The Altera Advantage News & Views

Newsletter for Altera Customers ◆ Second Quarter ◆ May 1996

Altera Ships 100,000-Gate PLD

Altera is now shipping the EPF10K100 device, which is not only the largest member of the FLEX 10K family, but also the largest device in the programmable logic industry. FLEX 10K devices contain both a logic array and an embedded array that can be used for RAM, ROM, or complex logic functions. With the added capability of the embedded array, the EPF10K100 offers 100,000 gates—a breakthrough for programmable logic.

The EPF10K100 is more than twice as large as any other programmable logic device (PLD) shipping today. Figure 1 compares the speed and density of the FLEX 10K family to the AT&T ORCA and Xilinx XC4000 families. The EPF10K100 contains approximately 10 million transistors; in contrast, the new Pentium Pro (P7) microprocessor from Intel contains 6.5 million transistors.

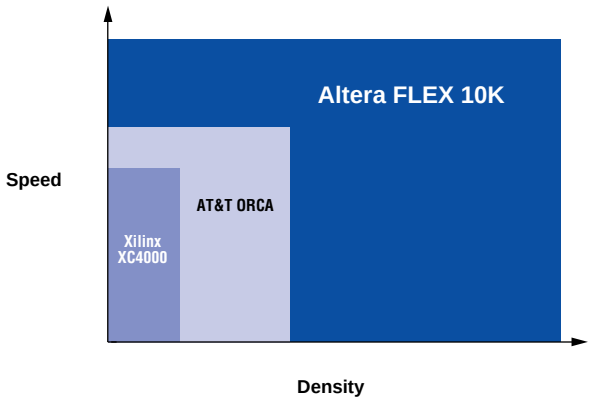
The EPF10K100 is an ideal prototyping and initial production device for ASIC designs. According to market analysts at Dataquest, 80% of all 1996 gate array design starts will require device densities of less than 100,000 gates. As a result, the FLEX 10K family meets the density demands of most gate array designs. In addition to 100,000 gates, the EPF10K100 contains embedded array blocks (EABs) that can integrate specialized arithmetic, digital signal processing (DSP), and large on-chip memory functions. With the EPF10K100, design engineers can now create and prototype 100,000-gate gate array designs with system speeds of over 70 MHz.

FLEX 10K Architecture

The FLEX 10K architecture implements complex functions as efficiently as embedded gate arrays—the fastest-growing segment of the gate array market. Like standard gate arrays, embedded gate arrays implement general logic in a conventional "sea-of-gates" architecture. In addition, embedded gate arrays have dedicated die areas for implementing large, specialized functions. Embedded gate arrays contain functions that are embedded in silicon, which provides reduced die area and increased speed compared to standard gate arrays. However, the embedded functions typically cannot be customized, thus limiting design flexibility.

In contrast, FLEX 10K devices are programmable, providing you with full control over logic while facilitating iterative design changes during debugging. Each FLEX 10K device contains an embedded array and a logic array. The embedded array can implement

Figure 1. Comparison of Speed & Density



Source: Altera Corporation

continued on page 3

Contents

Submit questions & ideas to:

Altera Applications Department
Attention: Martin S. Won (Editor)
2610 Orchard Parkway
San Jose, CA 95134-2020
Tel: (408) 894-7000
Fax: (408) 954-0348
E-mail: martinw@altera.com

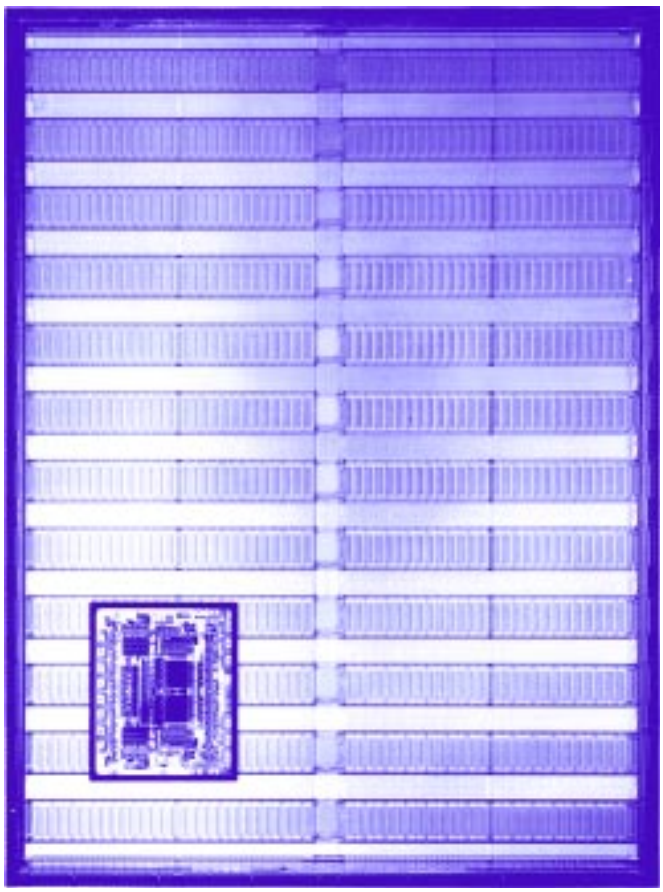
a variety of memory functions or complex logic functions, such as DSP, microcontroller, wide data-path manipulation, and data transformation functions. The logic array has the same function as the sea-of-gates in a gate array: it can implement general logic, such as counters, adders, state machines, and multiplexers. The combination of embedded and logic arrays provides the high performance and high density of embedded gate arrays, enabling you to implement an entire system on a single device.

The EPF10K100 contains 4,992 logic elements grouped into 624 logic array blocks (LABs), 12 EABs (up to 24,576 RAM bits), and up to 406 user I/O pins. Figure 2 compares the EPF10K100 die with the die of Altera's first PLD, the EP300.

Design Support

The EPF10K100 is supported by MAX+PLUS II version 6.2, Altera's fully integrated design environment for programmable logic. In addition to providing interfaces to the industry's most popular EDA tools, MAX+PLUS II offers support for schematic capture and high-level hardware description languages (HDLs), such as VHDL and Verilog HDL, and high-density design libraries such as the library of parameterized modules (LPM).

Figure 2. EPF10K100 & EP300 Dies



Creating complex designs at the 100,000-gate level requires equally complex development tools. As a result, Altera is taking high-level design one step further with Altera MegaCore functions and functions from the Altera Megafunction Partners Program (AMPP). By introducing MegaCore functions and AMPP, Altera is the first programmable logic vendor to supply reusable, synthesizable megafunctions.

MegaCore Functions

Altera MegaCore functions are developed, tested, documented, and licensed by Altera as MAX+PLUS II migration products. Designers can buy these pre-tested functions that are fully optimized for the target Altera device architecture. The first MegaCore functions will be available in the second quarter of 1996, and will consist of several different design files, including VHDL, Verilog HDL, and the Altera Hardware Description Language (AHDL). For more information, refer to "Megafunctions Streamline High-Density Design" on page 7.

AMPP Functions

AMPP focuses on supporting vendors who supply tested, simulated, and synthesizable megafunctions that are optimized for Altera device architectures. Currently, AMPP includes 16 partner companies who specialize in supporting the gate array market, and who have historically produced gate array or standard-cell implementations of their functions. AMPP partners now offer synthesizable netlist files of their products, providing technology and architecture independence. For the latest information on AMPP and for links to the web sites of AMPP partners, go to Altera's world-wide web site at **http://www.altera.com**.

Availability

The EPF10K100 is available in a 503-pin PGA package. Contact your local Altera representative for pricing information. For more information on FLEX 10K devices, contact Altera Customer Marketing.

Devices & Tools Updates

FLEX 10K

EPF10K10 Ships
Altera is currently shipping the 10,000-gate EPF10K10 programmable logic device. This latest member of the FLEX 10K family brings the power of Altera's revolutionary embedded array to the 10,000-gate density level. The EPF10K10 has 576 logic elements (LEs) grouped into 72 logic array blocks (LABs), and 3 embedded array blocks (EABs), yielding up to 6,144 bits of memory.

FLEX 10K devices are supported by Altera's MAX+PLUS II development system version 6.0 and higher, which is available for PC and workstation platforms. Support for the EPF10K10 is available in the entry-level PLS-ES MAX+PLUS II system for PC platforms.

MAX+PLUS II supports the special features of the FLEX 10K architecture via the library of parameterized modules (LPM), which allows you to build common logic and memory functions quickly and efficiently. The MAX+PLUS II Simulator allows you to read the contents of EABs. MAX+PLUS II provides seamless integration with tools from Cadence, Data I/O, Mentor Graphics, Synopsys, Viewlogic, and other leading EDA vendors.

The EPF10K10 is currently offered in a 208-pin plastic quad flat pack (PQFP) package. Future packages will include 144-pin thin quad flat pack (TQFP) and 84-pin plastic J-lead chip carrier (PLCC) packages. Contact your local Altera representative for pricing.

EPF10K50 Update
The price of the EPF10K50GC403-5 has been reduced over 50%. This device was originally priced at $995 each for 100-unit quantities. Additionally, the 50,000-gate EPF10K50 is now available in a 240-pin power quad flat pack (RQFP) package.

EPF10K30 Ships in 208-Pin Package
The 30,000-gate EPF10K30 is now shipping in a 208-pin power quad flat pack (RQFP) package. Contact your local Altera representative for pricing.

FLEX 8000

New FLEX 8000 Packages
The EPF8636A and EPF8820A devices are now available in 208-pin plastic quad flat pack (PQFP) packages. The EPF8820A devices will be available in a 144-pin, 1.0-mm thin QFP (TQFP) package in September. TQFP packages are ideal for space-sensitive or low-profile applications such as personal computer memory card international association (PCMCIA) designs.

More Industrial-Temperature FLEX 8000 Devices
Altera has recently introduced the EPF8452AQI160-3, EPF81188AQI208-3, and EPF81500ARI240-3 devices, expanding the portfolio of industrial-temperature devices. With these new product introductions, each FLEX 8000 device has at least one industrial-temperature version.

FLEX 8000 Price Projections
When compared with devices of comparable density, Altera FLEX 8000 devices have the lowest prices in the programmable logic industry. Volume price projections for the end of 1996 are shown below.

Device	Price *Note (1)*	Quantity
EPF8282ALC84-4	$ 5.00	25,000
EPF8452ALC84-4	$ 7.50	25,000
EPF8636ALC84-4	$13.00	25,000
EPF8820AQC160-4	$19.00	25,000
EPF81188AQC208-4	$29.00	10,000
EPF81500ARC240-4	$49.00	10,000

Note:
(1) Price in U.S. dollars for OEM direct orders.

Discontinued FLEX 8000 Ordering Codes
Altera is discontinuing all non-"A" FLEX 8000 ordering codes. You can use the lower-cost equivalent FLEX 8000A device as a drop-in replacement. The last order date for discontinued FLEX 8000 device ordering codes is May 30, 1996; the last ship date is August 31, 1996. For more information, refer to Product Discontinuance Notice (PDN) 9603. Discontinued ordering codes are listed in the following table.

Discontinued FLEX 8000 Ordering Codes (Part 1 of 2)

Device	Alternative
EPF81188GC232-2	EPF81188AGC232-4
EPF81188GC232-3	EPF81188AGC232-4
EPF81188GI232-2	Consult factory
EPF81188RC240-2	EPF81188ARC240-4
EPF81188GC240-3	EPF81188ARC240-4
EPF81188RI240-3	EPF81188ARI240-4
EPF81500GC280-2	EPF81500AGC280-4
EPF81500GC280-3	EPF81500AGC280-4
EPF81500RC304-2	EPF81500ARC304-4
EPF81500RC304-3	EPF81500ARC304-4
EPF8282LC84-2	EPF8282ALC84-4

Discontinued FLEX 8000 Ordering Codes (Part 2 of 2)	
Device	**Alternative**
EPF8282LC84-3	EPF8282ALC84-4
EPF8282LI84-3	EPF8282ALI84-4
EPF8282TC100-2	EPF8282ATC100-4
EPF8282TC100-3	EPF8282ATC100-4
EPF8282VLC84-3	EPF8282AVLC84-4
EPF8282VLC84-4	EPF8282AVLC84-4
EPF8282VTC100-3	EPF8282AVTC100-4
EPF8282VTC100-4	EPF8282AVTC100-4
EPF8452LC84-2	EPF8452ALC84-4
EPF8452LC84-3	EPF8452ALC84-4
EPF8452LI84-3	EPF8452ALI84-4
EPF8452QC160-2	EPF8452AQC160-4
EPF8452QC160-3	EPF8452AQC160-4
EPF8820GI192-3	Contact Altera
EPF8820RC208-2	EPF8820ARC208-4
EPF8820RC208-3	EPF8820ARC208-4
EPF8820RI208-3	EPF8820ARI208-4

MAX 9000

EPM9400 Ships

Altera is shipping the EPM9400—the fourth member of the MAX 9000 device family—in 208-pin and 240-pin RQFP packages. The EPM9400 is fabricated on a 0.65-micron process and supports in-system programmability (ISP) and JTAG boundary-scan testing. Additional MAX 9000 devices will be produced on a 0.65-micron process in 1996. The following table shows the expected process change schedule.

Device	Expected Process Change Date
EPM9320	October 1996
EPM9480	July 1996
EPM9560	May 1996

MAX 9000 Price Projections

As the MAX 9000 device family moves to more aggressive process technologies, Altera will provide lower prices. The following table shows MAX 9000 volume price projections for the end of 1996 and 1997.

Device	Quantity	End 1996 *(1)*	End 1997 *(1)*
EPM9320LC84-20	25,000	$30.00	$17.00
EPM9400LC84-20	25,000	$39.00	$26.00
EPM9480RC208-20	10,000	$67.00	$35.00
EPM9560RC208-20	10,000	$79.00	$39.00

Note:
(1) Prices in U.S. dollars for OEM direct order.

Contact your local Altera sales representative for package and speed grade pricing not shown in the table above.

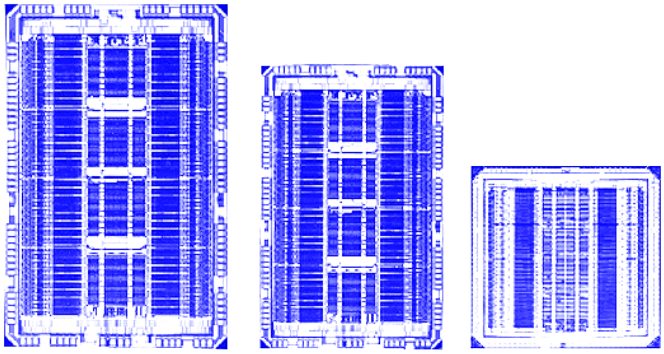
MAX 7000

EPM7128S Ships

Altera is shipping the EPM7128S in a 100-pin QFP package. The EPM7128S is the first member of the MAX 7000S device family, which supports in-system programmability (ISP). The EPM7128S also has JTAG boundary-scan test circuitry and an open-drain output option. The EPM7128S is pin- and programming file-compatible with the EPM7128 and the EPM7128E devices. The EPM7128SQC100 is available in 7.5-ns, 10-ns, and 15-ns speed grades.

The non-"S" EPM7128 was originally fabricated on a 0.8-micron CMOS process in 1992 and moved to a 0.65-micron process in 1995. The EPM7128S is fabricated on a 0.5-micron process. The figure below shows the relative die sizes of these devices.

EPM7128 Process Migration



0.8-µ EPM7128 1995 | 0.65-µ EPM7128 1996 | 0.5-µ EPM7128S 1992

Continuous improvement in both process geometry and device features allows Altera to provide the highest performance and most cost-effective design solutions available.

MAX 7000S Availability

Altera will introduce the remainder of the MAX 7000S device family throughout 1996. The table below shows the schedule for MAX 7000S availability. Check with your local Altera sales representative for specific package and speed grade availability.

Device	Availability
EPM7256S	Q3 1996
EPM7192S	June 1996
EPM7160S	Q4 1996
EPM7128S	Now
EPM7096S	Q4 1996
EPM7064S	Q3 1996
EPM7032S	Q4 1996

continued on page 6

continued from page 5

MAX 5000

Exchange Your MAX 5000 Programming Adapter for Free

Altera has qualified a 0.65-micron EPROM process for MAX 5000 devices and is migrating existing 0.8-micron MAX 5000 devices to 0.65-micron. This change will facilitate long-term support for the MAX 5000 family.

This migration will not change MAX 5000 ordering codes or the MAX 5000 timing parameters shown in the *MAX 5000 Programmable Logic Device Family Data Sheet*. However, new programming adapters are required to program the 0.65-micron devices.

Altera will exchange existing EPM5032, EPM5064, and EPM5130 programming adapters for new adapters *at no charge*. These new adapters are backwards-compatible and support all existing die revisions. The table at the right lists the existing adapters that can be exchanged for new adapters. Altera has already completed an exchange program for EPM5128 and EPM5192 programming adapters.

Contact Altera's Customer Service Department at (800) SOS-EPLD or your local Altera representative to take advantage of this offer.

Existing Adapter	New Adapter
PLED5032	PLMD5032A
PLMD5032	PLMD5032A
PLEJ5032	PLMJ5032A
PLM5032	PLMJ5032A
PLES5032	PLMS5032A
PLEJ5064	PLMJ5064A
PLMJ5064	PLMJ5064A
PLEG5130	PLMG5130A
PLEJ5130	PLMJ5130A
PLMJ5130	PLMJ5130A
PLEQ5130	PLMQ5130A
PLMQ5130	PLMQ5130A

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. Some ADVs and PDNs 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.

Device Family	Device	Last Order Date	Last Shipment Date	Reference
FLEX 8000	Military products (all 883B, DESC, and military temperature grades)	10/31/96	12/31/96	PDN 9513 PDN 9517
MAX 7000	Military products (all 883B, DESC, and military temperature grades)	10/31/96	12/31/96	PDN 9513
FLASHlogic	EPX780 (all packages, temperature grades, and speed grades)	6/28/96	9/30/96	PDN 9601
	EPX740 (all packages, temperature grades, and speed grades)	3/31/97	9/30/97	PDN 9516
MAX 5000	Military EPM5130W device	10/31/96	12/31/96	PDN 9513
	Selected MAX 5000 devices	9/30/96	12/31/96	ADV 9609
	EPM5016 (all packages, temperature grades, and speed grades)	3/31/97	9/30/97	PDN 9516
Classic	EP22V10, EP22V10E, EP310I, EP320I (all packages, temperature grades, and speed grades)	6/28/96	9/30/96	PDN 9516 PDN 9511
	Military products (all 883B, DESC, and military temperature grades)	10/31/96	12/31/96	PDN 9513
	EP220, EP224, EP312, EP324 (all packages, temperature grades, and speed grades)	3/31/97	9/30/97	PDN 9516
	Selected EP6*xx* devices	6/28/96	9/30/96	ADV 9518
	Selected EP9*xx* devices	9/30/96	12/31/96	ADV 9608
	Selected EP18*xx* devices	3/31/97	6/30/97	ADV 9608
Function-Specific	EPS448, EPC1213 military (all 883B, DESC, and military temperature grades)	10/31/96	12/31/96	PDN 9513 PDN 9517
	EPS448, EPS464 (all commercial and industrial temperature grades; military devices have earlier last order and last shipment dates)	3/31/97	9/30/97	PDN 9516

Megafunctions Streamline High-Density Design

The FLEX 10K family, with its unique embedded array architecture, offers both the density and performance to meet increasingly demanding design requirements. With up to 100,000 gates, FLEX 10K is the first programmable logic device (PLD) family that can provide system-on-a-chip integration. However, this increase in density poses a new set of challenges to PLD designers, who must design efficiently and quickly at the 100,000-gate level.

Historically, designers used Boolean equations and schematic capture to develop lower-density PLD designs. As device densities rose above 10,000 gates, designers turned to high-level hardware description languages (HDLs), such as VHDL and Verilog HDL, and high-density design libraries, such as the library of parameterized modules (LPM), to improve their productivity. These design methods permit the designer to describe the behavior of a circuit rather than its implementation, significantly reducing the time required to create and debug a circuit. Synthesis tools then automatically map and optimize the design in silicon.

To continue providing the PLD industry's most powerful and easy-to-use development tools, Altera is taking high-level design one step further. With Altera-created megafunctions, called MegaCore functions, and functions created through the Altera Megafunction Partners Program (AMPP), Altera is the first programmable logic vendor to supply reusable, synthesizable megafunctions.

MegaCore functions are pre-verified HDL design files for complex system-level functions such as microprocessors, microcontrollers, DSP engines, and RAM. MegaCore functions reduce the design task to creating only the custom logic surrounding these commonly used system-level functions, dramatically shortening the design cycle and leveraging existing intellectual property. MegaCores permit designers to focus more time and energy improving and differentiating the design and final product, rather than redesigning common off-the-shelf functions from the ground up.

MegaCore functions are developed, tested, documented, and licensed by Altera as MAX+PLUS II migration products. Designers can buy these pre-tested megafunctions fully optimized for the target Altera device architecture, including FLEX 10K, FLEX 8000, MAX 9000, and MAX 7000 devices. The first MegaCore functions will be available late in the third quarter of 1996, and include the following functions:

- 8051 8-bit processor
- 6502 8-bit processor
- 16450 universal asynchronous receiver/transmitter (UART)
- 6402 UART
- 6850 asynchronous communications interface adapter (ACIA)

Altera MegaCore functions consist of several different design files. A post-synthesis AHDL design file is used for design implementation (i.e., fitting) in the target Altera device. In addition, VHDL or Verilog HDL functional simulation models are supplied for design and debugging with standard EDA simulation tools. MegaCore functions are optimized for the architectural features of Altera devices, which ensures that user-specified performance and die area goals are met.

Megafunctions are also available via the partners in the AMPP. To date, Altera has formed partnerships with 16 companies. Like Altera, AMPP vendors develop megafunctions that are optimized for Altera devices. Altera trains AMPP vendors on Altera device architectures and provides them with the MAX+PLUS II software. Customers can negotiate directly with AMPP vendors to either license standard megafunction products or have a custom design service performed. A listing of current AMPP partner companies and their product specialties appears in "Now Available: First AMPP Megafunctions" on page 27.

Altera & Synopsys Optimize PLD Benefits

Altera participated in the most recent Synopsys Users Group conference, an open technical forum that focused on customer issues. Altera presented information on how to use the Altera/Synopsys Design Kit to program and synthesize designs targeted for Altera FLEX 10K devices, which offer up to 100,000 gates of logic. This article highlights the information that Altera presented at the conference.

High-level logic designers are faced with many decisions when creating a design methodology that optimizes the benefits of programmable logic devices (PLDs). One goal is to combine design and synthesis tools that will work with the widest range of silicon, while minimizing the total development time.

The Altera/Synopsys Design Kit provides a complete design solution, integrating the Altera MAX+PLUS II Compiler with Synopsys logic and synthesis tools. With this process, high-density logic designs are easily targeted to any of Altera's broad range of PLDs without changing the design descriptions or methodologies. This flexibility and integration leads to faster time-to-market. Combined with the densities of Altera devices, these tools give designers a competitive advantage.

Embedded Array Blocks

Altera FLEX 10K embedded array blocks (EABs) are flexible: they can implement either memory or logic functions. The Altera/Synopsys design flow allows designers to effectively access each 2,048-bit EAB for both types of functions. An EAB can implement both synchronous and asynchronous RAM, as well as FIFO functions and dual-port RAM. Synopsys design tools allow easy use of all memory functions, and include complete functional simulation and timing-driven synthesis models. Logic functions are accessible through the Synopsys hierarchical synthesis feature.

The contents of EABs can be modified on-the-fly, allowing designers to change a portion of a design without disturbing the operation of the rest of the device. The FLEX 10K family also links EABs with a continuous interconnect structure to provide predictable speeds regardless of design—an advantage unavailable to gate array designers.

Carry & Cascade Features

The FLEX architecture uses a four-input look-up table (LUT) as its basic building block, offering a fast, efficient implementation of general logic. The architecture also offers carry and cascade chains for implementing counters, adders, and comparators, all of which can use significant logic resources. Implementing these functions in a FLEX 10K or FLEX 8000 device is easy with the Altera/Synopsys Design Kit, which contains a DesignWare library that provides area optimization and high performance. The result is optimal circuit speed in the smallest possible die area.

High-Speed I/O Cells

The FLEX 10K and FLEX 8000 architecture ensures high in-system performance with fast input, setup, and clock-to-output times via high-speed registers located in I/O cells on the periphery of the device. These I/O cell registers are easily accessed in the Synopsys environment by attaching a property to the register.

Partnerships Enhance Value

The close ties between Altera and Synopsys provide a seamless integration of design solutions, tools, and silicon. This relationship is an example of how Altera ensures that users of Altera PLDs have the widest range of tools available.

For example, the Altera Commitment to Cooperative Engineering Solutions (ACCESS) program consists of EDA vendors who have developed design entry, synthesis, verification, and/or device programming products that support Altera PLDs. Altera is continually evaluating and adding new ACCESS partners.

For more information on the Altera/Synopsys interface, refer to the ***Altera & Synopsys Software Interface Guide***.

Altera's FLEX DSP Solution

In the past, designers of digital signal processing (DSP) systems were forced to choose between the flexibility of a DSP processor and the high performance of a DSP ASIC. Now, however, Altera offers a FLEX DSP solution that provides both flexibility and real-time performance.

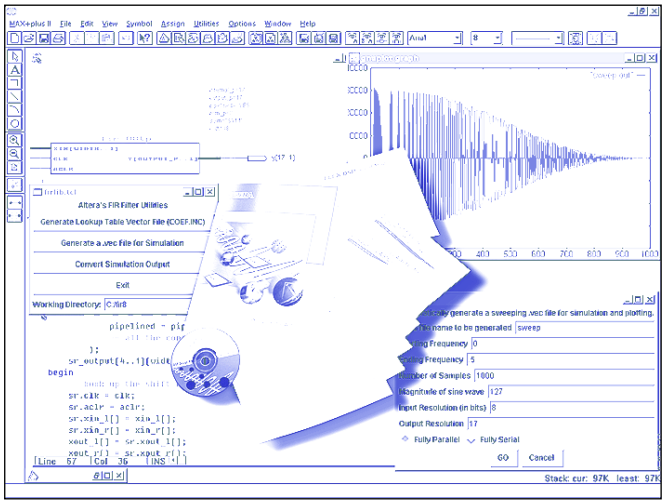
DSP Design Kit

Altera provides a free DSP design kit (see figure at right), which includes customizable building blocks for implementing DSP functions in FLEX 10K and FLEX 8000 devices. The DSP design kit contains support literature and the following reference designs:

- *Parallel finite impulse response (FIR) filters with parameterized features*—8-, 16-, 24-, 32-, and 64-tap functions with parameterized coefficient width, symmetry, and pipelining.
- *Serial FIR filters*—16- and 64-tap functions that accommodate designs with larger tap widths.
- *Arithmetic functions*—Floating-point multiplier, adder/subtractor, and integer divider functions.
- *3 × 3 video convolver*—Convolution filter with parameterized coefficient width and pipelining.

The DSP design kit can improve productivity and shorten a designer's development cycle. For example, one designer created a cable modem design that used a modified FIR filter reference design to create a YUV-to-RGB converter function, and the `fir_32tp` reference design to create a 32-tap filtering function. Both designs were completed in under 10 minutes, saving valuable development time. For information on other DSP design kit applications, refer to *Technical Brief 4 (Using FLEX Devices as DSP Coprocessors).*

The DSP Design Kit is available from Altera Literature Services. Reference designs and support literature are also available from Altera's world-wide web site at **http://www.altera.com**.



Altera at DSPx

Altera presented two papers at the Digital Signal Processing Applications Conference & Exhibition (DSPx) held in Santa Clara, CA from March 11 through 14. Caleb Crome, an Altera Applications Engineer, and Martin Langhammer, a Kaytronics Field Applications Engineer, presented "Image Processing Acceleration Using Altera FLEX Programmable Logic" at the conference. Martin Langhammer also presented "Adaptive Filtering Architecture for Distributed Arithmetic Applications in Altera FLEX." EDN Magazine showcased Altera's FLEX DSP solution in "New Product Presentations."

Preserving Resource Assignments in VHDL Designs

When designing for Altera devices using Synopsys tools, you can pass device resource assignments from a VHDL Design File (**.vhd**) to MAX+PLUS II via an EDIF netlist file. You can use a script that commands the Synopsys Design Compiler to compile the VHDL Design File and make adjustments to the resulting EDIF netlist. For example, the following **dc_shell** script passes pin and I/O cell assignments from the VHDL Design File (**.vhd**), **ministate.vhd** (a sample file included with MAX+PLUS II for workstations) to the EDIF netlist file, **ministate.edf**.

```
read-format vhdl ministate.vhd
compile
write_name_nets_same_as_ports = true
set_attribute find(port, ps2) "CHIP_PIN_LC"
   -type string "ministate@4"
set_attribute find(cell, state_reg[1])
   "LOGIC_OPTION"
   -type string "io_cell_register=on"
edifout_dc_script_flag="altera"
edifout_write_attributes="true"
edifout_write_attributes_properties_list=
   {CHIP_PIN_LC LOGIC_OPTION}
write -format edif -hierarchy - output
   ministate.edf
```

The **dc_shell** script reads and compiles the VHDL design, using the design's port names as net names in the resulting netlist file. Then, the **dc_shell** script generates the pin and I/O cell assignments and adds them to the EDIF netlist file. The two **set attribute** commands (highlighted in blue) assign port **ps2** to pin 4, and implement register **state_reg[1]** in an I/O cell register when the EDIF netlist file is compiled.

You can synthesize the project through MAX+PLUS II or the Altera-provided **setacf** utility. Before you compile the EDIF netlist file in MAX+PLUS II, the project must be assigned to a specific device. Otherwise, MAX+PLUS II defaults to the *AUTO* device selection and resource assignments are not preserved. To assign the project to a specific device in MAX+PLUS II:

1. Choose **Device** (Assign menu). The **Device** dialog box is displayed.
2. Select a device family—e.g., FLEX 8000—from the *Device Family* drop-down list box.
3. Choose a specific device—e.g., EPF8282LC84—in the *Devices* box.
4. Choose **OK**.

When you compile the design, the MAX+PLUS II Compiler synthesizes the design for the chosen device and makes the resource assignments specified in the EDIF netlist file.

New Altera Publications

New Altera publications are available from Altera Literature Services, Altera Express, and the Altera world-wide web site (see "How to Access Altera" on page 30 of this newsletter). Document part numbers are shown in italics.

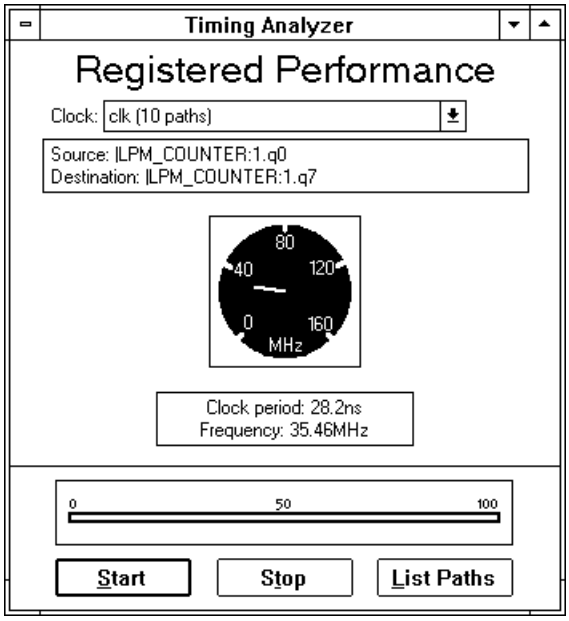
- **PIB 22 (Design Tools for 100,000-Gate Programmable Logic Devices)** ***A-PIB-022-01***
  Discusses how to use ASIC design tools with tools offered by programmable logic vendors to create large designs quickly while optimizing for the silicon features of programmable logic devices (PLDs).
- **FLEX 10K Embedded Programmable Logic Device Family Errata Sheet** ***A-ES-F10K-1.1***
  Provides updated configuration information for EPF10K10, EPF10K50, and EPF10K100 devices with data codes prior to *x*9639 (i.e., devices manufactured prior to the 39th week of 1996).
- **AN 53 (Implementing Multipliers in FLEX 10K Devices)** ***A-AN-053-01***
  Describes how to implement large multipliers using several embedded array blocks (EABs) and compares parallel multiplier and time-domain-multiplexed multiplier implementations.

Analyzing Registered Performance with the Timing Analyzer

The MAX+PLUS II Timing Analyzer permits you to analyze the performance of a design after it is synthesized by the Compiler. You can use the Timing Analyzer to trace all signal paths in a project, as well as to determine critical speed paths and paths that limit the performance of a design.

Calculating Clock Performance of a Design

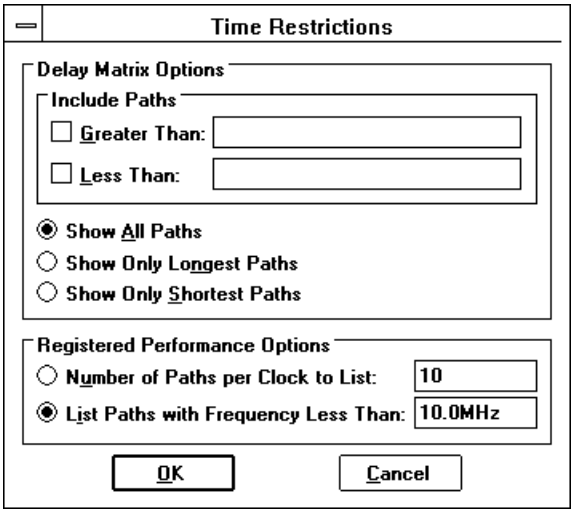
The Timing Analyzer's Registered Performance Display shows the worst-case registered performance, i.e., the maximum clock frequency for every clock signal in the circuit. See the following figure.



You can open the Registered Performance Display by choosing **Registered Performance** (Analysis menu) in the Timing Analyzer. The Registered Performance Display measures the maximum delay from the Q output of all flipflops to the data and clock enable inputs of all other flipflops, including:

- Clock-to-output delay of the source flipflop
- Combinatorial and interconnect delays between the source and destination flipflops
- Internal setup time of the destination flipflop

You can use the default settings in the Registered Performance Display to evaluate whether a design is meeting performance goals. If a design does not meet your goals, you can use the Registered Performance Display to determine the number of paths that fail and the amount by which they fail. A design that is far from meeting performance goals requires a different strategy than a design that meets performance requirements on all but one or two paths. You can use the options in the **Time Restrictions** dialog box (Options menu) to list either all paths that fail to meet a specified clock frequency or a specified number of paths. See the following figure.



Analyzing Bottlenecks in a Design

After the Timing Analyzer finds the longest delay paths, you can view information on the paths by choosing **List Paths**. The Message Processor displays the propagation delays calculated between nodes, including the internal setup time and the propagation delay through the flipflop. You can choose **Locate** to locate and highlight each signal path. Then, you can trace a path in either the source design file or in the Floorplan Editor.

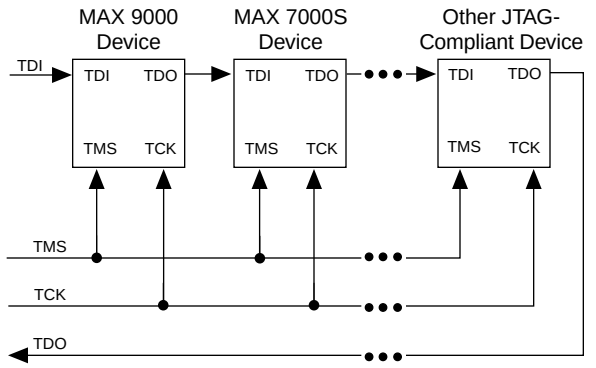
For more information on the Timing Analyzer, refer to MAX+PLUS II Help.

Using the JTAG Interface for Multi-Device Programming

The new features in MAX+PLUS II version 6.0 and higher permit you to program multiple MAX 9000, MAX 7000S, and FLASHlogic devices in-system via the JTAG interface and the BitBlaster serial download cable. You can define the JTAG chain and add the necessary JTAG parameters for non-Altera devices so they can be bypassed during programming. Furthermore, programming multiple devices requires only a single JTAG-compatible header interface.

The JTAG chain can contain any number of IEEE standard 1149.1 JTAG-compliant devices—including Altera devices and non-Altera devices. The following figure shows a sample JTAG chain.

Devices in a JTAG Chain



For more information on programming with the BitBlaster, see the *BitBlaster Serial Download Cable Data Sheet* in the current Altera data book.

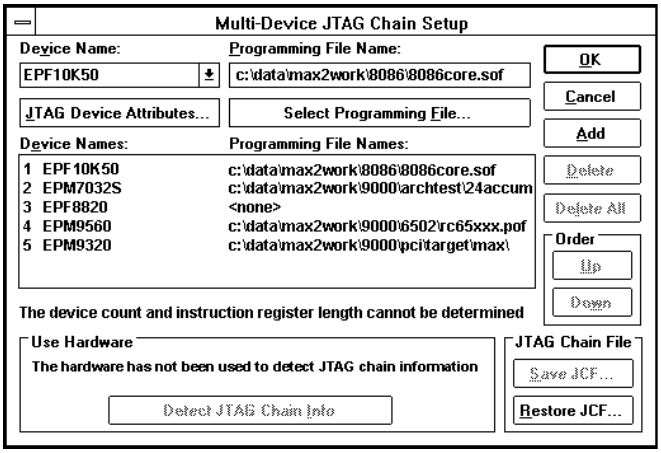
Defining a JTAG Chain

The JTAG chain must be defined before programming so the correct JTAG data is constructed. To define a JTAG chain:

1. In the MAX+PLUS II Programmer, choose **Multi-Device JTAG Chain Setup** (JTAG menu).

2. Build a list of devices in the JTAG chain, along with the appropriate programming files, in the **Multi-Device JTAG Chain Setup** dialog box, which is shown below.

Multi-Device JTAG Chain Setup Dialog Box



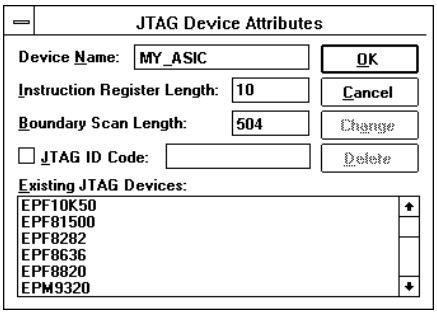
a. Select the device name for the first device in the JTAG chain from the *Device Name* drop-down list box. If necessary, modify the selected device's attributes by choosing the **JTAG Device Attributes** button.

c. To add an Altera device to the chain for programming, enter the programming file in the *Programming File Name* box or select it with the **Select Programming File** button.

d. Choose the **Add** button to add the device to the JTAG chain.

e. To add a non-Altera device to the JTAG chain, enter the name in the *Device Name* box in the **JTAG Device Attributes** dialog box (see the following figure), and specify values in the *Instruction Register Length*, *Boundary Scan Length*, and (optional) *JTAG ID Code* boxes. Choose **OK**.

JTAG Device Attributes Dialog Box



f. Repeat steps a through e for each device in the chain.

3. If necessary, select a device name in the **Multi-Device JTAG Chain** dialog box and choose the **Up** or **Down** buttons in the *Order* box to match the device's position in the list with its physical position in the JTAG chain on the printed circuit board.

4. If necessary, choose the **Delete** button to delete devices from the list.

5. (Optional) Select a device name in the **Multi-Device JTAG Chain** dialog box and choose **Detect JTAG Chain Info** to verify that the device count, JTAG ID code, and total instruction length are correct.

6. Save the JTAG chain information by choosing the **Save JCF** button in the **Multi-Device JTAG Chain** dialog box. Enter a name for the JTAG Chain File (**.jcf**) and choose **OK**.

The JTAG Chain File records the name of each device in the chain, as well as the name of its associated programming file, if any.

Programming Multiple Devices in a JTAG Chain

Follow the steps below to configure the MAX+PLUS II Programmer to program multiple devices with the BitBlaster:

1. Connect the BitBlaster to the PC or workstation, and to the PCB. Configure the BitBlaster according to the directions in the *BitBlaster Serial Download Cable Data Sheet*.

2. Choose the **Hardware Setup** command (Options menu) and select *BitBlaster* in the **Hardware Type** box. Choose **OK**.

3. Turn on the **Multi-Device JTAG Chain** command (JTAG menu) to specify a multi-device JTAG chain.

4. Choose the **Restore JCF** command (JTAG menu) to specify the JCF. The Programmer window displays "Multi-Device JTAG Chain" and the number of programming files.

5. Choose the **Program** button to program the device(s).

MAX+PLUS II Verificaton Options

The new verification options in the MAX+PLUS II Programmer allow you to reduce the cycle time for ISP. These options can reduce both single-device and multi-device JTAG chain programming times. In the **Programming Options** dialog box (Options menu), the following options are turned on by default:

- *Blank-Check Before Programming*—This option automatically checks that a device is blank before being programmed. The Programmer issues a message indicating whether or not the device is blank. Blank-checking is not required for MAX 9000, MAX 7000S, and FLASHlogic devices; therefore, this option can be safely turned off when you are using ISP.

- *Verify During Programming*—This option automatically verifies the device during programming and checks whether each bit is sufficiently programmed. Turning off this option can increase the speed of device programming.

- *Verify After Programming*—This option automatically verifies the device after programming and checks for insufficiently erased bits. Turning off this option can increase the speed of device programming.

- *Test After Programming*—This option automatically performs post-programming functional testing for all devices programmed with the current programming file. This option is not available for multiple devices programmed via ISP, however, it is available for single-device ISP.

For more information, see *Technical Brief 7* (*In-System Programming Times for MAX 9000 Devices*).

FLEX 8000 Devices "Grab" Vitana's Fancy

Contributed by Andrew Nelson, a hardware engineer with Vitana Corporation.

"As we planned to provide easy upgrades to our customers in the field, it was imperative that these devices be reconfigurable or reprogrammable in-system."

—Andrew Nelson, Vitana Corporation

Vitana Corporation, based in Ottawa, offers two- and three-dimensional imaging products as integrated solutions in the resource, industrial design, parts inspection, volume measurement, contour analysis, and medical imaging industries. Vitana is a leader in three-dimensional imaging and has experience in laser-based imaging, digital signal processing (DSP), and high-speed digital design. The company's flagship product, *ShapeGrabber*, is a second-generation laser-scanning system that fits into a standard PC. My job was to design the high-speed digital video circuitry to control and implement the complex DSP algorithms that make *ShapeGrabber* the finest-quality imaging system available. See Figure 1.

The Challenge

From the onset, I knew that *ShapeGrabber* would be a single card, packed with features. Although the core of DSP functionality was provided by a Texas Instruments floating-point DSP device (a TMS320C44), we determined that if we had a support device for video peak detection and sub-pixel interpolation, we could meet our goal of providing real-time video DSP for three-dimensional imaging. Therefore, the first challenge was to identify a device that would allow us to build high-speed digital filters.

In addition to real-time video processing, our other major performance requirement was high host/embedded transfer rates (> 80 Mbytes per second). For that, we decided on a peripheral component interconnect (PCI) bus master/slave interface and a high-speed local bus to support data transfers between the DSP devices on the *ShapeGrabber* card.

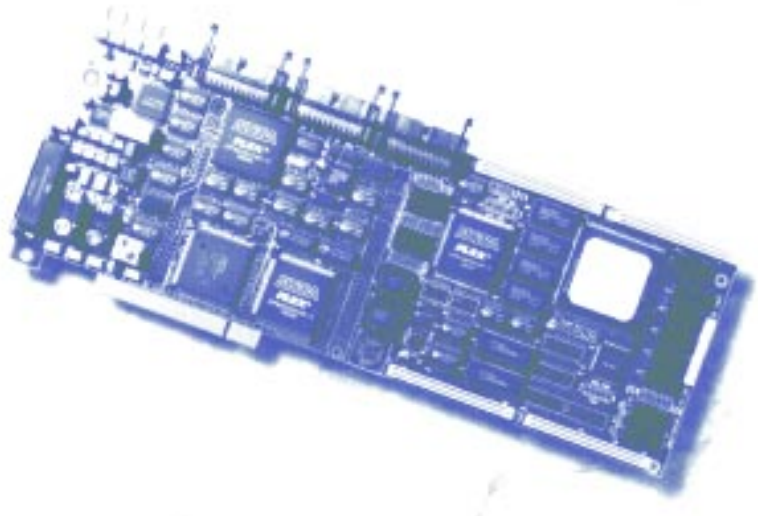
Finally, as we planned to provide easy upgrades to our customers in the field, it was imperative that these devices be reconfigurable or reprogrammable in-system. Another benefit of in-circuit reconfigurability (ICR) is quick and easy prototyping iterations.

The Solution

Three types of devices could have fit our needs: gate arrays, off-the-shelf programmable filters, and programmable logic devices (PLDs). Although gate arrays could be less expensive in large quantities, their long development times and inflexibility made gate arrays ultimately unsuitable for our project. None of the programmable filters we considered were flexible enough for our needs. In particular, these filters could not implement the different types of filters for the upgrades and product modifications we envisioned.

We had some previous experience with programmable anti-fuse devices; however, our need to configure the devices in-circuit in the field precluded our use of these one-time-programmable devices. We also considered using field-programmable gate arrays (FPGAs) with segmented routing, but ultimately decided against them out of concern for their poor routability and our need to guarantee timing and fitting with

Figure 1. ShapeGrabber Board Shot



each design iteration through future upgrades. The PLDs that met all of our needs, at least from the initial examination, were Altera FLEX 8000 devices.

The Implementation

When creating the designs for the FLEX 8000 devices, I began by grouping functions and assigning descriptive names and icons to the devices that would perform the functions. It didn't take long before our devices had lives of their own. The devices and their functionality are summarized below.

Device Name	Device Function	Device Used
TwinPeaks	Video peak detector and sub-pixel interpolator	EPF81188AQC208-3
GateKeeper	PCI to TMS320C44	EPF8452AQC160-3
GrabMaster	Video and I/O control logic	EPF8452ALC84-3
Diablo	TMS320C44 local bus arbiter and memory controller	EPF8282ATC100-3

Of the four designs, TwinPeaks was by far the most challenging. I designed TwinPeaks to be a video preprocessor, with the intention of having the device perform peak detection and validation external to the DSP device, freeing up more of its processing time. For the TwinPeaks design—which included two FIR filters—I chose the EPF81188A device. The peak locator was a 16-bit, 8-tap filter and the peak location validator was a 14-bit, 3-tap filter. Both filters ran at video rates. The TwinPeaks design also contained the sub-pixel interpolator, which interpolates the position of the peak to 1/64 of a pixel. If I had placed these functions in the DSP device, they would have consumed 80% of the device's functionality. By using the EPF81188A as a DSP coprocessor, we were able to meet our speed requirements. Figure 2 shows the TwinPeaks block diagram.

Development Time

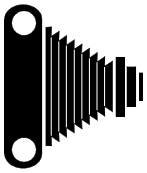
In our first-generation *ShapeGrabber*, we had used anti-fuse-based devices. It took us about two painful months to build that version—painful both in terms of design and cost because we had to throw the devices away after each iteration. However, using Altera's MAX+PLUS II software, we designed the second-generation *ShapeGrabber* in less than a day, and completed full simulation of the entire design in the following two weeks.

continued on page 16

TwinPeaks

GateKeeper

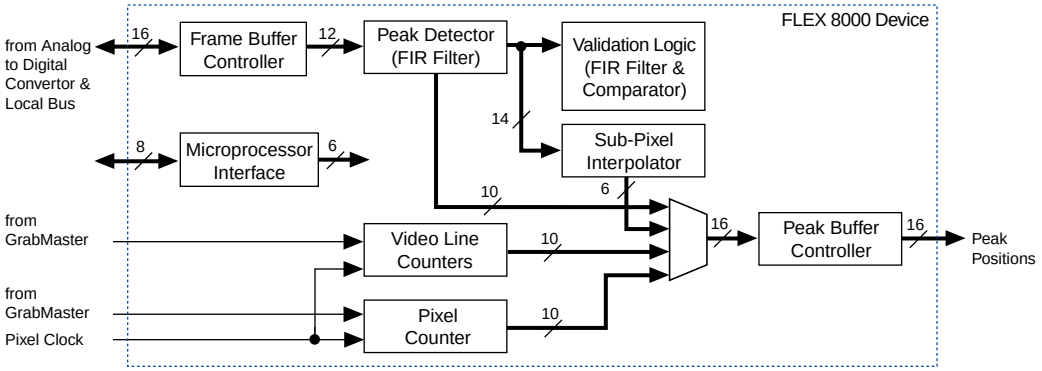
GrabMaster



Diablo

Vitana Corporation
25-5470 Canotek Rd.
Gloucester, Ontario
K1J9H3
Tel. (613) 749-4445
Fax (613 749-4087

Figure 2. TwinPeaks Block Diagram *Note (1)*



Note:

(1) The TwinPeaks design is patented by Vitana Corporation.

FLEX 8000 Devices "Grab" Vitana's Fancy
continued from page 15

To create the design, I used MAX+PLUS II version 6.0, which supports the library of parameterized modules (LPM). This solution permitted me to create custom building blocks that were not limited by conventional logic sizes. Because the LPM functions were optimized for any target device I chose, I was able to concentrate on my design without worrying about the lower-level implementation details. MAX+PLUS II version 6.0 also had better device fitting support, as evidenced when the software fit designs faster and easier than the previous version.

FLEX 8000 devices support in-circuit reconfigurability (ICR), which vastly decreased the amount of time it took me to prototype *ShapeGrabber*. I could make a design change, implement it, and physically test it in minutes. Using ICR and the Altera Hardware Description Language (AHDL), I was able to make, simulate, and fully test 10 to 20 design changes a day. With these rapid prototyping features, our time-to-market decreased significantly.

In-Field Upgrades via ICR & the PCI Bus

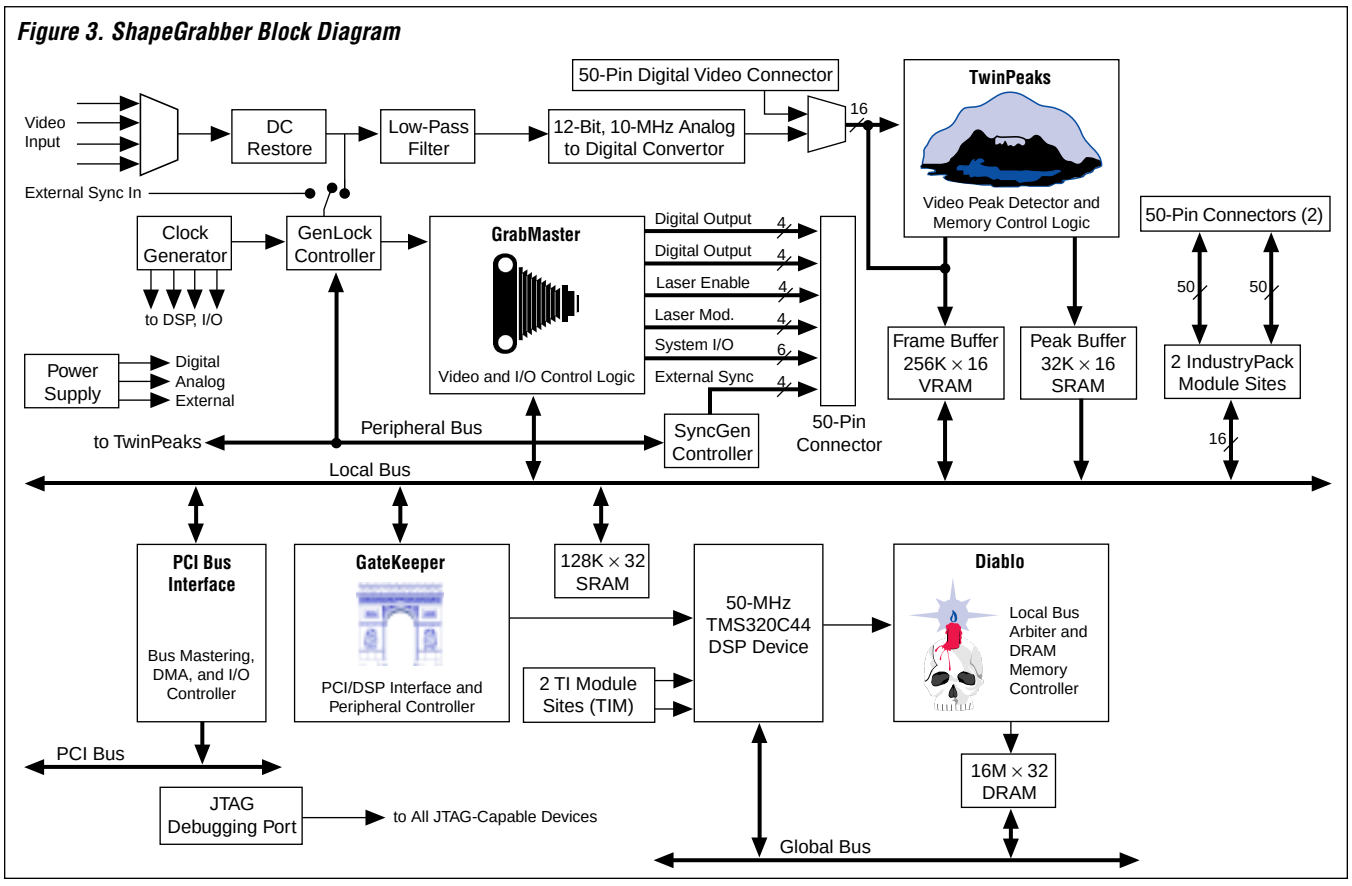
One of the features we had planned for this generation of *ShapeGrabber* was the ability to perform in-field upgrades and modifications to meet special customer needs. We accomplished this by reconfiguring the FLEX 8000 devices via the PCI bus. We wrote a software driver that resides on the host platform and converts a FLEX 8000 programming file to configuration data. The driver also downloads the configuration data to the appropriate FLEX 8000 device via the PCI bus upon startup. All that is required for upgrades or modifications is a single new programming file that can be sent to the customer electronically. Figure 3 provides a block diagram of *ShapeGrabber*.

Some of the modifications we are planning include changing the filtering in the TwinPeaks design. For example, instead of peak detection, we may consider valley detection, or performing low-pass or high-pass filtering on the video signal. In any case, ICR gives us a greater ability to service customers in the field.

Conclusion

Having worked with a number of device architectures and design packages, I can truly say that using FLEX 8000 devices along with the MAX+PLUS II software were the ideal choices for creating our product. *ShapeGrabber* meets our goal of being the most advanced three-dimensional imaging product worldwide.

Figure 3. ShapeGrabber Block Diagram



FLEX 10K Memory Support in Synopsys

The Altera/Synopsys interface offers full memory support for the FLEX 10K device family, including asynchronous and synchronous RAM and ROM, cycle-shared FIFO functions, and cycle-shared dual-port RAM. You can use the Altera software utility **genmem**—which is provided with Altera's Synopsys support for MAX+PLUS II—to generate functional simulation models and timing models for designs with different sizes of RAM and ROM. When you install MAX+PLUS II, **genmem** is placed in the **/usr/maxplus2/bin** directory.

To run the **genmem** utility, type the following command at a UNIX prompt:

```
genmem <memory type><memory size> [-vhdl]
    [-verilog][-viewlogic] ↵
```

The variable <*memory type*> specifies a valid memory type, including:

`asynram`	Asynchronous RAM
`asynrom`	Asynchronous ROM
`synram`	Synchronous RAM
`synrom`	Synchronous ROM
`csfifo`	Cycle-shared FIFO function
`csdpram`	Cycle-shared dual-port RAM

The variable <*memory size*> specifies the size of the memory model using *word* × *width* format, where:

word	Must be between 8 and 32,768 words deep
width	Must be between 1 and 32 bits wide

The word and width values must be separated by an `x`, such as `256x15`.

The following options are available:

`-vhdl`	generates VHDL output (default setting)
`-verilog`	generates Verilog HDL output
`-viewlogic`	generates VHDL output for Viewlogic

For example, to create a 256 × 15 asynchronous ROM in VHDL output, type:

```
genmem asynrom 256x15 -vhdl ↵
```

The **genmem** utility generates an asynchronous ROM model (**asyn_rom_256x15.vhd**) that is 256 words deep and 15 bits wide, a Component Declaration template (**asyn_rom_256x15.cmp**), and a timing model (**asyn_rom_256x15.lib**).

The timing model contains pin-to-pin delay information that can be used by the Synopsys Design Compiler. For the Design Compiler to access the timing information, you must add the asynchronous ROM timing model to the existing library by typing the following commands at a UNIX prompt:

```
read -f db flex10k.db ↵
update_lib flex10K <RAM/ROM function
    name>.lib ↵
```

During compilation, the Synopsys VHDL Compiler for VHDL or the HDL Compiler for Verilog HDL automatically translates the design into a Synopsys database format with the extension **.db**. To update the database file from the previous example (**flex10k.db**), you can add the following optional command:

```
write_lib flex10K -o flex10k.db ↵
```

Designs that contain RAM or ROM functions must include the bus structure in the EDIF netlist file that is generated from the design. To include the bus structure in the EDIF netlist file, set `edifout_no_array = "false"` and add the following lines to the **.synopsys_dc.setup** file in the Synopsys Design Compiler:

```
compile_fix_multiple_port_nets = true
bus_naming_style = "%s<%d>";
bus_dimension_separator_style = "><"
bus_inference_style = "%s<%d>";
```

ROM requires an initialization file in a Hexadecimal (Intel-format) file (**.hex**). The Synopsys **intelhex** utility, provided with the Synopsys VHDL System Simulator (VSS) Software Tool, can be used to translate a Synopsys memory file into a Hex File. Refer to the *Synopsys VHDL System Simulator Software Tool* manual for details about using the **intelhex** utility.

For more information about Synopsys support of memory functions in Altera devices, refer to the ***Synopsys & MAX+PLUS II Software Interface Guide***, available from Altera Literature Services or on Altera's world-wide web site at **http://www.altera.com**.

Implementing Internal Buses in MAX+PLUS II

Beginning with version 6.0, MAX+PLUS II provides internal bus synthesis support for all Altera devices. Although Altera devices do not support internal tri-state buses, synthesis support allows designers to use internal tri-state bus structures in their designs. MAX+PLUS II then converts the bus structures to multiplexers. Figure 1 shows a schematic of an internal bus.

In Figure 1, the input bus to each of the TRI buffers and the net from the junction point of the internal bus are named. For example, the junction net is named d3[5..0]. This naming is necessary because the single TRI buffer on the schematic really represents a primitive array of TRI buffers, one for each bit of the bus. MAX+PLUS II synthesizes the appropriate number of TRI buffers according to the size of the bus, which is specified by the attached name. If the bus is not named, MAX+PLUS II may report the following error:

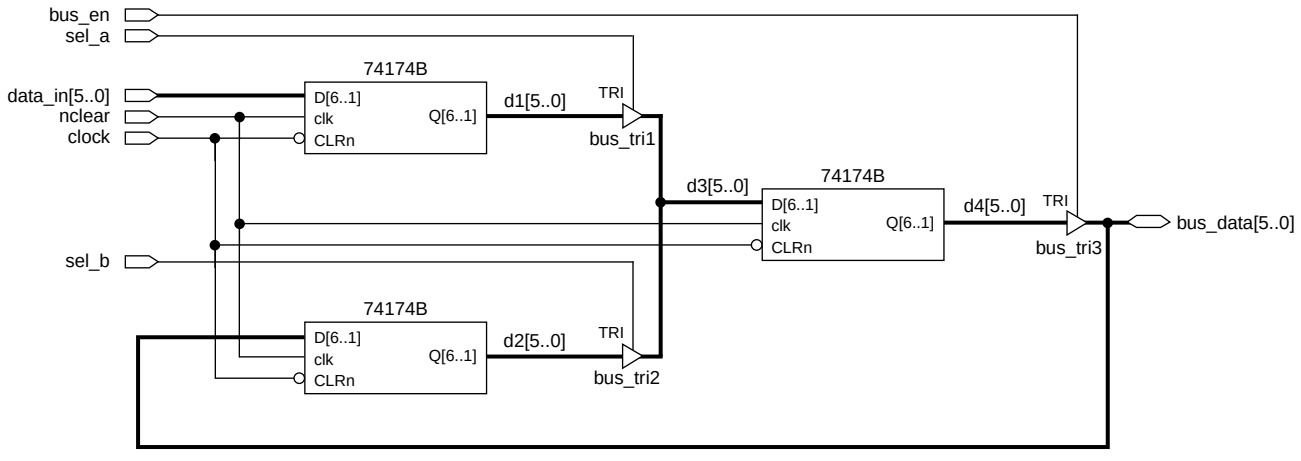
Width Mismatch in pinstub <*name*> (<*instance number.name*>).

The Altera Hardware Description Language (AHDL) equivalent of the design in Figure 1 is shown below.

```
INCLUDE "74174b";
SUBDESIGN tri_ex
(
    data_in[5..0], bus_en, sel_a      : INPUT;
    sel_b, nclear, clock              : INPUT;
    bus_data[5..0]                    : BIDIR;
)

VARIABLE
    bus_tri1[5..0] : TRI;
    bus_tri2[5..0] : TRI;
    bus_tri3[5..0] : TRI;
    d1[5..0]       : NODE;
    d2[5..0]       : NODE;
    d3[5..0]       : TRI_STATE_NODE;
    d4[5..0]       : NODE;

BEGIN
    d1[] = 74174b(nclear, clock, data_in[]);
    bus_tri1[].in = d1[];
    bus_tri1[].oe = sel_a;
    d3[] = bus_tri1[].out;

    d2[] = 74174b(nclear, clock, bus_data[]);
    bus_tri2[].in = d2[];
    bus_tri2[].oe = sel_b;
    d3[] = bus_tri2[].out;

    d4[] = 74174b(nclear, clock, d3[]);
    bus_tri3[].in = d4[];
    bus_tri3[].oe = bus_en;
    bus_data[] = bus_tri3[].out;
END;
```

The node at the junction of the internal bus connection, d3[5..0], must be defined as a TRI_STATE_NODE. This definition directs MAX+PLUS II how to synthesize this junction.

Figure 1. Internal Buses in MAX+PLUS II



The following equations are synthesized for `bus_data1`:

```
bus_data1 = TRI(d41, bus_en);
d41       = DFFE( _EQ001 $ VCC,
clock, nclear,
            VCC, VCC);
   _EQ001 = (!d21 & sel_b) # (!d11 &
sel_a);
```

Figure 2 shows the graphical representation of these equations as they are implemented in logic, while Figure 3 shows the simplified representation of the equations (excluding the XOR gate).

For more information on tri-state buses, refer to MAX+PLUS II Help.

Figure 2. Schematic Equivalent of Equations for bus_data1

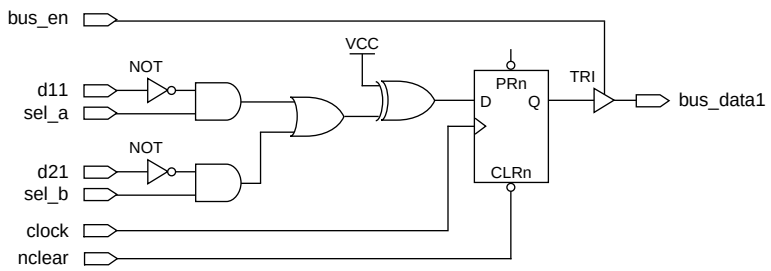
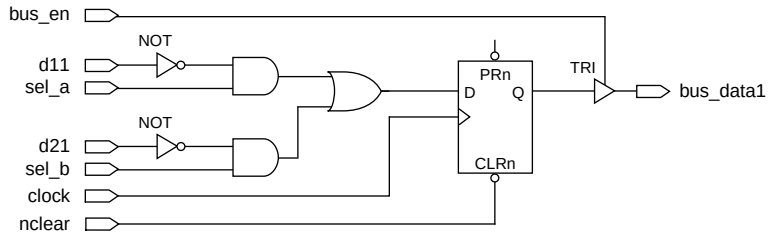


Figure 3. Simplified Schematic Equivalent of Equations for bus_data1



Altera Device Selection Guide

All current information for the Altera FLEX 10K, FLEX 8000, MAX 9000, and MAX 7000 devices is listed here. Information on other Altera products is given in the Altera ***1995 Data Book***. Contact Altera or your local sales office for current product availability.

FLEX 10K Devices

	Typical Gates	Pin/Package Options	I/O Pins (1)	Temp.	Speed Grade	Flip-flops	Logic Elements	RAM Bits
EPF10K10	10,000	84-Pin PLCC, 144-Pin TQFP, 208-Pin RQFP	59, 107, 134	C	-4	720	576	6,144
EPF10K10	10,000	84-Pin PLCC, 144-Pin TQFP, 208-Pin RQFP	59, 107, 134	C, I	-5	720	576	6,144
EPF10K20	20,000	208-Pin RQFP, 240-Pin RQFP	147, 198	C	-4	1,344	1,152	12,288
EPF10K20	20,000	208-Pin RQFP, 240-Pin RQFP	147, 198	C, I	-5	1,344	1,152	12,288
EPF10K30	30,000	208-Pin RQFP, 240-Pin RQFP, 319-Pin PGA, 356-Pin BGA	147, 198, 246	C	-4	1,968	1,728	12,288
EPF10K30	30,000	208-Pin RQFP, 240-Pin RQFP, 319-Pin PGA, 356-Pin BGA	147, 198, 246	C, I	-5	1,968	1,728	12,288
EPF10K40	40,000	208-Pin RQFP, 240-Pin RQFP	147, 189	C	-4	2,576	2,304	16,384
EPF10K40	40,000	208-Pin RQFP, 240-Pin RQFP	147, 189	C, I	-5	2,576	2,304	16,384
EPF10K50	50,000	240-Pin RQFP, 356-Pin BGA, 403-Pin PGA	189, 274, 310	C	-4	3,184	2,880	20,480
EPF10K50	50,000	240-Pin RQFP, 356-Pin BGA, 403-Pin PGA	189, 274, 310	C, I	-5	3,184	2,880	20,480
EPF10K70	70,000	240-Pin RQFP, 503-Pin PGA	189, 358	C	-4	4,096	3,744	18,432
EPF10K70	70,000	240-Pin RQFP, 503-Pin PGA	189, 358	C, I	-5	4,096	3,744	18,432
EPF10K100	100,000	503-Pin PGA	406	C	-4	5,392	4,992	24,576
EPF10K100	100,000	503-Pin PGA	406	C, I	-5	5,392	4,992	24,576

(1) Six I/O pins are dedicated inputs.

continued on page 20

continued from page 19

FLEX 8000 Devices

	Usable Gates	Pin/Package Options	I/O Pins *(2)*	Temp.	Speed Grade	Flip-flops	Logic Elements
EPF8282A	2,500	84-Pin PLCC, 100-Pin TQFP	68, 78	C	A-2	282	208
EPF8282A	2,500	84-Pin PLCC, 100-Pin TQFP	68, 78	C, I	A-3	282	208
EPF8282A	2,500	84-Pin PLCC, 100-Pin TQFP	68, 78	C, I	A-4	282	208
EPF8282AV *(1)*	2,500	84-Pin PLCC, 100-Pin TQFP	68, 78	C	A-3	282	208
EPF8282AV *(1)*	2,500	84-Pin PLCC, 100-Pin TQFP	68, 78	C, I	A-4	282	208
EPF8452A	4,000	84-Pin PLCC, 100-Pin TQFP, 160-Pin PGA/PQFP	68, 120	C	A-2	452	336
EPF8452A	4,000	84-Pin PLCC, 100-Pin TQFP, 160-Pin PGA/PQFP	68, 120	C, I	A-3	452	336
EPF8452A	4,000	84-Pin PLCC, 100-Pin TQFP, 160-Pin PGA/PQFP	68, 120	C, I	A-4	452	336
EPF8636A	6,000	84-Pin PLCC, 160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	68, 110, 136	C	A-2	636	504
EPF8636A	6,000	84-Pin PLCC, 160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	68, 110, 136	C, I	A-3	636	504
EPF8636A	6,000	84-Pin PLCC, 160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	68, 110, 136	C, I	A-4	636	504
EPF8820A	8,000	144-Pin TQFP, 160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP, 225-Pin BGA	120, 152	C	A-2	820	672
EPF8820A	8,000	144-Pin TQFP, 160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP, 225-Pin BGA	120, 152	C, I	A-3	820	672
EPF8820A	8,000	144-Pin TQFP, 160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP, 225-Pin BGA	120, 152	C, I	A-4	820	672
EPF81188A	12,000	208-Pin PQFP, 232-Pin PGA, 240-Pin RQFP	148, 184	C	A-2	1,188	1,008
EPF81188A	12,000	208-Pin PQFP, 232-Pin PGA, 240-Pin RQFP	148, 184	C, I	A-3	1,188	1,008
EPF81188A	12,000	208-Pin PQFP, 232-Pin PGA, 240-Pin RQFP	148, 184	C, I	A-4	1,188	1,008
EPF81500A	16,000	240-Pin RQFP, 280-Pin PGA, 304-Pin RQFP, BGA	181, 208	C	A-2	1,500	1,296
EPF81500A	16,000	240-Pin RQFP, 280-Pin PGA, 304-Pin RQFP, BGA	181, 208	C, I	A-3	1,500	1,296
EPF81500A	16,000	240-Pin RQFP, 280-Pin PGA, 304-Pin RQFP, BGA	181, 208	C	A-4	1,500	1,296

(1) V indicates 3.3-V voltage supply.
(2) Four I/O pins are dedicated inputs.

MAX 9000 Devices

	Macrocells	Pin/Package Options	I/O Pins *(1)*	Temp.	Speed Grade
EPM9320	320	84-Pin PLCC, 208-Pin RQFP, 280-Pin PGA	60, 132, 168	C	-12
EPM9320	320	84-Pin PLCC, 208-Pin RQFP, 280-Pin PGA	60, 132, 168	C	-15
EPM9320	320	84-Pin PLCC, 208-Pin RQFP, 280-Pin PGA	60, 132, 168	C, I	-20
EPM9400	400	84-Pin PLCC, 208-Pin RQFP, 240-Pin RQFP	59, 139, 159	C	-12
EPM9400	400	84-Pin PLCC, 208-Pin RQFP, 240-Pin RQFP	59, 139, 159	C	-15
EPM9400	400	84-Pin PLCC, 208-Pin RQFP, 240-Pin RQFP	59, 139, 159	C, I	-20
EPM9480	480	208-Pin RQFP, 240-Pin RQFP	146, 175	C	-15
EPM9480	480	208-Pin RQFP, 240-Pin RQFP	146, 175	C, I	-20
EPM9560	560	208-Pin CQFP, 240-Pin RQFP, 280-Pin PGA, 304-Pin RQFP	153, 191, 216	C	-15
EPM9560	560	208-Pin CQFP, 240-Pin RQFP, 280-Pin PGA, 304-Pin RQFP	153, 191, 216	C, I	-20

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

MAX 7000 Devices

	Macrocells	Pin/Package Options	I/O Pins *(2)*	Temp.	Speed Grade	t_{PD} (ns)	f_{CNT} (MHz)
EPM7032, EPM7032S	32	44-Pin PLCC/TQFP	36	C	-5	5	178.6
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	-10	10	100
EPM7032	32	44-Pin PLCC/TQFP	36	C, I	-12	12	90.9
EPM7032, EPM7032S	32	44-Pin PLCC/TQFP	36	C, I	-15	15	76.9
EPM7032V *(1)*	32	44-Pin PLCC/TQFP	36	C	-12	12	90.9
EPM7032V *(1)*	32	44-Pin PLCC/TQFP	36	C	-15	15	76.9
EPM7032V *(1)*	32	44-Pin PLCC/TQFP	36	C, I	-20	20	62.5
EPM7064, EPM7064S	64	44-Pin PLCC/TQFP, 68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	36, 52, 68	C	-6	6	150
EPM7064, EPM7064S	64	44-Pin PLCC/TQFP, 68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	36, 52, 68	C	-7	7.5	125
EPM7064, EPM7064S	64	44-Pin PLCC/TQFP, 68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	36, 52, 68	C, I	-10	10	100
EPM7064	64	44-Pin PLCC/TQFP, 68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	36, 52, 68	C	-12	12	90.9
EPM7064, EPM7064S	64	44-Pin PLCC/TQFP, 68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	36, 52, 68	C, I	-15	15	76.9
EPM7096, EPM7096S	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C	-6	6	150
EPM7096, EPM7096S	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C	-7	7.5	125
EPM7096, EPM7096S	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C, I	-10	10	100
EPM7096	96	68-Pin PLCC, 64-Pin PLCC, 100-Pin PQFP	52, 64, 76	C	-12	12	90.9
EPM7096, EPM7096S	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C, I	-15	15	76.9
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C	-7	7.5	125
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C, I	-10(P)	10	100
EPM7128E	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C	-12	12	90.9
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP, 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	20	62.5
EPM7128SV *(1)*	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	64, 84, 100	C	-10	10	100
EPM7128SV *(1)*	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	64, 84, 100	C	-15	15	76.9
EPM7160E, EPM7160S	160	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	64, 84, 104	C	-7	7.5	125
EPM7160E, EPM7160S	160	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	64, 84, 104	C, I	-10(P)	10	100
EPM7160E	160	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C	-12	12	90.9
EPM7160E, EPM7160S	160	84-Pin PLCC, 100-Pin PQFP, 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	20	62.5
EPM7192E, EPM7192S	192	160-Pin PQFP/PGA	124	C	-7	7.5	125
EPM7192E, EPM7192S	192	160-Pin PQFP/PGA	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	124	C, I	-15	15	76.9
EPM7192E	192	160-Pin PQFP/PGA	124	C, I	-20	20	62.5
EPM7256E, EPM7256S	256	160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	132, 164	C	-7	7.5	125
EPM7256E, EPM7256S	256	160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	132, 164	C	-10	10	100
EPM7256E	256	160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	132, 164	C	-12(P)	12	90.9
EPM7256E, EPM7256S	256	160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	132, 164	C, I	-15	15	76.9
EPM7256E	256	160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	132, 164	C, I	-20	20	62.5

(1) V indicates 3.3-V voltage supply.

(2) Four I/O pins are dedicated inputs.

QUESTIONS & ANSWERS

Q I just received the latest MAX+PLUS II upgrade. I want to install the upgrade, but I'm in the middle of a project. What should I do?

A Altera recommends using the same version of MAX+PLUS II throughout a project cycle. However, if you decide to upgrade to a newer version, you should archive your project so that you can reload the previous version if necessary. For directions on archiving a project, search for "Archiving a Project" in MAX+PLUS II Help.

Q I'm using an EPF10K50 in a 403-pin PGA package. Does Altera sell sockets for this device?

A Altera does not sell sockets for this device, but recommends the third-party socket manufacturers listed below:

Socket Type	Vendor	Part Number
Test and Burn-in Sockets	AMP	1-382320-7
	3M/Textool	2-0403-08450-390-019-002
	Yamaichi	NP-178-64401-Ks14828
Low-profile, printed circuit board sockets	McKenzie	PZA Family
	Mil Max/Preci-Dip	Series 518 Family

Q I'm creating a custom configuration for the EPF81188 on my printed circuit board. What is the exact bit count for the configuration bitstream that MAX+PLUS II generates for this device?

A The following table lists the exact bit counts for single-device configuration bitstreams generated by MAX+PLUS II for FLEX 8000 devices. Because of the overhead from combining multiple bitstreams, the bit count for a multiple-FLEX 8000 device configuration bitstream is *not* the sum of the single-device bitstream bit counts. Summing the single-device bit counts can be used as an estimation of the lower bound of a multi-device bit count.

Device	Single-Device Bit Count *Note (1)*
EPF8282A	40,960
EPF8452A	63,112
EPF8636A	93,304
EPF8820A	123,496
EPF81188A	183,880
EPF81500A	253,112, *Note (2)*

Notes:

(1) Bit counts are subject to change without notice.

(2) Add 129 bits when using EPC1213 Configuration EPROMs.

*Q I am creating a large FLEX 8000 design on the PC. The **read.me** file for MAX+PLUS II says that Altera recommends a minimum of 64 Mbytes of memory (RAM and virtual memory) for FLEX 8000 projects. How can I find out how much memory I have and how can I increase that amount?*

A To determine how much memory you have and how to increase the available memory, refer to "Maximizing Available Memory in Windows 3.1 & Windows for Workgroups 3.11" on page 7 of the ***MAX+PLUS II Getting Started*** manual.

Q Does MAX+PLUS II support open-drain I/O pins?

A MAX+PLUS II version 6.0 and higher supports open-drain I/O pins for FLEX 10K devices via the OPNDRN primitive. Open-drain I/O pins for MAX 7000S devices are supported in MAX+PLUS II version 6.1 and higher. If the input to an OPNDRN primitive is low, the output will be low. If the input is high, the output will be a high-impedance logic level. If an OPNDRN primitive is used in a design targeted for Altera devices other than FLEX 10K and MAX 7000S, the OPNDRN primitive is converted to a TRI primitive.

If you turn on the *Automatic Open-Drain Pins* option in the **Global Project Logic Synthesis** dialog box (Assign menu) for a FLEX 10K or MAX 7000S design, the MAX+PLUS II Compiler converts the following structures to the OPNDRN primitive:

- TRI primitive whose output enable input is fed by any signal, but whose primary input is fed by a GND primitive
- A TRI primitive whose output enable input is fed by the complement of its primary input

When you use an OPNDRN buffer, you must observe the following rules:

- An OPNDRN can drive only one BIDIR or BIDIRC pin
- If an OPNDRN feeds logic, it must also feed a BIDIR or BIDIRC pin
- If an OPNDRN feeds a BIDIR or BIDIRC pin, it cannot feed any other outputs

Q *Can I use my design files created with MAX+PLUS II for PCs in MAX+PLUS II for workstations?*

A Yes, and vice versa. With the MAX+PLUS II extended feature set for your workstation (PLSM-EXTWS), all of your files—including Graphic Design Files (**.gdf**)—can be used on both the workstation and the PC. With the base MAX+PLUS II workstation product (PLS-WS), you can transfer AHDL design files and programming files between platforms.

Q *In the past I have used MAX+PLUS II on a PC, but I have recently starting using MAX+PLUS II on a workstation. How does MAX+PLUS II handle the UNIX environment's case sensitivity?*

A In the UNIX workstation environment, MAX+PLUS II uses all lowercase filenames for Altera-provided megafunctions and macrofunctions, as well as their corresponding Symbol File (**.sym**) and Include File (**.inc**) names. You should use lowercase characters for all references to these functions in text-based design files. Altera also recommends using all lowercase names for your own megafunctions, macrofunctions, Symbol Files, and Include Files. However, Altera provides a variable that you can add to your **maxplus2.ini** file to lessen the impact of the UNIX environment's case-sensitivity during compilation: FILE_CASE_SEARCH_NODE=<*setting*> where <*setting*> is one of the following:

LOWER	MAX+PLUS II searches for the exact filename case. If the exact case is not found, it searches for a lowercase version of the name. This is the default setting for MAX+PLUS II version 6.0 and higher.
SMART	MAX+PLUS II searches for the exact filename case. If the exact case is not found, it searches for a lowercase version of the name; if that is not found, it searches for the name in any case.
ALL	MAX+PLUS II searches for the exact filename case. If the exact case is not found, it searches for the name in any case. This option can lower the speed performance of MAX+PLUS II.

Altera does not recommend creating multiple files whose names differ only by case in the same directory.

Q *When I use MAX+PLUS II to compile an EDIF file generated by the Synopsys FPGA Compiler, I get the following error:* Can't find design file 'LUT'.

What am I doing wrong?

A Before you generate the EDIF netlist file with the Synopsys tools, you must execute the **replace_fpga** command. The Synopsys FPGA Compiler maps logic to a cell structure, while MAX+PLUS II looks for a gate-level netlist in the EDIF file. The **replace_fpga** command instructs the Synopsys tools to replace the look-up table (LUT) cell structure with a gate-level netlist in the EDIF file.

Q *I installed MAX+PLUS II on my workstation. When I try to run the software, I get the following message:*

```
maxplus2 was not installed for sunos in /usr/
maxplus2/bin
```

What should I do?

A The MAX2_HOME variable may not be set properly. The error message indicates that MAX2_HOME is pointing to **/usr/maxplus2/bin**—instead of **/usr/maxplus2** or the directory where you installed MAX+PLUS II. To reset this variable, place the following line in your **.cshrc** file:

```
setenv MAX2_HOME <path to MAX+PLUS II>
```

Then, source the **.cshrc** file.

Q *I would like to upgrade from SunOS 4.1.5. Does MAX+PLUS II support Solaris 2.5?*

A Yes. MAX+PLUS II versions 6.0 and higher support Solaris 2.5. You can also run MAX+PLUS II version 6.0 and higher on the Solaris-based Sun Ultra 1 or Ultra 2 workstations. Because MAX+PLUS II takes advantage of the 64-bit mode of the UltraSPARC chip used in the Sun Ultra 1, you can expect significantly higher performance when running MAX+PLUS II on the Ultra 1 or Ultra 2 workstations.

Q *I have a Case Statement in my Altera Hardware Description Language (AHDL) design that has several bits labeled as "Don't Care." For example:*

```
CASE b[] IS
    WHEN B"XXX10000" =>
        sig_out[] = temp[];
```

(continued on page 28)

Parameterized Function Support in AHDL

The industry-standard library of parameterized modules (LPM) provides architecture-independent functions that can simplify the design process. The LPM is supported by Altera and other EDA vendors such as Viewlogic, Mentor Graphics, Cadence, and VeriBest. The LPM includes logic blocks that range from simple functions (e.g., AND gates and registers) to more complex functions (e.g., adders and multiplexers). In addition to simplifying design entry, designing with the LPM and other parameterized functions ensures efficient device utilization. To support parameterized functions, Altera has added new features to MAX+PLUS II and the Altera Hardware Description Language (AHDL).

Explicit Iterative Assignments

In the past, AHDL supported iteration as an implicit statement. For example, the following statement implies four separate assignments:

```
bus_one[] = bus_two[3..0];
```

With MAX+PLUS II version 6.0 and higher, AHDL also supports explicit iterative logic assignments using the For Generate Statement. The following example illustrates how to make explicit assignments for the parameterized function **my_adder**.

```
PARAMETERS
(
    width= 8
);

SUBDESIGN my_adder
(
    a[width..1],
    b[width..1],
    cin                    : INPUT;
    sum[width..1], cout    : OUTPUT;
);

VARIABLE
    carr_ch[width+1..1]    : NODE;

BEGIN
    carr_ch[1] = cin;
    FOR i IN 1 to width GENERATE
        sum[i] = a[i] $ b[i] $ carr_ch[i];
        carr_ch[i+1]  = a[i] & b[i] # a[i] &
            carr_ch[i] # b[i] & carr_ch[i];
END GENERATE;
    cout = carr_ch[width+1];
END;
```

Conditional Logic Generation

When building your own parameterized functions, you can use the If Generate Statement in an AHDL file to select the function that is optimized for the device family in your project. Altera-provided parameterized functions contain similar statements to ensure that each function uses a device as efficiently as possible. See the following example.

```
IF DEVICE_FAMILY == "FLEX 10K" GENERATE
    my_adder10k
    .
    .
    .
END GENERATE;
    -- for efficient FLEX 10K implementation
```

Rule Checking

LPM functions offer designers a fast, efficient, and flexible design entry method. With this flexibility, designers require easy verification of the design's logical and legal parameter values. The Assert Statement can be used for automatic parameter checking during compilation and is an effective way to ensure that LPM functions and other parameterized functions are used as intended. A sample Assert Statement is shown below.

```
ASSERT  (width < 1)
    REPORT "LPM Width less than 1 detected"
    SEVERITY   ERROR;
```

The structure of the Assert Statement is shown below.

ASSERT( )	The condition is listed inside the parentheses. Any arithmetic expression can be used. However, a separate Assert Statement must be used for each condition to be checked.
REPORT	A user-created message is placed within quotation marks. If the assertion condition is true, this message is displayed in the MAX+PLUS II Message Processor and the Report File (**.rpt**).
SEVERITY	Possible values are: INFO, WARNING, or ERROR. The severity type is displayed immediately before the REPORT message in the Message Processor. If the value of SEVERITY is ERROR, compilation stops when the Assert Statement is true.

For more information on AHDL, refer to MAX+PLUS II Help or the ***MAX+PLUS II AHDL*** manual.

AHDL Design Concurrency

In most programming languages, the command sequence affects the functionality of the system. In contrast, the placement of symbols in a schematic or devices on a printed circuit board (PCB) does not change the logical functionality of the system. Like schematics or PCBs, all statements in the Altera Hardware Description Language (AHDL) are evaluated concurrently. This article highlights the effect of concurrency on a design, using the following AHDL Text Design File (**.tdf**) as an example.

```
SUBDESIGN example
(
   reset, load      : INPUT;
   c[2..0]          : OUTPUT;
)

BEGIN
   c[] = 4;         -- Statement 1
   IF load THEN     -- Statement 2
      c[] = 1;
   END IF;
   IF reset THEN    -- Statement 3
      c[] = 0;
   END IF;
   c[] = 2;         -- Statement 4
END;
```

In the design above, `c[2..0]` has the following values:

load	reset	Result
0	0	c[2..0] = 6
1	0	c[2..0] = 7
1	0	c[2..0] = 6
1	1	c[2..0] = 7

These values can be surprising to some designers because the AHDL design does not provide the same results as a sequential programming language (such as C++). Instead, when an AHDL TDF is synthesized by MAX+PLUS II, all statements are evaluated concurrently (for conditional statements, such as Case and If Then Statements, only the first true statement is evaluated concurrently). Equations that assign multiple values to the same AHDL node or variable are logically connected (ORed if the node or variable is active high, ANDed if it is active low). Therefore, when MAX+PLUS II evaluates the four statements in the design above during compilation, the order of statements is irrelevant.

In the preceeding example, when `load` = 0 and `reset` = 0, `c[]` has two assignments: 4 (statement 1) and 2 (statement 4). These assignments are combined and the value of `c[]` is 6 (110_2). When `load` = 1, the first If Then Statement (statement 2) adds the assignment `c[]` = 1 (001_2) to the assignments, causing `c[]` to have the value 7 (111_2). When `reset` = 1, the second If Then Statement (statement 3) assigns `c[]` to 0 (000_2). Because this assignment has no bits set to 1, the assignment has no affect on `c[]`. MAX+PLUS II considers all these possibilities and synthesizes the appropriate logic. These results are summarized in the table below.

load	reset	Result	Comment
0	0	c[2..0] = 6	Assignments made by statements 1 and 4
1	0	c[2..0] = 7	Assignments made by statements 1, 2, and 4
1	0	c[2..0] = 6	Assignments made by statements 1, 3, and 4
1	1	c[2..0] = 7	Assignments made by statements 1, 2, 3, and 4

To ensure the desired results during compilation, Altera recommends placing all assignments for a signal in a single conditional structure. For example, to obtain the following result, only a single If Then Statement is needed.

load	reset	Result
1	0 or 1	c[2..0] = 1
0	1	c[2..0] = 0
0	0	c[2..0] = 6

An AHDL TDF that implements these values is shown below.

```
SUBDESIGN example
(
   reset, load   :   INPUT;
   c[2..0]       :   OUTPUT;
)

BEGIN
   IF load THEN
      c[] = 1;
   ELSIF reset THEN
      c[] = 0;
   ELSE
      c[] = 6;
   END IF;
END;
```

MAX 9000 Programming Times

When you structure the manufacturing flow to include in-system programmability (ISP), device programming times are very important. Altera MAX 9000 devices can be programmed in-system through the Joint Test Action Group (JTAG) interface in a matter of seconds. The time required to program and verify a MAX 9000 device via the JTAG interface is a function of the programming pulse width, verification pulse width, and test input clock frequency (TCK).

MAX 9000 programming times can be calculated with the following equation:

$$t_{PROG} = (p_{PULSE} \times t_{WIDTH}) + (TCK_{CYCLES} \times 1/f_{TCK})$$

where:

t_{PROG} = Total programming time

p_{PULSE} = Number of programming pulses required

t_{WIDTH} = Programming pulse width

TCK_{CYCLES} = Number of clock cycles on TCK

The table below provides the values of these parameters for each MAX 9000 device.

Device	p_{PULSE}	t_{WIDTH} (ms)	TCK_{CYCLES}	t_{PROG} (seconds) *Note (1)*
EPM9560	354	10	760,000	3.6
EPM9480	348	10	700,000	3.6
EPM9400	342	10	640,000	3.5
EPM9320	336	10	580,000	3.4

Note:

(1) Programming times assume TCK = 10 MHz.

MAX 9000 verification times can be calculated with the following equation:

$$t_{VERIFY} = (v_{PULSE} \times t_{WIDTH}) + (TCK_{CYCLES} \times 1/f_{tck})$$

where:

t_{VERIFY} = Total verification time

v_{PULSE} = Number of verification pulses

t_{WIDTH} = Verification pulse width

The table below provides the values for these parameters for each MAX 9000 device.

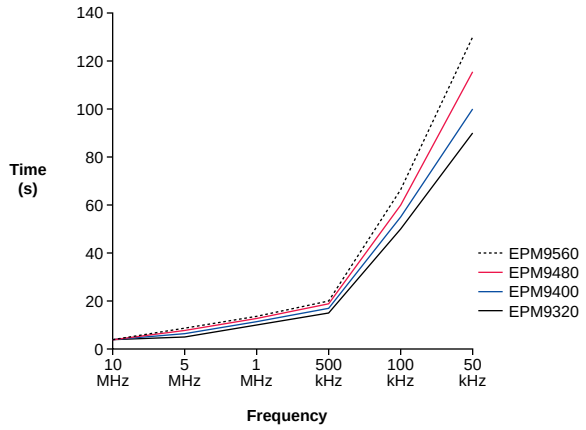
Device	v_{PULSE}	t_{WIDTH} (µs)	TCK_{CYCLES}	t_{VERIFY} (seconds) *Note (1)*
EPM9560	6018	15	2,800,000	0.37
EPM9480	6012	15	2,500,000	0.34
EPM9400	6006	15	2,200,000	0.31
EPM9320	6000	15	1,900,000	0.28

Note:

(1) Programming times assume TCK = 10 MHz.

The figure below shows the combined programming and verification times for each MAX 9000 device as a function of the JTAG test input clock frequency.

Programming & Verification Times for MAX 9000 Devices



You can use the programming options in the MAX+PLUS II Programmer to adjust the verfication times for MAX 9000 devices. For more information on these options, refer to either "Using the JTAG Interface for Multi-Device Programming" on page 14 or MAX+PLUS II Help.

Look for the 1996 Data Book in July

Now Available: First AMPP Functions

The Altera Megafunction Partners Program (AMPP) is a partnership between Altera and third-party megafunction developers. Designers can use megafunctions to shorten design cycles and speed time-to-market.

The first megafunctions developed under AMPP are the V6502 from VAutomation and a 32-bit reconfigurable finite impulse response (FIR) filter from Integrated Silicon Systems, Ltd. (ISS). VAutomation has re-worked the architecture of the V6502 for low gate usage and high performance. The 32-bit FIR filter from ISS processes 12-bit data at 171 KHz. The coefficients for the filter are stored in the FLEX 10K embedded array block (EAB), enabling the taps to be reconfigured on-the-fly to adapt the filter to changing system requirements.

The following table summarizes the current AMPP partners, their product specialties, and contact information. For the latest megafunction listings and availability information, go to Altera's world-wide web site at **http://www.altera.com**. A catalog of AMPP megafunctions will be available at DAC in early June 1996.

AMPP Partners

Partner	Specialty	Initial Product	Telephone	E-Mail
3Soft	Microperipherals	8051, PCMCIA	(408) 451-5700	sales@3soft.com
Advancel	ATM		(408) 453-0600	info@advancel.com
CAST	Microperipherals, 8-bit controllers	8254, 68450	(914) 354-4945	info@cast-inc.com
Digital Design & Development	XMIDI digital music	XMIDI controller	(32) 2 270-27-97	73261.530@compuserve.com
Eureka Technology	PowerPC, PCI interfaces	PowerPC bus	(408) 888-0439	info@eurekatech.com
Excellent Design	Microperipherals, graphics, compression	Z80 suite	(81) 45-474-9410	ampp@exd.co.jp
Infinite Solutions	Optimized DSP core	Greencore DSP	(408) 986-1686	info@InfiniteSolutions.com
Integrated Silicon Systems	DSP products	32-bit FIR filter	(44) 232-664664	info@iss-dsp.com
Logic Innovations	Set-top technologies, ATM, SONET, PCI	ATM, PCI	(619) 455-7200	info@logici.com
Object Oriented Hardware	Broad library of functions	Reed Solomon CODEC, Linked List Access Controller (LILAC)	(44) 1171-538-4114	info@ooh.com
Sierra Research & Technology	Processors, Ethernet, ATM	6805	(818) 991-1509	cores@srti.com
Silicon Engineering	Graphics controllers, CPUs	VGA Controller	(408) 438-5330	info@sei.com
SIS Microelectronics	Embedded Application blocks	Synchronizing FIFO	(303) 776-1667	info@sismicro.com
Synova	R3000, compression	FFT	(407) 728-8889	ampp@synova.com
VAutomation	CPUs, controllers	V6502	(603) 882-2282	sales@VAutomation.com
Virtual Chips (formerly RAVIcad)	PCI interfaces	PCI	(408) 452-1600	sales@vchips.com

Current Software Versions

The latest versions of Altera software products are shown below:

- MAX+PLUS II version 6.2
  (PC, Sun SPARCstation, HP 9000 Series 700, and IBM RISC System/6000 platforms)
- PLDshell Plus version 5.0
  (PC only)

Data I/O Programming Support

Data I/O provides programming hardware support for select Altera devices. Algorithms are supplied via Data I/O's Keep Current Express - Bulletin Board Service (KCE-BBS). Programming support for Configuration EPROM, MAX 9000, and MAX 7000 devices is shown in the following tables. A checkmark (✓) indicates that KCE-BBS support is available now. A date represents Altera's best estimate for when support will be available from Data I/O. All estimated availability dates are subject to change. KCE-BBS directs Data I/O customers with a current maintenance agreement to obtain qualified algorithms electronically from the KCE-BBS.

Configuration EPROM Support

Device	2900 Version 3.9	3900 Version 2.9	UniSite Version 5.1
EPC1213P-8	Hex File Only	Hex File Only	Hex File Only
EPC1213L-20	Hex File Only	Hex File Only	Hex File Only
EPC1064P-8	Hex File Only	Hex File Only	Hex File Only
EPC1064L-20	Hex File Only	Hex File Only	Hex File Only
EPC1064T-20	Hex File Only	Hex File Only	Hex File Only
EPC1064VL	Hex File Only	Hex File Only	Hex File Only
EPC1064VT	Hex File Only	Hex File Only	Hex File Only

The following MAX 7000 devices are supported by the 3900 version 2.9 and UniSite version 5.1.

- EPM7032L-44
- EPM7032Q-44
- EPM7032T-44
- EPM7032VL-44
- EPM7032VT-44
- EPM7064L-44
- EPM7064L-68
- EPM7064L-84
- EPM7064Q-100
- EPM7096L-68 (EPROM)
- EPM7096L-84 (EPROM)
- EPM7096Q-100 (EPROM)
- EPM7096L-68 (EEPROM)
- EPM7096L-84 (EEPROM)
- EPM7096Q-100 (EEPROM)
- EPM7128L-84
- EPM7128Q-100
- EPM7128Q-160
- EPM7128EL-84
- EPM7128EQ-100
- EPM7128EQ-160
- EPM7160L-84
- EPM7160Q-160
- EPM7160EL-84
- EPM7160EQ-100
- EPM7160EQ-160
- EPM7192G-160
- EPM7192Q-160
- EPM7192EG-160
- EPM7192EQ-160
- EPM7256G-192
- EPM7256W-208
- EPM7256M-208
- EPM7256EG-192
- EPM7256EG-160
- EPM7256ER-208

MAX 9000 Support

Device	3900 Version 2.9	UniSite Version 5.1
EPM9320LC84	✓	✓
EPM9320GC280	✓	✓
EPM9320RC208	✓	✓
EPM9400	*Note (1)*	*Note (1)*
EPM9480	*Note (1)*	*Note (1)*
EPM9560GC280	✓	✓
EPM9560RC240	✓	✓
EPM9560WC208	May 1996	May 1996
EPM9560RC304	✓	✓

Note:

(1) Data I/O does not currently support programming for this device. Contact Altera Applications for more information on availability.

Questions & Answers continued from page 23

When I compile this design, it seems to use a lot of logic cells and does not fit in the device I want to use. Do you have any suggestions?

A When you define a bit as "don't care," additional logic may be required to handle all possible states for that bit, which can use excessive logic resources. Changing the "don't care" bits to "0" can often reduce the amount of logic required for this portion of your design, sometimes up to 75%. Analyze your design to see if replacing your "don't care" bits with "0" will give you the results you require. If so, this change improves logic utilization and increases your chance of fitting your design in the selected device.

Programming Hardware Compatibility

The table below contains the latest programming adapter information. You should always use the software version shown in "Current Software Versions" on page 27 to ensure correct programing. "PLM" prefix adapters can be used only with the MPU.

Programming Hardware		
Device	**Package**	**Hardware**
FLEX 10K devices	All packages	PL-BITBLASTER
FLEX 8000 devices	All packages	PL-BITBLASTER
EPC1064, EPC1064V, EPC1213 (all FLEX 8000 devices), *Note (1)*	DIP, J-lead TQFP	PLMJ1213 PLMT1064
EPC1 (all FLEX 10K and FLEX 8000 devices), *Note (1)*	DIP J-lead	PLMJ1213 PLMJ1213
EPM9320	PGA J-lead (84-pin) RQFP (208-pin)	PLMG9000-280 PLMJ9320-84 PLMR9000-208
EPM9480, EPM9400	RQFP (208-pin) RQFP (240-pin)	PLMR9000-208 PLMR9000-240
EPM9560	PGA RQFP (208-pin) RQFP (240-pin) RQFP (304-pin)	PLMG9000-280 PLMR9000-208 PLMR9000-240 PLMR9000-304
EPM7032, EPM7032V	J-lead PQFP TQFP	PLMJ7000-44 PLMQ7000-44 PLMT7000-44
EPM7064	J-lead (68-pin) J-lead (84-pin) PQFP	PLMJ7000-68 PLMJ7000-84 PLMQ7000-100
EPM7096	J-lead (68-pin) J-lead (84-pin) PQFP	PLMJ7000-68 PLMJ7000-84 PLMQ7000-100
EPM7128, EPM7128E	J-lead (84-pin) PQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100 PLMQ7128/7160-160
EPM7128S	All packages	PL-BITBLASTER
EPM7160, EPM7160E	J-lead PQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100 PLMQ7128/7160-160
EPM7192, EPM7192E	PGA PQFP	PLMG7192-160 PLMQ7192/7256-160
EPM7256E	PGA MQFP, RQFP PQFP	PLMG7256-192 PLMR7256-208 PLMQ7192/7256-160
EPX780	J-lead	PLMJ780-84
EPX880	All packages	PL-BITBLASTER, *Note (2)*
FLASHlogic devices	All packages	PL-FLDLC, PL-BITBLASTER, *Notes (2), (3)*
MAX 5000 devices	All packages	*Note (4)*
Classic devices	All packages	*Note (4)*
EPS448	All packages	*Note (4)*

Notes to table:

(1) The hardware products for these devices are included with the FLEX Download Cable.

(2) MAX+PLUS II version 6.0 and higher provides programming support for all FLASHlogic devices via the BitBlaster. The EPX880 can only be programmed with the BitBlaster.

(3) You can use the FLASHlogic Download Cable (PL-FLDLC) with PLDshell Plus to program and configure all FLASHlogic devices, except the EPX880.

(4) Refer to the Altera ***1995 Data Book*** for device adapter information. Altera offers an adapter exchange program for 0.8-micron EPM5032, EPM5064, and EPM5130 programming adapters. See "Exchange Your MAX 5000 Programming Adapter for Free" on page 6 of this newsletter for more information.

Software Utilities

eau000.exe Overview of electronic application utilities

eau003.exe EP310 to EP330 JEDEC File converter

eau005.exe JEDPACK JEDEC File compactor

eau007.exe JEDSUM JEDEC checksum generator

eau017.exe LEF2AHDL converts A+PLUS LEF files to AHDL

eau018.exe PLD2EQN PAL/GAL/PLA file converter

eau019.exe ABEL2MAX file converter

eau020.exe PASM2TDF PALASM file converter

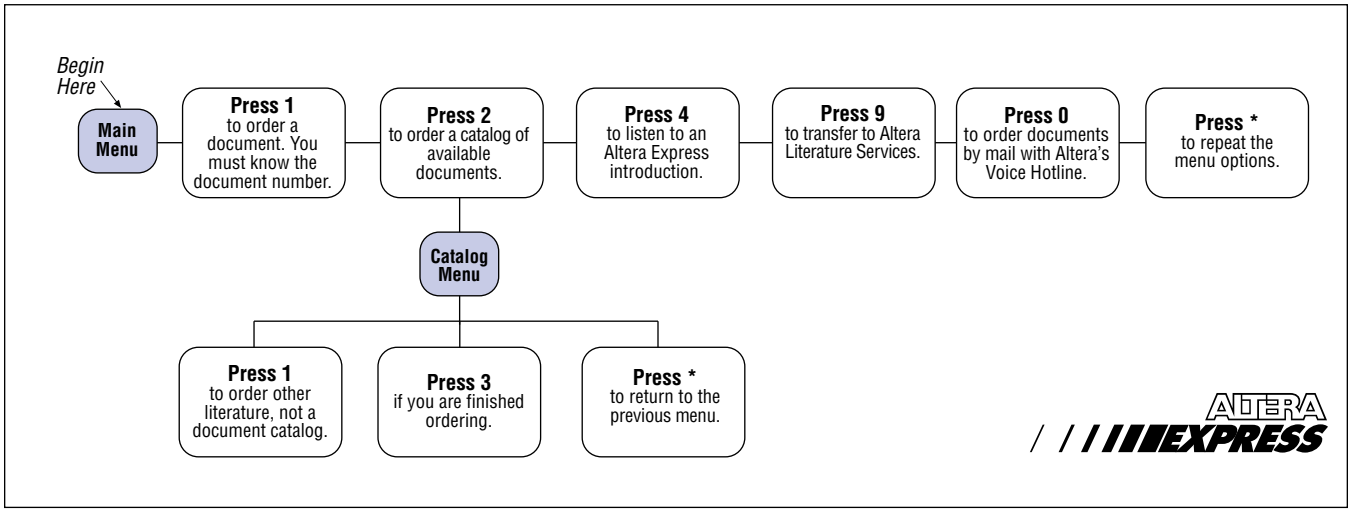
eau022.exe PLA2PDS PLA to PALASM file converter

The **ttf2rbf** file converter utility has been integrated into the MAX+PLUS II software version 6.0 and higher.

Utilities are available from the Altera BBS via modem at (408) 954-0104 and the Altera FTP site at **ftp.altera.com**.

How to Request Altera Publications

Altera publications are available through Altera Express, a 24-hour, 7-day-a-week, automated fax service. In the U.S. and Canada, call (800) 5-ALTERA; international callers can retrieve information by calling (408) 894-7850 from a fax phone. See the following figure. Documents can also be obtained from Altera Literature Services at (408) 894-7144.



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	Altera Express	(800) 5-ALTERA	(408) 894-7850
	Altera Literature Services	(408) 894-7144	(408) 894-7144 *(1)*
Non-Technical Customer Service	Telephone Hotline	(800) SOS-EPLD	(408) 894-7000
	Fax	(408) 954-8186	(408) 954-8186
Technical Support	Telephone Hotline (8 a.m. to 5 p.m. Pacific Time)	(800) 800-EPLD (408) 894-7000	(408) 894-7000 *(1)*
	Fax	(408) 954-0348	(408) 954-0348 *(1)*
	Bulletin Board System	(408) 954-0104	(408) 954-0104
	Electronic Mail	sos@altera.com	sos@altera.com
	FTP Site	ftp.altera.com	ftp.altera.com
	CompuServe	go altera	go altera
General Product Information	Telephone	(408) 894-7104	(408) 894-7104 *(1)*
	World-Wide Web	http://www.altera.com	http://www.altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative. See the Altera ***1995 Data Book*** for a list of sales offices and representatives.

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.	Current Publications	1	2	3	4	5
8.	Software Utilities	1	2	3	4	5
9.	Current Software Versions	1	2	3	4	5

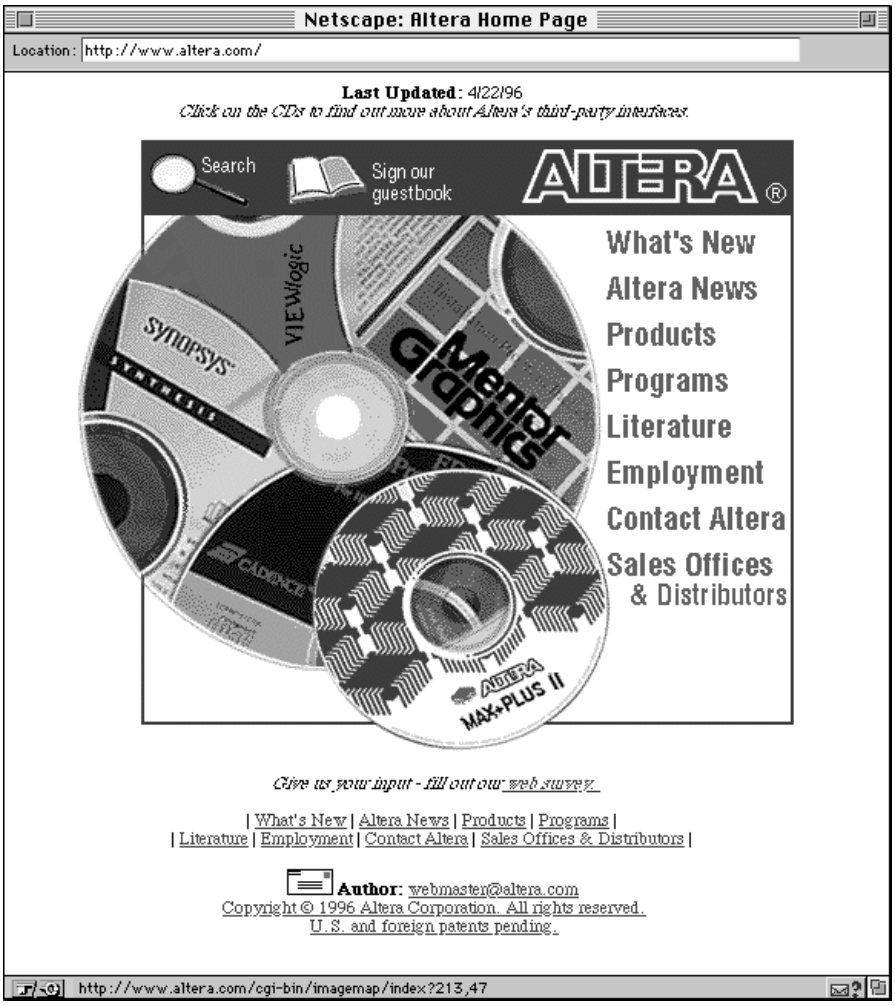
Please write your comments about *News & Views* in the space below (use additional pages if necessary). Which subjects are not getting enough coverage? What questions do you still have? What new features would you like to see?

Please fax or mail a copy of this page to:

Martin S. Won, Editor ◆ Altera Corporation ◆ 2610 Orchard Pkwy. ◆ San Jose, CA 95134-2020

Fax: (408) 954-0348 ◆ martinw@altera.com

ALTERA®



For the latest information on Altera devices and software tools, go to the Altera world-wide web site at http://www.altera.com.

The ALTERA Advantage

Altera Corporation ◆ 2610 Orchard Pkwy. ◆ San Jose, CA 95134-2020 ◆ Tel: (408) 894-7000 ◆ Fax: (408) 944-0952