 6000* in the *Device Family* drop-down list box. Choose **OK**.
3. Choose **Select Device** (Options menu). In the **Select Device** dialog box, select *FLEX 6000* in the *Device Family* drop-down list box.
4. Select *EPF6016BC256* from the *Available Devices* drop-down list box.
5. Choose the **Enable** button and enter *4SBNYL* in the *Password* box.
6. Choose **Add**. You should now see the password in the *Existing Passwords* box.
7. Choose **OK**.

The MAX+PLUS II software can now support EPF6016BC256 devices.

Q *In what order do I program two EPC1 devices for FLEX 10K device configuration?*

A Two EPC1 devices are required to configure EPF10K100 and larger devices. The MAX+PLUS II software automatically generates two Programmer Object Files (**.pof**) for the two devices.

The first EPC1 device in the chain (i.e., the EPC1 device with the `CONF_DONE` pin connected to the `nCS` pin) is programmed with <*project name*>**.pof**. The second EPC1 device in the chain is programmed with <*project name*>**_1.pof** .

For further information, go to the *Configuration EPROMs for FLEX Devices Data Sheet*.

Q *Which programming adapter should I use to program the EPC1441 Configuration EPROM?*

A When programming EPC1441 Configuration EPROMs, you should use PLMJ1213 adapters for 8-pin DIP and 20-pin PLCC packages and PLMT1064 adapters for 32-pin TQFP packages.

Q *When will Data I/O provide programming support for EPC1441 devices?*

A The algorithm for programming EPC1441 devices is currently under final testing. The algorithm will be included in Data I/O software version 5.7, which is scheduled for release in March 1998. The programming algorithm will be posted on Data I/O's Keep Current - Bulletin Board Service (BBS) in early December 1997. Go to the Data I/O web site for more information (**http://www.dataio.com**).

Q *Why aren't my pin assignments visible in my Graphic Design File (**.gdf**)?*

A Pin assignments may be invisible for the following two reasons:

1. The **Show Pins/Locations/Chips** command in the MAX+PLUS II Graphic Editor is turned off. To turn it on, choose **Show Pins/Locations/Chips** (Options menu).

2. You have made pin assignments to a bus pin. Assign each bus signal to individual pins and label them accordingly. Ensure that all brackets ([]) are removed from pin names, because "`a[0]`" specifies a bus pin with a width of one.

Q *Why do I get hold time violations when simulating my FLEX 10K design in MAX+PLUS II version 8.1?*

A When using EPF10K70, EPF10K100, or EPF10K100A devices in any speed grade, you may see hold time violations when simulating with the MAX+PLUS II Simulator or a third-party simulator. If the hold time violation is caused by one register driving the clock enable of another register in the same logic array block (LAB), a spurious hold time violation has occurred. This violation can be ignored. All other hold time violations are still valid.

A software update (MAX+PLUS II version 8.14) that corrects this hold time violation is available on the Altera web site at **http://www.altera.com**.

Q *In MAX+PLUS II version 8.1, why don't the input pin numbers show up in my EDIF Output File (**.edo**)?*

A When the MAX+PLUS II Compiler creates an EDIF Output File, the file should contain all of the pin numbers for input, output, and bidirectional pins. However, when the **Optimize Timing SNF** command (Processing menu) is turned on, the EDIF Output File will not list input pin numbers. Turn off the **Optimize Timing SNF** command to include the input pin numbers in your EDIF Output File.

The **Optimize Timing SNF** command is used to make simulations run faster. Turning off this option does not change the functionality or performance of the design; on UNIX workstations it makes **.edo** files smaller.

Altera VIEWPOINT

Achieving Performance Goals

Jack Ogawa
Senior Product Planning Manager

Historically, designers developing digital logic had two clear options. For designs requiring high density, high performance, and the lowest unit cost, ASICs were the best choice. For designs that required flexibility and fast time-to-market, programmable logic devices (PLDs) offered a solid solution.

The decision process in today's digital logic marketplace is quite different. With the advent of 250,000-gate devices and a cost-per-gate price that rivals gate arrays, today's PLDs can provide a solution that once was exclusively the domain of ASICs. Like ASICS, designing for these high-density PLDs can be a challenge. Higher levels of abstraction are required for design entry efficiency, making the optimization process much more challenging.

Once a design has been functionally verified as part of a top-down design flow, two common tactics can be used to minimize the logic levels in the design's critical path. The simplest method is to use the best synthesis tool for the task. The second technique is to modify the design files to optimize area and performance.

Choosing a Development Tool

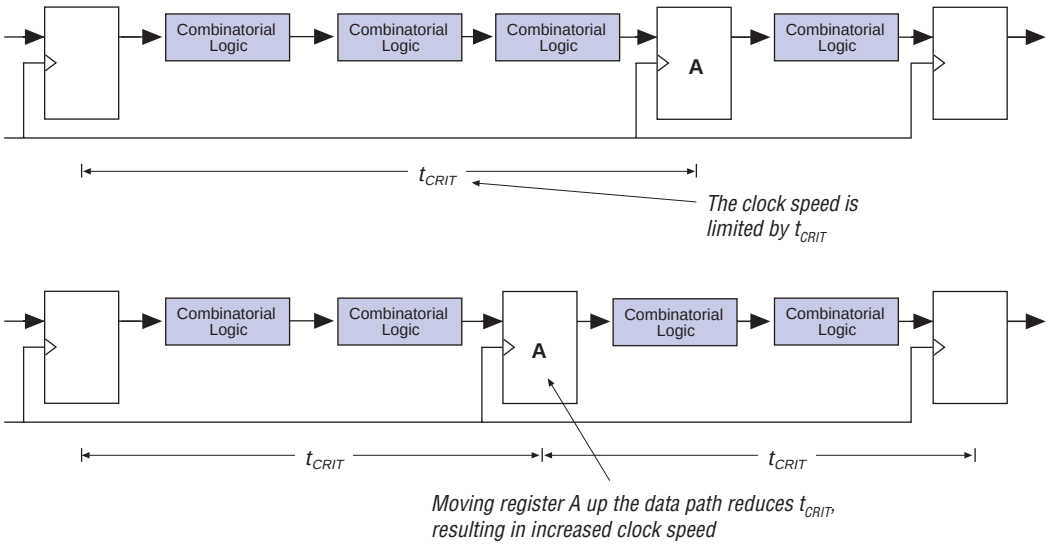
Today's synthesis tools have powerful features that allow you to make area/speed tradeoffs to achieve your design goals. Depending on the tool, you may be able to indicate a preference for area or speed, specify timing constraints for a given clock or data path, or balance register-to-register delays to maximize clock performance (see Figure 1).

When choosing a development tool, you should weigh the features and capabilities of the tool versus the price. Many EDA tool vendors provide different tiers of tools for different prices. The design flow and goals should drive designers to select the right development tool.

Using Effective Design Techniques

By far, the most effective strategy for improving a design's area utilization and performance is working with the design's source files. For performance, the goal is to reduce the logic levels in the critical path by tailoring the design source code. You can either design at the gate level to remove any inefficiency caused by the

Figure 1. Balancing Register-to-Register Delays



synthesis tool, or change the implementation of the critical function. At first, this procedure seems like a painful task. However, designing hierarchically can minimize the pain and allow "what-if" scenarios to be implemented easily .

The Verilog HDL examples in Figure 2 show two implementations of an up/down counter function. The use of an optimized counter yields dramatically improved results, both in resource area and performance. In a design hierarchy, these implementations can be easily switched to determine which one yields the best results.

When implementing the examples from Figure 2 in an EPF6016 device, using the pre-developed, optimized counter more than doubled clock performance and reduced logic element (LE) usage to a fourth of the original size (see Table 1).

Table 1. Performance & Optimization Improvement

Item	Inferred	Instantiated
LEs	65	16
Clock speed	58 MHz	135 MHz

Another option is to use functions from the library of parameterized modules (LPM). With LPM functions, you can implement commonly used functions that can be controlled via user-specified parameters. Figure 3 shows the symbol for the LPM function `lpm_mult`. The parameters of the `lpm_mult` function can be changed easily to achieve the desired design results, for example, using pipeline stages to improve clock speeds.

Conclusion

The density and performance of PLDs has grown to accommodate system-level functional blocks. This growth also brings a renewed awareness of the area and performance requirements of these blocks, much like in traditional ASIC design flows. If a you can plan for the optimization process by selecting the right development tools and using a hierarchical design structure, the full potential of a PLD solution can be realized, offering the best of both worlds.

Figure 2. Inferred & Instantiated Verilog HDL Counter

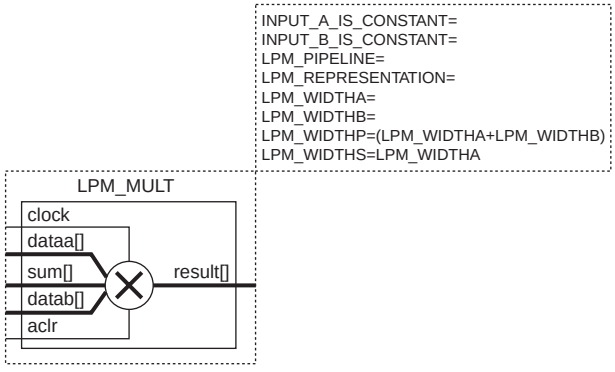
Inferred

```
module cntr(din, load, up_dn, reset, clk_en. clk, count);
    input [15:0] din;
    input load;
    input up_dn;
    input reset;
    input clk_en;
    input clk;
    output [15:0] count;

    reg [15:0] creg;
    wire [15:0] nreg;
    assign count = creg;

    assign nreg = (~reset) ? 0
        : (load) ? din
        : (up_dn) ? creg + 1
        : creg - 1;
    always @(posedge clk) begin
        creg = nreg;
    end
endmodule
```

Instantiated

```
module updn_ctr_rpl
    (data, up_dn, load, clk, reset, count);
    parameter width = 16;
    input [width-1 : 0] data;
    input up_dn, load, clk, reset;
    output [width-1 : 0] count;

/* Synopsys dc_script_begin set_implementation rpl U0 */

    // instantiate DW03_UPDN_CTR DW03_updn_ctr # (width)
    U0(data, up_dn, load, 1, clk, reset, count);
endmodule
```

Figure 3. lpm_mult Multiplier



Customer Application

Success with the Jam Language

*If Cisco Systems designers had used a Serial Vector Format (**.svf**) File, they would have had to devote a large amount of memory and management software to properly store the **.svf** file.*

"Before we used the Jam language, we had problems updating boards in the field," explained Joerg Czech, a member of the hardware and systems development team at Heitec Datentechnik Gmbh. "With the Jam language, we can save a lot of time and money. We send updated Jam Files and the Jam Player to our customers via e-mail, and they update the boards themselves."

Since its introduction on July 8, 1997, there have been several successful applications of the Jam™ programming and test language. The Jam language, a software-level standard for in-system programmability (ISP), has successfully addressed issues concerning ease-of-use, fast programming times, and small file sizes. With these capabilities, the Jam language has provided solutions to problems that have plagued in-system programming.

Cisco Systems

Cisco Systems, Inc. is currently using the Jam language to program an EPM7192S device on a main controller board. The processor on this board not only programs the EPM7192S device in-system, but also identifies and programs an EPM7128S device that is attached to the JTAG chain in an expandable daughter card. Using Serial Vector Format Files (**.svf**) would have required a large amount of memory and management software to properly store the SVF Files. Instead, Cisco used Jam Files (**.jam**), which are much smaller. Because Jam Files are small enough to fit in one memory location, extra memory is available for other tasks or files, giving designers more flexibility.

Lauterbach Datentnik GMBH

Heitec Datentechnik Gmbh was the first company to implement a working Jam system. One of their recent projects was to develop testing hardware for GSM base stations. A key goal of the project was high system performance (i.e., 90-MHz system clock). Additionally, designers wanted to be able to upgrade the system in the field.

The Heitec Datentechnik design team chose a combination of EPM7128S and EPM9560 devices for their project. Because the devices can be programmed in-system via the JTAG ports, the team was able to implement a prototype system quickly.

It took only three hours to port the Jam Player to their 32-bit Siemens processor. The Jam language allowed designers to program the EPM7128S devices using a ByteBlaster™ cable, a PC, and the Jam Player, which reduced their overall cost and board size. Designers were also able to replace a 500-Kbyte SVF File with a 25-Kbyte Jam File.

Before the Jam language was created, Heitec Datentechnik had to recall boards from the field to make system changes. Using Jam Files allows the boards to be upgraded in the field, saving time and costs. "Before we used the Jam language, we had problems updating boards in the field," explained Joerg Czech, a member of the hardware and systems development team. "With the Jam language, we can save a lot of time and money. We send updated Jam Files and the Jam Player to our customers via e-mail, and they update the boards themselves."

Conclusion

For customers such as Cisco Systems and Heitec Datentechnik, the Jam language has successfully reduced file sizes and programming times. Because the Jam language is both silicon-vendor-independent and platform-independent, it can address other concerns such as proprietary file formats, vendor-specific programming algorithms, and the ability to work with existing and future devices manufactured on different processes. As an open standard, the Jam language can benefit all silicon vendors, manufacturers, and programmers. For more information about the Jam programming and test language, go to Altera's world-wide web site at **http://www.altera.com**.

Using the Jam Language to Configure FLEX 10K & FLEX□10KA Devices
continued from page 13

Brackets ([]) indicate optional parameters. Table 1 shows the variables required after the `-d` option to configure a FLEX 10K device with the Jam language.

For example, if you type the following text at a DOS or command prompt, the Jam Player will configure a device, limit the memory to 1 Mbyte, and write/read data through the parallel port at base address 0x378 for the Jam File **chiptrip.jam**:

```
jam -p378 -m1000000 -dDO_CONFIGURE=1
   chiptrip.jam
```

JTAG Timing Parameters & Waveforms

You should use JTAG timing parameters and waveforms to ensure proper Jam Player operation for any system. Table 2 shows the TAP state machine timing specifications. These timing parameters are consistent with those given in the IEEE Std. 1149.1 specification.

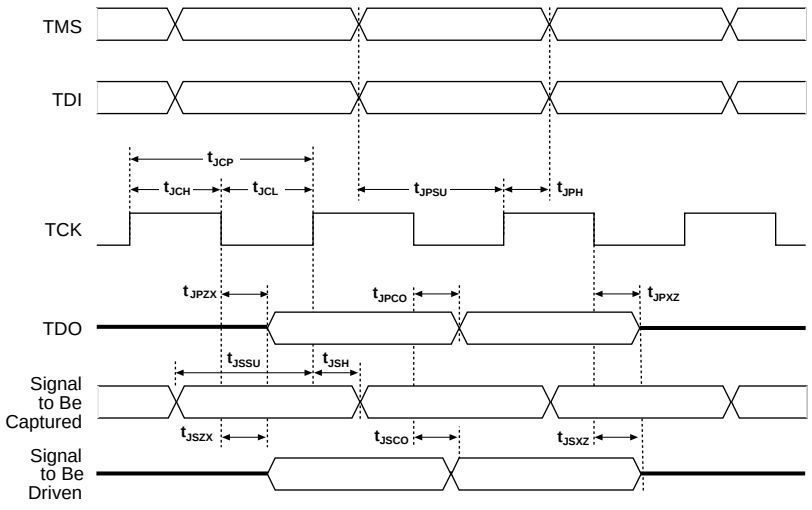
Figure 3 illustrates waveforms that correspond to each timing parameter.

Table 2. JTAG Timing Parameters for FLEX 10K & FLEX 10KA Devices

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, *Note (1)*	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		25	ns
t_{JSZX}	Update register high-impedance to valid output		25	ns
t_{JSXZ}	Update register valid output to high-impedance		25	ns

Note:
(1) The hold time is dependent on the falling edge of the JTAG configuration clock pin (TCK).

Figure 3. JTAG Waveforms



Tips to Boost Registered Performance

Coercing your design to meet the desired speed (e.g., by assigning and re-assigning cliques, adjusting synthesis settings, etc.) can be a complicated task. Although each project should be analyzed on a case-by-case basis, reducing the number of critical paths that do not meet the specified speed can often give your design the extra help it needs. This article describes two tips to boost registered performance:

- Identify logic paths that cross routing rows and group related logic in the same row
- Reduce the longest delay path

Identify Logic Paths that Cross Routing Rows & Group Related Logic in the Same Row

Logic paths that cross rows incur an additional column delay. Grouping time-critical paths in the same row can reduce the overall delay. To identify paths that cross rows, perform the following steps in the MAX+PLUS® II Timing Analyzer Registered Performance display:

1. Perform a timing analysis of your project.
2. Choose **List Paths** and review the number of paths that do not meet the specified performance frequency. To set the performance frequency, select **Time Restrictions** (Options menu) in the Timing Analyzer and adjust the settings in the *Registered Performance Options* box as needed for your design.
3. Begin with the slowest path and work your way down the list. Sometimes, making adjustments to the slowest path alone will boost the project to the desired speed. After selecting the slowest path in the MAX+PLUS II Meggage Processor, perform the following steps:
   a. Turn on *Locate in Floorplan Editor* in the Message Processor window.
   b. Choose **Locate All**.
   c. In the Floorplan Editor, select the logic cells that cross routing rows and assign them to a clique. See Figure 1. When assigning a clique, the Compiler usually groups paths together within the same row. However, you should make sure that the paths are actually in the same row. To find an assigned clique, choose **Find Clique in Floorplan** (Utilities menu). Search for "Clique Assignments Section" in MAX+PLUS II Help for more information.
   d. Recompile and re-analyze your project. If the project still does not meet the desired performance, select the next longest delay path and repeat steps a through c.

Instead of assigning cliques, you can assign all the cells in a timing path to the same row. In the Floorplan Editor, select the timing path, and choose **Pin/Location/Chip** (Assign menu) to assign the functions to a row. This process keeps the related logic in the same row without locking down the exact logic cells.

Reduce the Longest Delay Path

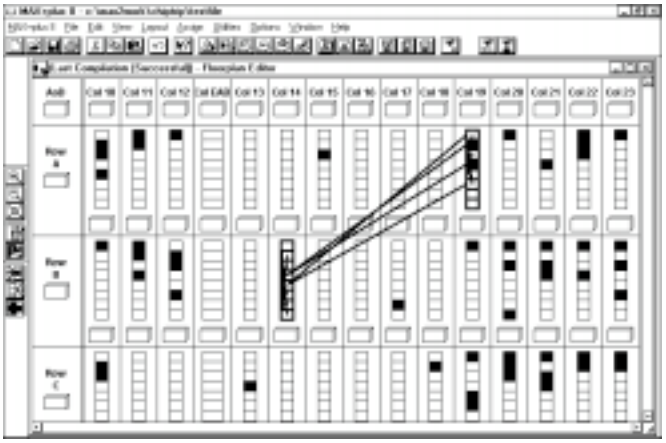
This section describes ways to reduce the longest delay path:

- Register balancing
- Pipelining your project
- Using the look-ahead decode method

Register Balancing

A project will sometimes have a longer delay between two registers than desired, but a shorter delay between two other adjacent registers in the same path. In this situation, shifting the delay from one register to another can balance the registered performance, thereby boosting the project's speed. In Figure 2, moving the register shortens the longest path by 5 ns (i.e., from 15 ns to 10 ns).

Figure 1. Using the Floorplan Editor



Pipelining your Project

Pipelining your project is similar to register balancing because it balances the delay from one register to another. In pipelining, however, you add registers to the project. Each pipeline stage (i.e., each added register) has a one-clock latency delay. Because the FLEX® architecture includes a register with each look-up table (LUT), pipelining your project generally does not require additional device resources. In Figure 3, adding a pipeline stage reduces the longest path by 15 ns (i.e., from 25 ns to 10 ns).

Using the Look-Ahead Decode Method

Two levels of logic are associated with decoding the data of a 16-bit counter. If the decoded data fans out to combinatorial logic, the delay path becomes longer. To reduce the delay path, decode the data one clock cycle earlier and register the output to provide a shorter delay path. See Figure 4.

Figure 2. Shifting the Delay from One Register to Another

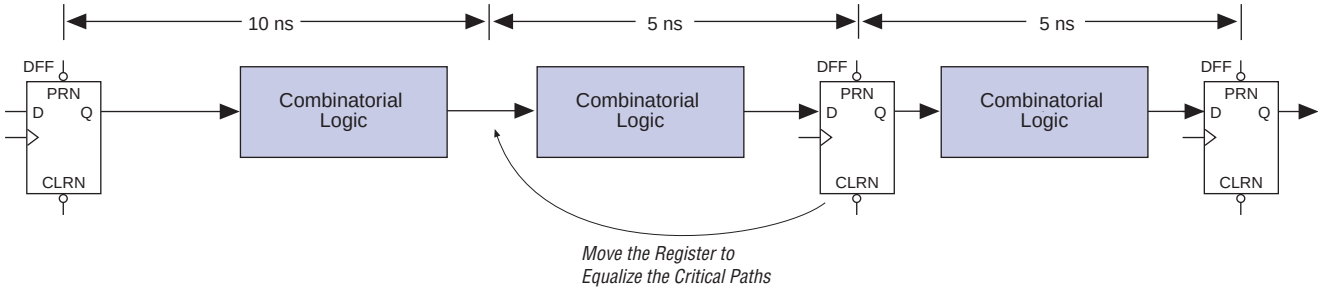


Figure 3. Pipelining your Design

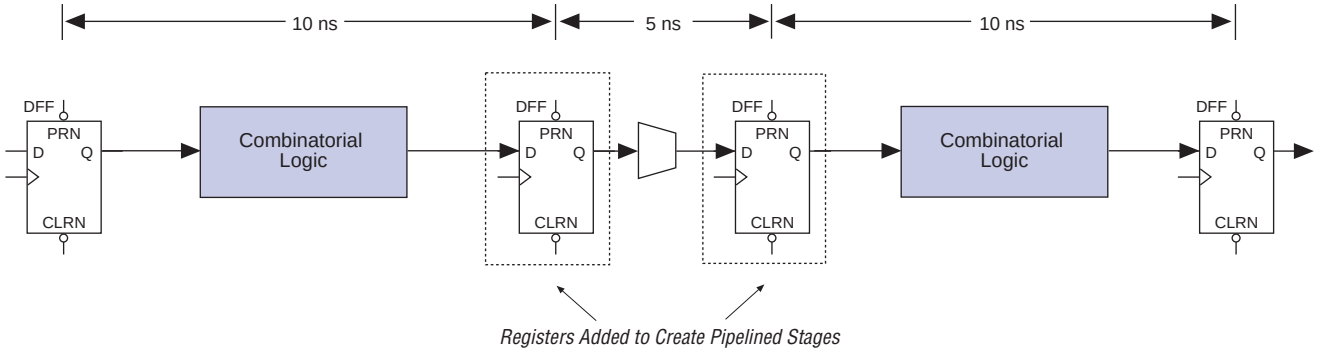
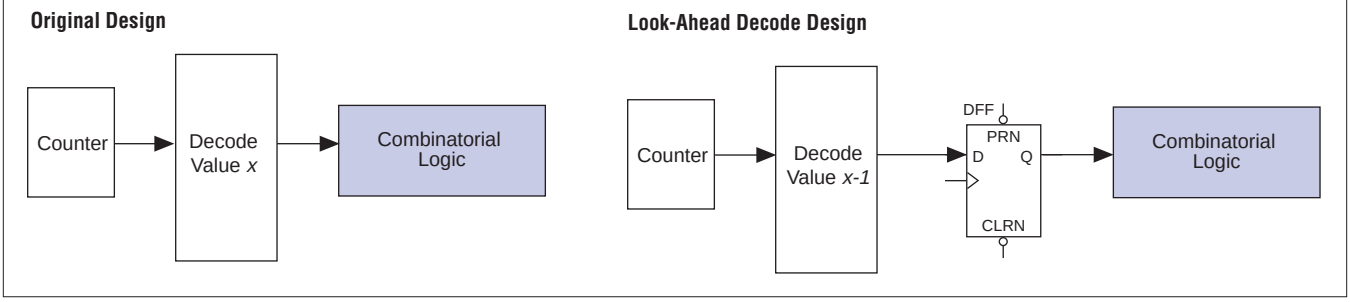


Figure 4. Using the Look-Ahead Decode Method



Altera NEWS

Altera Forms Design Consultants Alliance

As programmable logic device (PLD) densities continue to increase, Altera recognizes that designers require innovative tools and design expertise to boost their productivity and enable them to take advantage of the growing PLD capacity. Today's increasing PLD densities encourage more complex designs, but complex designs usually require more expertise and design time. Thus, the demand for both intellectual property and design consultation has increased.

To address the increasing demand for intellectual property, Altera created the Altera Megafunction Partners Program (AMPP) and Altera MegaCore™ functions. AMPP℠ and MegaCore functions provide a growing library of megafunctions targeted for specific Altera® device architectures.

To meet the design consultation demand, Altera recently created another innovative alliance. On September 15, 1997, Altera announced the formation of the Altera Consultants Alliance Program (ACAP). The ACAP℠ partnership is a worldwide alliance between Altera and specially trained design consultants that broadens support for designers using Altera PLDs. ACAP consultants provide design services that target Altera devices, giving designers a new resource to reduce design cycle times and time-to-market.

The ACAP goal is to attract experienced design consultants familiar with Altera PLDs, certify them, and make their skills available to designers worldwide. Before Altera certifies and recommends ACAP consultants, the consultants receive advanced training in Altera architectures and software, and are equipped with state-of-the-art design tools. After training, Altera lists ACAP consultants by geography, area of expertise, and third-party development tool knowledge.

Certified ACAP consultants are currently working in North America and Europe. You can review consultants' qualifications and areas of expertise by referring to the Altera world-wide web site at **http://www.altera.com**. Current ACAP consultants are shown below:

- Advanced Logical Design, Inc.
- ASIC Designs, Inc.
- Design Analysis Associates, Inc.
- Eberwein & Associates, Inc.
- I/F/I
- Locke's Digital Development, Ltd.
- Northwest Logic Design
- Software and Systems Engineering, Inc.
- Systems Design Group

Contact ACAP consultants directly for more information. As independent contractors, consultants are solely responsible for contractual agreements. Altera does not participate in negotiating consulting terms or fees, or warrant consultants' work.

To inquire or comment about the ACAP program, please send e-mail to **acap-info@altera.com**. To find out more about becoming an ACAP member, please send e-mail to **acap@altera.com**.

Altera at ICSPAT & DSP World Expo

Altera participated in the International Conference on Signal Processing and Technology (ICSPAT) and DSP World Expo September 14 through 17 in San Diego, California. Altera engineers presented 2 papers, *Pipelined Adaptive Filters in Altera PLDs* and *DSP Processor Core for FLEX□10K*and co-presented *PLD-Based FFTs* and *A PLD-Based Solution for Cable Modem* with AMPP partner Integrated Silicon Systems. All four papers were presented in the "FPGAs in DSP" session.

Over 500 customers stopped by the Altera DSP World Expo booth to see the FLEX DSP presentation, *A Total Solution for Your Most Complex DSP Applications*. For more information on Altera's DSP solutions, go to Altera's web site at **http://www.altera.com**.

Altera Target Applications

Target Applications provide solutions to application-specific needs. Target Applications leverage MegaCore™ functions and Altera Megafunction Partners Program (AMPP℠) functions to provide integrated solutions that deliver significant time-to-market benefits. The complete Target Applications solution includes both the megafunctions and the documentation that is critical to get these megafunctions and reference designs working. Altera Target Applications focuses on markets such as digital signal processing (DSP), peripheral component interconnect (PCI), and wireless and broadband communications. For more information, go to the Altera world-wide web site at **http://www.altera.com**. Altera Target Applications include:

- FLEX DSP
  - DSP Building Blocks
  - DSP Imaging Functions
  - DSP Wireless and Broadband Communications
- Bus Interfaces
  - PCI
  - Universal Serial Bus (USB) Communications
- Communications
  - Data Communications and Telecommunications
  - Asynchronous Transfer Mode (ATM)

New Altera Publications

New Altera publications are available from Altera Literature Services. Individual documents are available on Altera's world-wide web site at **http://www.altera.com**. Document part numbers are shown in italic type.

- Configuration EPROMs for FLEX Devices Data Sheet (*A-DS-EPROM-07*)
- FLEX 10K Embedded Programmable Logic Family Data Sheet Supplement (*A-DSS-F10K-2.5*)
- EPF10K100A Embedded Programmable Logic Family Data Sheet Supplement (*A-DSS-F10K-2.6*)
- EPF10K50V Embedded Programmable Logic Device Data Sheet Supplement (*A-DSS-F10K-2.7*)
- EPF10K100A Embedded Programmable Logic Device Data Sheet Supplement (*A-DSS-F10K-2.8*)
- FLEX 6000 Programmable Logic Device Family Data Sheet Supplement (*A-DSS-F6KPLD-2.1*)
- SB 22: CAN Bus Megafunction (*A-SB-022-01*)
- TB 29: Internal Tri-State Emulation (*M-TB-029-01*)
- TB 30: Authorization Codes Now via the WWW (*M-TB-030-01*)
- TB 31: The Advantages of FLEX 10K Devices vs. Lucent Orca Devices (*M-TB-031-01*)
- TB 32: ISP Programming Methods & Ordering Codes (*M-TB-032-01*)

Discontinued Devices

In recent months, Altera has announced that various products will be discontinued (see the table below). Altera distributes advisories (ADVs) and product discontinuance notices (PDNs) that provide information on discontinued devices. To obtain a copy of a specific ADV or PDN, contact your local Altera sales representative. Selected ADVs, PDNs, and a complete listing of discontinued devices are also available on Altera's world-wide web site at **http://www.altera.com**. Rochester Electronics, an after-market supplier, offers support for many discontinued Altera products. Contact Rochester Electronics at (508) 462-9332 for more information. You can also go to their web site at **http://www.rocelec.com**.

Discontinued Device Ordering Codes

Device Family	Device	Last Order Date	Last Shipment Date	Reference
FLASHlogic	EPX880 and EPX8160 (all packages, temperature grades, and speed grades)	6/30/97	6/30/98	PDN 9625
MAX 5000	EPM5032SC-15	6/30/97	12/31/97	PDN 9624
MAX 7000	EPM7256SRC208-12	12/31/97	3/31/98	PDN 9713

Altera at PLD World '97

The 4th annual Altera PLD World show, hosted by Altera Japan, was held on October 24 in Tokyo, Japan. This one-day event featured the latest Altera products and strategies, raised awareness of programmable logic devices (PLDs), tools, and design methodologies, and positioned Altera as the gate-array alternative. The presentations given by Altera employees included:

- *Reshaping the ASIC Landscape* (keynote address) by Rodney Smith
- *Altera: The Proven PCI Platform* by Masaru Hamada
- *Altera Data Communication & High-End Video Applications* by Leo Wong
- *Gate Array Design Flow* by Leo Wong
- *Challenges & Solutions for the New Millenium* by Craig Lytle

Additional presentations were given by each of the 18 participating third-party companies, including many of Altera's ACCESS℠ and AMPP℠ partners. Companies participating in PLD World this year included Synopsys, Viewlogic, and Data I/O. The photographs below illustrate some of Altera's booths at the show.

In Every ISSUE

Third-Party Programming Support

Data I/O and BP Microsystems provide programming hardware support for selected Altera devices. Algorithms are supplied via Data I/O's Keep Current Express-Bulletin Board Service (KCE-BBS) and BP Microsystems' BBS. Programming support for Configuration EPROM, MAX® 9000, and MAX 7000 devices is shown in the table below. All information is subject to change.

Third-Party Programming Hardware Support

Device	Data I/O *(1)*	BP Microsystems *(2)*
EPC1064	✓	✓
EPC1213	✓	✓
EPC1	✓	✓
EPC1441	*Note (3)*	✓
EPM7032	✓	✓
EPM7064	✓	✓
EPM7064S	✓	✓
EPM7096	✓	✓
EPM7128E	✓	✓
EPM7128S	✓	✓
EPM7160E	✓	✓
EPM7192E	✓	✓
EPM7192S	✓	✓
EPM7256E	✓	✓
EPM7256S	✓	✓
EPM9320	✓	✓
EPM9400	✓	✓
EPM9480	✓	✓
EPM9560	✓	✓

Notes:
(1) These devices are supported by Data I/O 3900 version 5.6, and UniSite version 5.6 programmers.
(2) These devices are supported by BP Microsystems programmers version 3.28.
(3) Data I/O plans to support EPC1441 devices. Contact Data I/O directly for more information.

Current Software Versions

The latest versions of Altera software products are shown below:

- MAX+PLUS II version 8.1 (PC, Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 platforms)

Altera Programming Hardware Support

The following tables contain the latest programming hardware information for Altera devices. For correct programming, use the software version shown in "Current Software Versions"on page 26. See Table 1.

Table 1. Altera Programming Adapters (Part 1 of 2) *Note (1)*

Device	Package	Adapter
EPC1064 *(2)*, EPC1064V *(2)*, EPC1441 *(3)*	DIP, J-lead TQFP	PLMJ1213 PLMT1064
EPC1 *(3)*, EPC1213, *(2)*	DIP J-lead	PLMJ1213 PLMJ1213
EPM9320A	J-lead (84-pin) RQFP (208-pin) RQFP (240-pin)	PLMJ9320-84 PLMR9000-208NC, *(4)* PLMR9000-240NC, *(4)*
EPM9320	PGA J-lead (84-pin) RQFP (208-pin)	PLMG9000-280 PLMJ9320-84 PLMR9000-208
EPM9400	J-lead (84-pin) RQFP (208-pin) RQFP (240-pin)	PLMJ9400-84 PLMR9000-208 PLMR9000-240
EPM9480	RQFP (208-pin) RQFP (240-pin)	PLMR9000-208 PLMR9000-240
EPM9560A	RQFP (208-pin) RQFP (240-pin)	PLMR9000-208NC, *(4)* PLMR9000-240NC, *(4)*
EPM9560	PGA (280-pin) RQFP (208-pin) RQFP (240-pin) RQFP (304-pin)	PLMG9000-280 PLMR9000-208 PLMR9000-240 PLMR9000-304
EPM7032S	J-lead (44-pin) TQFP (44-pin)	PLMJ7000-44 PLMT7000-44
EPM7032, EPM7032V	J-lead (44-pin) PQFP (44-pin) TQFP (44-pin)	PLMJ7000-44 PLMQ7000-44 PLMT7000-44
EPM7064S	J-lead (44-pin) TQFP (44-pin) TQFP (100-pin)	PLMJ7000-44 PLMT7000-44 PLMT70000-100NC, *(4)*
EPM7064	J-lead (44-pin) TQFP (44-pin) J-lead (68-pin) J-lead (84-pin) PQFP (100-pin)	PLMJ7000-44 PLMT7000-44 PLMJ7000-68 PLMJ7000-84 PLMQ7000-100
EPM7096	J-lead (68-pin) J-lead (84-pin) PQFP (100-pin)	PLMJ7000-68 PLMJ7000-84 PLMQ7000-100
EPM7128S	J-lead (84-pin) PQFP (100-pin) TQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100NC, *(4)* PLMT7000-100NC, *(4)* PLMQ7128/160-160NC

Table 1. Altera Programming Adapters (Part 2 of 2) *Note (1)*

Device	Package	Adapter
EPM7128, EPM7128E	J-lead (84-pin) PQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100 PLMQ7128/7160-160
EPM7160S	J-lead PQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100NC, *(4)* PLMQ7128/7160-160NC, *(4)*
EPM7160E	J-lead PQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100 PLMQ7128/7160-160
EPM7192S, *(4)*	PQFP (160-pin)	PLMQ7192/256-160NC
EPM7192E	PGA (160-pin) PQFP (160-pin)	PLMG7192-160 PLMQ7192/7256-160
EPM7256S, *(4)*	RQFP (208-pin)	PLMQ7256-208NC
EPM7256E	PQFP (160-pin) PGA (192-pin) RQFP (208-pin)	PLMQ7192/7256-160 PLMG7256-192 PLMR7256-208

Notes:
(1) Refer to the Altera ***1996 Data Book*** for device adapter information on MAX 5000 and Classic devices. Altera offers an adapter exchange program for 0.8-micron EPM5032, EPM5064, and EPM5130 programming adapters. See "MAX 5000 & Classic Product Transitions" on page 7 of this newsletter for more information.
(2) FLEX 8000 Configuration EPROM.
(3) FLEX 10K, FLEX 8000, or FLEX 6000 Configuration EPROM.
(4) These devices are not shipped in carriers.

Table 2 provides programming and configuration compatibility information for the BitBlaster™ serial and ByteBlaster™ parallel port download cables.

Table 2. BitBlaster & ByteBlaster Cable Compatibility

Device	Package	Hardware
FLEX 10K FLEX 10KA FLEX 10KB	All packages	PL-BITBLASTER PL-BYTEBLASTER
FLEX 8000	All packages	PL-BITBLASTER PL-BYTEBLASTER
FLEX 6000	All packages	PL-BITBLASTER PL-BYTEBLASTER
MAX 9000	All packages	PL-BITBLASTER PL-BYTEBLASTER
MAX 7000S MAX 7000A	All packages	PL-BITBLASTER PL-BYTEBLASTER

Altera Device Selection Guide

All current information for the Altera FLEX 10K, FLEX 8000, FLEX 6000, MAX 9000, and MAX 7000 devices is listed here. Information on other Altera products is located in the Altera ***1996 Data Book***. Contact Altera or your local sales office for current product availability.

FLEX 10K Devices *Note (1)*

DEVICE	GATES	PIN/PACKAGE OPTIONS	I/O PINS *Note (2)*	TEMP.	SPEED GRADE	FLIP-FLOPS	LOGIC ELEMENTS	RAM BITS
EPF10K10	10,000	84-Pin PLCC, 144-Pin TQFP, 208-Pin QFP	59, 107, 134	C	-3, -4	720	576	6,144
EPF10K10	10,000	144-Pin TQFP, 208-Pin QFP	107, 134	I	-4	720	576	6,144
EPF10K10A	10,000	144-Pin TQFP, 208-Pin QFP	107, 134	C	-1, -2, -3	720	576	6,144
EPF10K10A	10,000	144-Pin TQFP, 208-Pin QFP	107, 134	I	-3	720	576	6,144
EPF10K20	20,000	144-Pin TQFP, 208-Pin QFP, 240-Pin QFP	107, 147, 189	C	-3, -4	1,344	1,152	12,288
EPF10K20	20,000	144-Pin TQFP, 208-Pin QFP, 240-Pin QFP	107, 147, 189	I	-4	1,344	1,152	12,288
EPF10K30	30,000	208-Pin QFP, 240-Pin QFP, 356-Pin BGA	147, 189, 246	C	-3, -4	1,968	1,728	12,288
EPF10K30	30,000	208-Pin QFP, 240-Pin QFP	147, 189	I	-4	1,968	1,728	12,288
EPF10K30A	30,000	144-Pin TQFP, 208-Pin QFP, 240-Pin QFP, 256-Pin BGA	107, 147, 189, 189	C	-1, -2, -3	1,968	1,728	12,288
EPF10K30A	30,000	144-Pin TQFP, 208-Pin QFP, 240-Pin QFP, 256-Pin BGA	107, 147, 189, 189	I	-3	1,968	1,728	12,288
EPF10K30B	30,000	144-Pin TQFP, 208-Pin QFP, 240-Pin QFP, 256-Pin BGA	107, 147, 189, 189	C	-1, -2, -3	1,968	1,728	12,288
EPF10K30B	30,000	144-Pin TQFP, 208-Pin QFP, 240-Pin QFP, 256-Pin BGA	107, 147, 189, 189	I	-3	1,968	1,728	12,288
EPF10K40	40,000	208-Pin QFP, 240-Pin QFP	147, 189	C	-3, -4	2,576	2,304	16,384
EPF10K50	50,000	240-Pin QFP, 356-Pin BGA, 403-Pin PGA	189, 274, 310	C	-3, -4	3,184	2,880	20,480
EPF10K50	50,000	240-Pin QFP	189	I	-4	3,184	2,880	20,480
EPF10K50V	50,000	240-Pin QFP, 356-Pin BGA	189, 274	C	-1, -2, -3, -4	3,184	2,880	20,480
EPF10K50V	50,000	240-Pin QFP, 356-Pin BGA	189, 274	I	-4	3,184	2,880	20,480
EPF10K50B	50,000	208-Pin QFP, 240-Pin QFP, 256-Pin BGA, 356-Pin BGA	147, 189, 189, 274	C	-1, -2, -3	3,184	2,880	20,480
EPF10K50B	50,000	208-Pin QFP, 240-Pin QFP, 256-Pin BGA, 356-Pin BGA	147, 189, 189, 274	I	-3	3,184	2,880	20,480
EPF10K70	70,000	240-Pin QFP, 503-Pin PGA	189, 358	C	-2, -3, -4	4,096	3,744	18,432
EPF10K100	100,000	503-Pin PGA	406	C	-3, -4	5,392	4,992	24,576
EPF10K100A	100,000	240-Pin QFP, 356-Pin BGA, 600-Pin BGA	189, 274, 406	C	-1, -2, -3	5,392	4,992	24,576
EPF10K100A	100,000	240-Pin QFP, 356-Pin BGA, 600-Pin BGA	189, 274, 406	I	-3	5,392	4,992	24,576
EPF10K100B	100,000	208-Pin QFP, 240-Pin QFP, 356-Pin BGA, 600-Pin BGA	147, 189, 274, 406	C	-1, -2, -3	5,392	4,992	24,576
EPF10K100B	100,000	208-Pin QFP, 240-Pin QFP, 356-Pin BGA, 600-Pin BGA	147, 189, 274, 406	I	-3	5,392	4,992	24,576
EPF10K130V	130,000	599-Pin PGA, 600-Pin BGA	470, 470	C	-2, -3, -4	7,120	6,656	32,768
EPF10K130B	130,000	240-Pin QFP, 356-Pin BGA, 599-Pin PGA, 600-Pin BGA	189, 274, 470, 470	C	-1, -2, -3	7,120	6,656	32,768
EPF10K130B	130,000	240-Pin QFP, 356-Pin BGA, 600-Pin BGA	189, 274, 470	I	-3	7,120	6,656	32,768
EPF10K180B	180,000	240-Pin QFP, 356-Pin BGA, 600-Pin BGA	189, 274, 470	C	-1, -2, -3	10,534	9,728	32,768
EPF10K250A	250,000	599-Pin PGA, 600-Pin BGA	470, 470	C	-1, -2, -3	12,624	12,160	40,960
EPF10K250A	250,000	600-Pin BGA	470	I	-3	12,624	12,160	40,960
EPF10K250B	250,000	356-Pin BGA, 599-Pin PGA, 600-Pin BGA	274, 470, 470	C	-1, -2, -3	12,624	12,160	40,960
EPF10K250B	250,000	356-Pin BGA, 600-Pin BGA	274, 470	I	-3	12,624	12,160	40,960

Notes:
(1) Select FLEX 10KA and FLEX 10KB devices will be available in 1998.
(2) Six I/O pins are dedicated inputs.

FLEX 6000 Devices

DEVICE	GATES	PIN/PACKAGE OPTIONS	I/O PINS *(1)*	TEMP.	SPEED GRADE	FLIPFLOPS	LOGIC ELEMENTS
EPF6016	16,000	100-Pin TQFP, 144-Pin TQFP, 208-Pin QFP, 240-Pin PQFP, 256-Pin BGA	81, 117, 171, 199, 204	C	-2	1,320	1,320
		100-Pin TQFP, 144-Pin TQFP, 208-Pin QFP, 240-Pin PQFP, 256-Pin BGA	81, 117, 171, 199, 204	C, I *(3)*	-3	1,320	1,320
EPF6016A *(2)*	16,000	100-Pin TQFP, 144-Pin TQFP, 208-Pin QFP, 240-Pin PQFP, 256-Pin BGA	81, 117, 171, 199, 204	C	-2	1,320	1,320
		100-Pin TQFP, 144-Pin TQFP, 208-Pin QFP, 240-Pin PQFP, 256-Pin BGA	81, 117, 171, 199, 204	C, I *(3)*	-3	1,320	1,320
EPF6024A *(2)*	24,000	208-Pin QFP, 240-Pin PQFP, 256-Pin BGA	171, 199, 215	C	-2	1,960	1,960
		208-Pin QFP, 240-Pin PQFP, 256-Pin BGA	171, 199, 215	C, I *(3)*	-3	1,960	1,960

Notes:
(1) Four I/O pins are dedicated inputs.
(2) An "A" indicates a 3.3-V voltage supply.
(3) The faster commercial temperature speed grade devices are de-rated to operate over the industrial temperature range.

FLEX 8000 Devices

DEVICE	GATES	PIN/PACKAGE OPTIONS	I/O PINS (1)	TEMP.	SPEED GRADE	FLIP-FLOPS	LOGIC ELEMENTS
EPF8282A	2,500	84-Pin PLCC, 100-Pin TQFP	68, 78	C	A-2	282	208
		84-Pin PLCC, 100-Pin TQFP	68, 78	C, I	A-3		
		84-Pin PLCC, 100-Pin TQFP	68, 78	C, I	A-4		
EPF8282AV (2)	2,500	100-Pin TQFP	78	C	A-4	282	208
EPF8452A	4,000	160-Pin PQFP	120	C	A-2	452	336
		84-Pin PLCC, 100-Pin TQFP, 160-Pin PGA/PQFP	68, 120	C, I	A-3		
		84-Pin PLCC, 100-Pin TQFP, 160-Pin PGA/PQFP	68, 120	C, I	A-4		
EPF8636A	6,000	208-Pin PQFP	136	C	A-2	636	504
		84-Pin PLCC, 160-Pin PQFP, 192-Pin PGA, 208-Pin PQFP	68, 118, 136	C	A-3		
		84-Pin PLCC, 160-Pin PQFP, 192-Pin PGA, 208-Pin PQFP	68, 118, 136	C, I	A-4		
EPF8820A	8,000	144-Pin TQFP, 160-Pin PQFP, 192-Pin PGA, 208-Pin PQFP	112, 120, 152	C	A-2	820	672
		144-Pin TQFP, 160-Pin PQFP, 192-Pin PGA, 208-Pin PQFP, 225-Pin BGA	112, 120, 152	C	A-3		
		144-Pin TQFP, 160-Pin PQFP, 192-Pin PGA, 208-Pin PQFP, 225-Pin BGA	112, 120, 152	C, I	A-4		
EPF81188A	12,000	208-Pin PQFP, 232-Pin PGA, 240-Pin PQFP	148, 184	C	A-2	1,188	1,008
		208-Pin PQFP, 232-Pin PGA, 240-Pin PQFP	148, 184	C, I	A-3		
		208-Pin PQFP, 232-Pin PGA, 240-Pin PQFP	148, 184	C, I	A-4		
EPF81500A	16,000	240-Pin PQFP, 280-Pin PGA, 304-Pin RQFP	181, 208	C	A-2	1,500	1,296
		240-Pin PQFP, 280-Pin PGA, 304-Pin RQFP	181, 208	C, I	A-3		
		240-Pin PQFP, 280-Pin PGA, 304-Pin RQFP	181, 208	C	A-4		

Notes:
(1) Four I/O pins are dedicated inputs.
(2) A "V" indicates a 3.3-V voltage supply.

MAX 9000 Devices

DEVICE	MACRO-CELLS	PIN/PACKAGE OPTIONS	I/O PINS (1)	TEMP.	SPEED GRADE
EPM9320	320	84-Pin PLCC, 208-Pin RQFP, 280-Pin PGA, 356-Pin BGA	60, 132, 168	C	-15
	320	84-Pin PLCC, 208-Pin RQFP, 280-Pin PGA, 356-Pin BGA	60, 132, 168	C, I	-20
EPM9320A	320	84-Pin PLCC, 208-Pin RQFP, 356-Pin BGA	60, 132, 168	C	-10
	320	84-Pin PLCC, 208-Pin RQFP, 356-Pin BGA	60, 132, 168	C, I	-15
EPM9400	400	84-Pin PLCC, 208-Pin RQFP, 240-Pin RQFP	59, 139, 159	C	-15, -20
EPM9480	480	208-Pin RQFP, 240-Pin RQFP	146, 175	C	-15, -20
EPM9480A	480	208-Pin RQFP, 240-Pin RQFP	146, 175	C	-10, -12
	480	208-Pin RQFP, 240-Pin RQFP	146, 175	C, I	-15
EPM9560	560	208-Pin RQFP, 240-Pin RQFP, 280-Pin PGA, 304-Pin RQFP, 356-Pin BGA	153, 191, 216	C	-15
	560	208-Pin RQFP, 240-Pin RQFP, 280-Pin PGA, 304-Pin RQFP, 356-Pin BGA	153, 191, 216	C,I	-20
EPM9560A	560	208-Pin RQFP, 240-Pin RQFP, 356-Pin BGA	153, 191, 216	C	-10, -12
	560	208-Pin RQFP, 240-Pin RQFP, 356-Pin BGA	153, 191, 216	C,I	-15

Note:
(1) Four I/O pins are dedicated inputs.

MAX 7000 Devices (Part 1 of 2)

DEVICE	MACRO-CELLS	PIN/PACKAGE OPTIONS	I/O PINS	TEMP.	SPEED GRADE	t_{PD} (ns)	f_{CNT} (MHz)
EPM7032, EPM7032S	32	44-Pin PLCC/TQFP	36	C	-6	6	150
EPM7032, EPM7032S	32	44-Pin PLCC/TQFP	36	C	-7	7.5	125
EPM7032, EPM7032S	32	44-Pin PLCC/TQFP	36	C, I (1)	-10	10	100
EPM7032	32	44-Pin PLCC/TQFP	36	C, I	-12 (2)	12	90.9
EPM7032	32	44-Pin PLCC/TQFP	36	C, I	-15 (2)	15	76.9
EPM7032V (2)	32	44-Pin PLCC/TQFP	36	C	-12	12	90.9
EPM7032V (2)	32	44-Pin PLCC/TQFP	36	C	-15	15	76.9
EPM7032V (2)	32	44-Pin PLCC/TQFP	36	C, I	-20	20	62.5
EPM7064S	64	44-Pin PLCC/TQFP	36	C	-5	5	178.6
EPM7064, EPM7064S	64	44-Pin PLCC/TQFP, 68-Pin PLCC (2), 84-Pin PLCC, 100-Pin PQFP (2)/TQFP (1)	36, 52, 68	C	-6	6	150
EPM7064, EPM7064S	64	44-Pin PLCC/TQFP, 68-Pin PLCC (2), 84-Pin PLCC, 100-Pin PQFP (2)/TQFP (1)	36, 52, 68	C	-7	7.5	125
EPM7064, EPM7064S	64	44-Pin PLCC/TQFP, 68-Pin PLCC (2), 84-Pin PLCC, 100-Pin PQFP (2)/TQFP (1)	36, 52, 68	C, I (1)	-10	10	100
EPM7064	64	44-Pin PLCC/TQFP, 68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	36, 52, 68	C	-12 (2)	12	90.9
EPM7064	64	44-Pin PLCC/TQFP, 68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	36, 52, 68	C, I	-15 (2)	15	76.9

continued on page 30

Altera Device Selection Guide, continued from page 29

MAX 7000 Devices (Part 2 of 2)

DEVICE	MACRO-CELLS	PIN/PACKAGE OPTIONS	I/O PINS	TEMP.	SPEED GRADE	t_{PD} (ns)	f_{CNT} (MHz)
EPM7096	96	68-Pin PLCC *(1)*, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C	-7	7.5	125
EPM7096	96	68-Pin PLCC *(1)*, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C, I	-10	10	100
EPM7096	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C	-12 *(1)*	12	90.9
EPM7096	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C, I	-15 *(1)*	15	76.9
EPM7128S	128	84-Pin PLCC, 100-Pin PQFP/TQFP, 160-Pin PQFP	68, 84, 100	C	-6	6	150
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP/TQFP *(2)*, 160-Pin PQFP	68, 84, 100	C	-7	7.5	125
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP/TQFP *(2)*, 160-Pin PQFP	68, 84, 100	C, I *(2)*	-10(P)	10	100
EPM7128E	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C	-12 *(1)*	12	90.9
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP/TQFP *(2)*, 160-Pin PQFP	68, 84, 100	C, I	-15	15	76.9
EPM7128E	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C, I	-20 *(1)*	20	62.5
EPM7160S	160	84-Pin PLCC, 100-Pin PQFP/TQFP, 160-Pin PQFP	64, 84, 104	C	-7	7.5	125
EPM7160E, EPM7160S	160	84-Pin PLCC, 100-Pin PQFP/TQFP *(2)*, 160-Pin PQFP	64, 84, 104	C, I *(2)*	-10(P)	10	100
EPM7160E	160	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C	-12 *(1)*	12	90.9
EPM7160E, EPM7160S	160	84-Pin PLCC, 100-Pin PQFP/TQFP *(2)*, 160-Pin PQFP	64, 84, 104	C, I	-15	15	76.9
EPM7160E	160	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C, I	-20 *(1)*	20	62.5
EPM7192S	192	160-Pin PQFP	124	C	-7	7.5	125
EPM7192S	192	160-Pin PQFP	124	C	-10	10	100
EPM7192E	192	160-Pin PQFP/PGA	124	C	-12(P)	12	90.9
EPM7192E, EPM7192S	192	160-Pin PQFP/PGA *(1)*	124	C, I *(1)*	-15	15	76.9
EPM7192E	192	160-Pin PQFP/PGA	124	C, I	-20 *(1)*	20	62.5
EPM7256S	256	208-Pin RQFP, 208-Pin PQFP	132, 164	C	-7	7.5	125
EPM7256S	256	208-Pin RQFP, 208-Pin PQFP	132, 164	C	-10	10	100
EPM7256E, EPM7256S	256	160-Pin PQFP *(1)*, 192-Pin PGA *(1)*, 208-Pin RQFP, 208-Pin PQFP *(2)*	132, 164	C	-12(P)	12	90.9
EPM7256E, EPM7256S	256	160-Pin PQFP *(1)*, 192-Pin PGA *(1)*, 208-Pin RQFP, 208-Pin PQFP *(2)*	132, 164	C, I *(1)*	-15	15	76.9
EPM7256E	256	160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP, 208-Pin PQFP *(2)*	132, 164	C, I	-20 *(1)*	20	62.5

Notes:
(1) Not available in MAX 7000S devices.
(2) Available in MAX 7000S devices only.

How to Access Altera

Getting information and services from Altera is now easier than ever. The table below lists some of the ways you can reach Altera:

Information Type	Access	U.S. & Canada	All Other Locations
Literature *(1)*	Altera Literature Services	(888) 3-ALTERA lit_req@altera.com	(408) 544-7144 *(2)* lit_req@altera.com
	World-Wide Web	http://www.altera.com	http://www.altera.com
Non-Technical Customer Service	Telephone Hotline	(800) SOS-EPLD	(408) 544-7000
	Fax	(408) 544-6403	(408) 544-6403
Technical Support	Telephone Hotline (8 a.m. to 5 p.m. Pacific Time)	(800) 800-EPLD (408) 544-7000	(408) 544-7000 *(2)*
	Fax	(408) 544-6401	(408) 544-6401 *(2)*
	Electronic Mail	sos@altera.com	sos@altera.com
	FTP Site	ftp.altera.com	ftp.altera.com
General Product Information	Telephone	(408) 544-7104	(408) 544-7104 *(2)*
	World-Wide Web	http://www.altera.com	http://www.altera.com

Notes:
(1) For MAX+PLUS II software manuals, contact Altera Customer Service or your local distributor.
(2) You can also contact your local Altera sales office or sales representative. See the Altera web site for a listing.

Fax Response Form

☐ **I would like a subscription to *News & Views*.**

☐ **I would like to have my design featured in *News & Views*.**

☐ **Please correct my address.**

Your Name: _____

Organization: _____

Street Address: _____

City, State, ZIP: _____

Phone: _____

E-Mail: _____

Tell Us What You Think

Please take a moment to help us improve *News & Views* by rating the usefulness of the following sections. Your answers will help shape the content of future issues.

		Not Useful				Very Useful
1.	Devices & Tools	1	2	3	4	5
2.	New Altera Publications	1	2	3	4	5
3.	Questions & Answers	1	2	3	4	5
4.	Technical "How To" Articles	1	2	3	4	5
5.	Information on Altera's	1	2	3	4	5
	EDA Partners & Interface Support	1	2	3	4	5
6.	Customer Applications	1	2	3	4	5
7.	Software Utilities & Current Software Versions	1	2	3	4	5
8.	Altera News	1	2	3	4	5

Please write your comments about *News & Views* in the space below (use additional pages if necessary). Which subjects are not getting enough coverage? What questions do you still have? What new features would you like to see?

Please fax or mail a copy of this page to:

Altera Literature Distribution Services
Altera Corporation
101 Innovation Drive
San Jose, CA 95134
Fax: (408) 544-7809
E-mail: n_v@altera.com