The Altera Advantage News & Views

Newsletter for Altera Customers ◆ Second Quarter ◆ May 1997

Altera Announces MAX Roadmap with 3.3-V, ISP-Capable Michelangelo Family

Altera recently unveiled plans for the next-generation MAX® programmable logic device (PLD) family, code-named Michelangelo. Expected to begin shipping in 1998, Michelangelo devices will range from 32 to 1,008 macrocells, operate at 3.3-V, and offer in-system programmability (ISP).

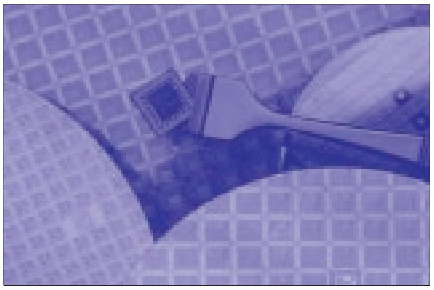
Altera plans to manufacture Michelangelo devices on a 0.35-micron, quad-layer metal EEPROM process. These devices will operate at a 3.3-V core voltage level, however, with Altera's MultiVolt™ interface, Michelangelo devices will be able to interface with 5.0-V, 3.3-V, or 2.5-V devices. A lower core voltage can improve performance and reduce power consumption by 40%.

Vertical pin migration and architectural similarities will make it easy for MAX 7000 and MAX□9000 designers to implement designs in Michelangelo devices. Altera plans to make 32-, 64-, 128-, and 256-macrocell Michelangelo devices pin compatible with the popular 5.0-V MAX 7000S family, allowing you to begin designing with 5.0-V MAX 7000S devices and later migrate to 3.3-V Michelangelo devices.

ISP Provides Flexibility

All members of the Michelangelo family will be 3.3-V, ISP-capable and will provide superior pin-locking capability. You will be able to program Michelangelo devices with automated test equipment (ATE), embedded processors, or Altera programming hardware. ISP enables in-field upgrades, reducing the need for expensive equipment rework.

MultiVolt Interface



Altera's MultiVolt™ interface will allow you to seamlessly incorporate Michelangelo devices with devices of varying voltage levels. Today's printed circuit boards are often a mix of conventional 5.0-V devices

continued on page 22

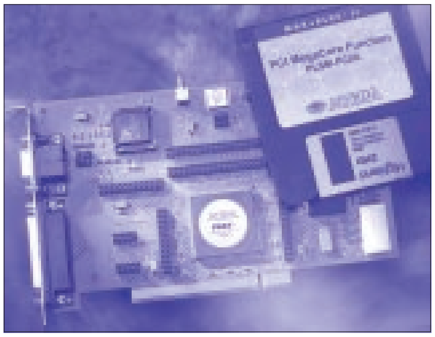
Altera Provides a Complete PCI Solution

Flexibility and density have made programmable logic devices (PLDs) an ideal choice for implementing peripheral component interconnect (PCI) subsystems. However, historically PCI interface solutions for PLDs have not been completely successful because:

- PCI specifications are difficult to meet in PLDs
- High-performance devices may not provide adequate density
- Solutions still require time-consuming development of a memory or DMA interface
- In-system testing of the PCI interface was performed late in the design cycle
- A methodology was not created to support PCI-compliant verification as designs are modified

Introducing Altera's PCI Solution

Finally, a solution exists that delivers as advertised. Altera recently shipped the PLD industry's first parameterized PCI master/target megafunction that delivers high performance and provides a complete design solution. With the `pci_a` MegaCore function, Altera offers the flexibility to meet your PCI design challenges. This zero-wait state, 33-MHz function includes a prototyping board to ease hardware



continued on page 3

Contents

Features

Altera News

Devices & Tools

Technical Articles

In Every Issue

ALTERA®

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

Erica Heidinger, Publisher
Craig Lytle, Technical Editor
2610 Orchard Parkway
San Jose, CA 95134-2020
Tel: (408) 894-7000
Fax: (408) 954-0348
E-mail: n_v@altera.com

Printed on recycled paper.

Altera, AHDL, AMPP, Atlas, BitBlaster, ByteBlaster, Classic, ClockLock, ClockBoost, FastTrack, FLASHlogic, FLEX, FLEX□10K, FLEX 10KA, FLEX□8000, FLEX 8000A, FLEX DSP, MAX 9000, MAX 9000A, MAX 7000, MAX 7000E, MAX□7000S, MAX 5000, MAX, MAX+PLUS, MAX+PLUS□II, MegaCore, MultiVolt, OpenCore, and specific device designations are trademarks and/or service marks of Altera Corporation in the United States and other countries. Altera acknowledges the trademarks of other organizations for their respective products or services mentioned in this document, specifically: SuperBGA is a registered trademark of Amkor/Anam. Verilog is a registered trademark of Cadence Design Systems. Data□I/O is a registered trademark of Data I/O Corporation. Synopsys is a registered trademark of Synopsys, Inc. Viewlogic is a registered trademark of Viewlogic Systems. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Forward-looking statements in this newsletter are made pursuant to the safe harbor provisions of the Private Securities Litigation Reform Act of 1995. Investors are cautioned that all forward-looking statements involve risks and uncertainty, including without limitation risks of dependence on third-party wafer suppliers, intellectual property rights and litigation, market acceptance of and demand for the Company's products as well as general market conditions, competition and pricing, and development of technology and manufacturing capabilities. Please refer to Altera's Securities and Exchange Commission filings, copies of which are available from Altera without charge, for further information.

NSAI
I.S. EN ISO 9001

Altera Provides a Complete PCI Solution
continued from page 1

testing of the complete system. Altera's PCI package includes:

- Parameterized configuration registers
- Prototyping board
- Software driver
- Embedded DMA engine and FIFO function
- Test vectors

The `pci_a` function has the following features:

- High data transfer rate
- Extensively tested, including hardware and simulation
- Uses FLEX® 10K embedded array blocks (EABs) for on-chip memory
- Supported by the OpenCore™ feature for instantiating and simulating designs in MAX+PLUS® II before purchase
- Compliant with requirements specified in the PCI Special Interest Group's (SIG) ***PCI Local Bus Specification***, revision 2.1 and ***Compliance Checklist***, revision□2.1

Figure 1. pci_a MegaCore Function Block Diagram

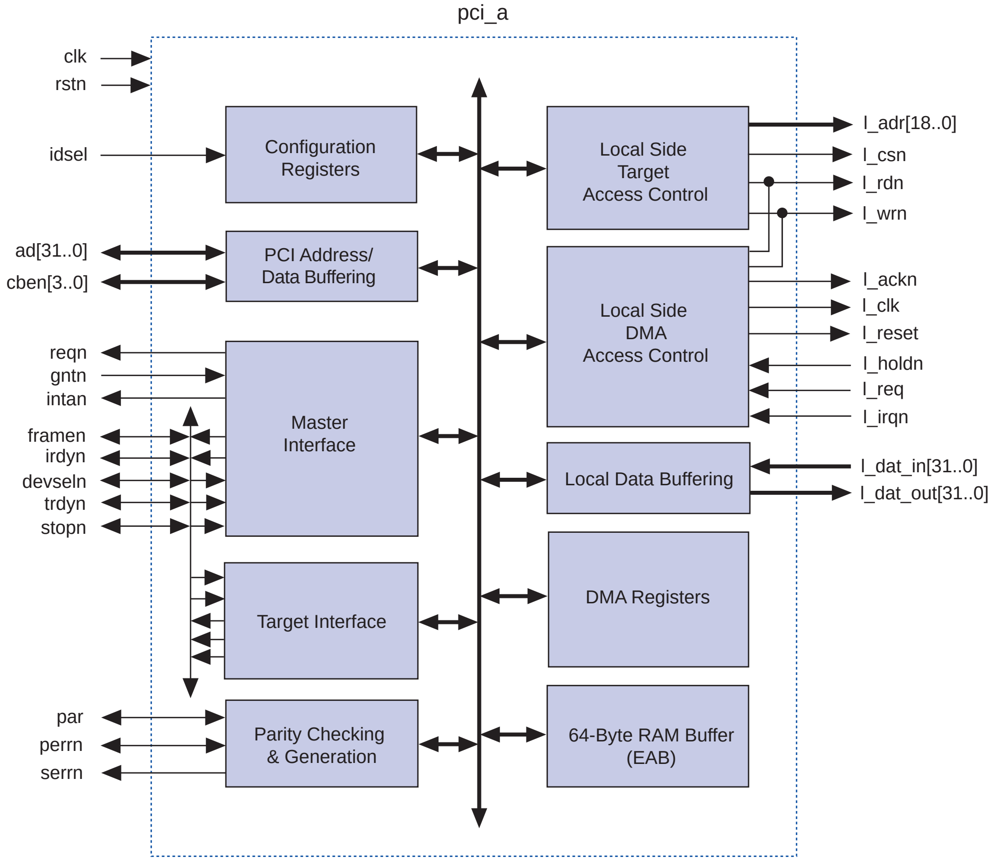


Figure 1 shows a block diagram of the `pci_a` MegaCore function. For detailed specifications, refer to "pci_a MegaCore Function" on page 11.

PCI MegaCore Function Roadmap

The "shrink-wrapped" `pci_a` MegaCore function integrates a complete memory controlling subsystem, including a high-performance DMA controller and a 64-Byte RAM buffer. This megafunction provides pre-packaged, drop-in functionality.

A second-generation PCI MegaCore function will provide the same 33-MHz, zero-wait state performance as well as a decoupled memory subsystem. This new function will give you the flexibility to use the existing DMA controller to minimize design and development effort, or design a custom memory interface to meet specific requirements of your design. Future PCI functions will provide enhanced performance and features.

Altera: The PCI Leader

The PCI master/target MegaCore function complements Altera's existing PCI portfolio, which includes many PCI functions for user-defined specifications. For example, AMPP™ partner Eureka Technology has delivered a PCI master megafunction and five different PCI target functions that provide variable numbers of base address registers and customized memory interfaces. Another AMPP partner, PLD Applications, began shipping PCI interface solutions in February 1997. AMPP partners provide the added value of customization resources.

In addition to the broadest portfolio of PCI megafunctions, Altera also provides the

continued on page 23

Devices & TOOLS

FLEX 10K

Coming Soon: EPF10K130V

Altera expects to ship the EPF10K130V, the industry's highest density PLD, in early May 1997. Altera will initially offer this device in 599-pin pin-grid array (PGA) packages. In the fourth quarter of 1997, Altera plans to follow with 600-pin ball-grid array (BGA) packages.

With 6,656 logic elements (LEs), 16 2-K□bit embedded array blocks (EABs), and 7,120 flipflops, this 130,000-gate device is the highest-density PLD in the market.

EPF10K250A Scheduled for December 1997

Altera plans to ship the EPF10K250A—containing 40,960 bits of RAM (20 EABs) and 12,160 LEs—in December 1997. Initially, EPF10K250A devices will be available in 599-pin PGA packages; 600-pin BGA packages will follow in 1998.

PCI-Compliant FLEX 10K Devices Now Available

Altera's FLEX 10K devices meet four aspects of the PCI Special Interest Group's (SIG) ***PCI Local Bus Specification***, revision 2.1:

- Timing compliance
- Electrical compliance (DC and AC)
- Protocol compliance
- Board layout (device pin-out)

EPF10K50, EPF10K40, EPF10K30, EPF10K20, and EPF10K10 devices in -3 speed grades comply with the PCI bus timing and electrical requirements (see the table below). FLEX 10K devices combined with PCI megafunctions, such as Altera® MegaCore functions or functions from the AMPP partners, comply with the PCI bus protocol. In addition, the FLEX 10K architecture provides the pinout flexibility needed to meet PCI board layout requirements. Contact your local Altera representative for more information on FLEX 10K devices and PCI compliance.

PCI Timing Requirements

PCI Symbol	PCI Parameter	Minimum PCI Specification	Maximum PCI Specification	EPF10K10-3 EPF10K20-3 EPF10K30-3 EPF10K40-3 EPF10K50-3
t_{VAL}	Clock to signal valid	2 ns	11 ns	✓
t_{ON}	Float to active delay	2 ns		✓
t_{OFF}	Active to float		28 ns	✓
t_{SU}	Input setup time	7 ns		✓
t_H	Input hold time	0 ns		✓
t_{CYCLE}	Clock cycle time	30 ns	∞	✓
t_{HIGH}	Clock high time	11 ns		✓
t_{LOW}	Clock low time	11 ns		✓
V_{CC}	Supply voltage	4.75 V	5.25 V	✓
V_{IH}	Input high voltage	2.0 V	V_{CC} + 0.5 V	✓
V_{IL}	Input low voltage	–0.5 V	0.8 V	✓
I_{IH}	Input high leakage current		70 µA	✓
I_{IL}	Input low leakage current		–70 µA	✓
V_{OH}	Output high voltage	2.4 V		✓
V_{OL}	Output low voltage		0.55 V	✓
C_{IN}	Input pin capacitance		10 pF	✓
C_{CLK}	Clock pin capacitance		12 pF	✓

FLEX 8000: Providing Low-Cost, High-Performance DSP Solutions

With the lowest prices and highest performance in the industry, FLEX 8000 devices are an ideal solution for many digital signal processing (DSP) applications. Designers have traditionally been forced to choose between the flexibility of DSP processors and the performance of DSP ASICs or application-specific standard products (ASSPs). FLEX 8000 devices simplify this decision by combining exceptional performance and low pricing with the flexibility to customize a solution for your unique system requirements. From 101 MSPS, 32-tap parallel finite impulse response (FIR) filters to 66-MHz convolutional interleavers to 77-MHz decimating filters, FLEX 8000 devices provide a fast, cost-effective way to bring your DSP solutions to market. To maximize your design productivity, pre-built DSP megafunctions—optimized for the FLEX□8000 architecture—are readily available through the Altera Megafunction Partners Program (AMPP).

MAX 9000

Altera Announces the MAX 9000A Family

Altera recently announced the MAX® 9000A family of devices, with densities ranging from 6,000 to 12,000 gates and propagation delays as fast as 7.5 ns. MAX□9000A devices will be manufactured on a 0.5-micron, triple-layer metal process, which will allow further reductions in price. The family will offer Altera's MultiVolt™ I/O interface, which allows 3.3-V or 5.0-V I/O operation on all devices. The MAX□9000A family will be pin-compatible with the MAX 9000 family.

Altera plans to ship 10-ns, 12,000-gate EPM9560A devices in August 1997 in 208- and 240-pin power quad flat pack (RQFP) packages and 356-pin BGA packages. These devices will also offer reduced power consumption. The table below summarizes the projected availability for MAX 9000A devices.

MAX 9000A Device Availability

Device	Fastest t_{PD} (ns)	Availability
EPM9320A	7.5	October 1997
EPM9400A	10	Q4 1997
EPM9480A	10	Q4 1997
EPM9560A	10	August 1997

MAX 7000

MAX 7000S Availability

The following table shows MAX 7000S device availability.

MAX 7000S Device Availability

Device	Package	Speed Grade	Availability
EPM7064S	44-pin PLCC	-7, -10	Now
	44-pin TQFP	-7, -10	Now
	100-pin TQFP	-7, -10	July 1997
EPM7128S	84-pin PLCC	-7, -10, -15	Now
	100-pin TQFP	-7 -10, -15	July 1997 Now
	100-pin PQFP	-7, -10, -15	Now
	160-pin PQFP	-7, -10, -15	Now
EPM7192S	160-pin PQFP	-10, -15	Now
EPM7256S	208-pin RQFP	-10	July 1997
	208-pin RQFP	-12, -15	Now

Now Shipping: More TQFP Packages

As designers add new and more complex features to their designs, they look to space-saving packages such as thin quad flat pack (TQFP) packages to minimize printed circuit board (PCB) size. In the past, TQFP packages were mainly used in applications where strict height restrictions were necessary, such as PCMCIA and line cards used in telecommunication switches. However, with more advanced manufacturing processes, TQFP packages have become broadly accepted as a solution for minimizing overall PCB size. Altera is now shipping MAX 7000S devices in TQFP packages with more planned introductions (see the table below).

MAX 7000S Devices in TQFP Packages

Device	Pins	Speed Grade	Availability
EPM7032S	44	-5, -7, -10	Q4 1997
EPM7064S	44	-7, -10	Now
	100	-7, -10	July 1997
EPM7096S	100	-7, -10, -15	Q4 1997
EPM7128S	100	-7 -10, -15	July 1997 Now
EPM7160S	100	-7, -10, -15	Q4 1997

MAX 5000 & Classic

Product Transitions

Altera is migrating existing MAX 5000 and Classic™ devices from a 0.8-micron process to a 0.65-micron process. Evaluation packets containing device samples and documentation are available from your local Altera sales representative. Table 1 on page 6 outlines the process migration schedule.

New Altera programming adapters are required to program the 0.65-micron MAX 5000 devices (0.65-micron Classic devices do not require new adapters). Altera will exchange existing EPM5032, EPM5064, and EPM5130 programming adapters for new adapters *for free*. These new adapters are backwards-compatible and support all existing die revisions. Table 2 on page 6 lists the existing MAX 5000 adapters that can be exchanged for new adapters. Contact your local Altera representative for more information.

continued on page 6

Devices & Tools
continued from page 5

Table 1. Product Migration Schedule

Description *(1)*	Reference *(2)*	Device	Date
MAX 5000 devices fabricated on a 0.65-micron process *Note (3)*	PCN 9407 ADV 9515 ADV 9606	EPM5032	Complete
		EPM5064	September 15, 1997
		EPM5128	Complete
		EPM5130	August 1, 1997
		EPM5192	Complete
Classic devices fabricated on a 0.65-micron process	PCN 9510 ADV 9607 ADV 9621	EP6*xx*	Complete
		EP9*xx*	Complete
		EP18*xx*	Complete

Notes:
(1) Data sheet parameters or 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.

Table 2. MAX 5000 Replacement Adapters

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

MAX+PLUS II

MAX+PLUS II Version 8.0 to Ship in June 1997

Altera continues to take a leadership position for development tools with the introduction of MAX+PLUS□II version 8.0 in June 1997. MAX+PLUS□II version 8.0 supports Viewlogic's Motive static timing analyzer and faster compilation times. MAX+PLUS□II version 8.0 also supports Altera's newest devices: EPM7064S (44-pin TQFP, 100-pin TQFP), EPM7128S (84-pin PLCC), and EPF10K130 (599-pin PGA).

Altera's software maintenance program gives you the latest features and software support for Altera's newest devices. For more information on how to purchase a software maintenance agreement, contact your local Altera sales representative.

Altera Sells 30,000th Programmable Logic Design Site

Altera has now sold 30,000 programmable logic design sites and registered approximately 13,000 designers for the ES site license. Recognized in the industry as the most efficient and user-friendly tool set, the MAX+PLUS II development system is a highly integrated package of design entry, compilation, and verification tools available on PCs and workstations.

A recent study of programmable logic users performed by *Electronic Engineering Times* showed that 50% of programmable logic users felt Altera is the leader in "quality and ease of use in design tools." Altera led in nearly every category of development tool leadership, including design tool expertise, performance in target device, low price, easiest learning curve, ease of use after learning, and PC platform support.

Available MegaCore Functions

MegaCore functions are pre-verified hardware description language (HDL) design files for complex system-level functions. These functions are developed, pre-tested, documented, and licensed by Altera as MAX+PLUS II migration products. The table below summarizes the currently available MegaCore functions. For more information, contact your local Altera representative.

Available MegaCore Functions

Function	Target Architecture	Ordering Code
Zero-wait state PCI bus	FLEX 10K	PLSM-PCI/A
Fully parameterizable fast Fourier transform (FFT) function	FLEX 10K	PLSM-FFT
Library of UART, DMA controller, and parallel port controller functions	MAX 7000 FLEX 8000	PLSM-MICROLIB
RGB-to-YCrCb color space converters	FLEX 8000	PLSM-CSC

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.

Discontinued Device Ordering Codes

Device Family	Device	Last Order Date	Last Shipment Date	Reference
FLEX 10K	EPF10K50 in -5 speed grades	3/31/97	6/30/97	ADV 9623
FLEX 8000	Selected speed grades (-6, -5, and -2A)	3/31/97	6/30/97	ADV 9622
FLASHlogic	EPX880 and EPX8160 (all packages, temperature grades, and speed grades)	6/30/97	6/30/98	PDN 9625
	EPX740 (all packages, temperature grades, and speed grades)	3/31/97	9/30/97	PDN 9516
MAX 5000	EPM5032SC-15	6/30/97	12/31/97	PDN 9624
	EPM5016 (all packages, temperature grades, and speed grades)	3/31/97	9/30/97	PDN 9516
Classic	EP220, EP224, EP312, EP324 (all packages, temperature grades, and speed grades)	3/31/97	9/30/97	PDN 9516
	Selected EP18*xx* ordering codes	3/31/97	6/30/97	ADV 9608
Function-Specific	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

New Altera Publications

New Altera publications are available from Altera Literature Services and Altera Express. Individual documents are available on the Altera world-wide web site. Document part numbers are shown in italics.

- **EPF10K130V Embedded Programmable Device Family Data Sheet Supplement** *A-DSS-EPF10K130V-2.3*
  Provides operation and configuration information for EPF10K130V devices.
- **MAX 9000 Programmable Logic Device Family Data Sheet Supplement** *A-DSS-M9000-4.1*
  Provides pin-out and package outline information for the EPM9320 and EPM9560 devices in 356-pin BGA packages.
- **MAX 7000 Programmable Logic Device Family Data Sheet Supplement**, *A-DSS-M7000S-4.2*
  Gives pin-out information for EPM7064S, EPM7096S, and EPM7160S devices in 100-pin TQFP packages.
- **Michelangelo Programmable Logic Device Family Advance Information Brief** *P-AIB-MANGELO-02*
  Provides advance device and package information for the Michelangelo family.
- **PCI Master/Target MegaCore Function with DMA Data Sheet** *A-DS-PCI1-01*
  Provides operating information for the `pci_a` MegaCore function, including target, configuration, and master transactions.
- **AN 86: Implementing the pci_a Master/Target in FLEX 10K Devices** *A-AN-086-01*
  Explains how to compile and simulate the `pci_a` function in a FLEX 10K design using the MAX+PLUS II software.
- **fft Fast Fourier Transform Data Sheet** *A-DS-FFT-02*
  Provides operating information for the `fft` MegaCore function, including ports, parameters, and memory interfaces.
- **AN 84: Implementing fft with On-Chip RAM in FLEX 10K Devices** *A-AN-084-01*
  Describes how to instantiate and simulate the `fft_on_chip` reference design in a FLEX 10K design using the MAX+PLUS II software.
- **Altera In-System Programmability CD-ROM** *M-CD-ISP-01*
  Provides current literature on the ISP feature available in MAX 9000 and MAX 7000S devices.

continued on page 15

Technical
ARTICLES

Instantiating a Parameterized Multiplier in Verilog HDL

The library of parameterized modules (LPM) offers easy implementation and quick design times to designers who use Altera devices. This article provides an example of how to infer the parameterized multiplier `lpm_mult` using Verilog HDL and Synopsys tools by adding a "black box" to your design, and then "filling" the black box with a predefined multiplier design created in the MAX+PLUS II software. Although this example shows how to infer the `lpm_mult` function using Synopsys tools and Verilog HDL, the concept applies to any function or synthesis tool.

Instantiating a parameterized function in Verilog HDL involves the following steps:

1. Create your Verilog HDL design with Synopsys tools.
2. Convert the design to an EDIF netlist file.
3. Create a MAX+PLUS II design file describing the LPM function.
4. Compile the file in the MAX+PLUS II software.

1. Create Your Verilog HDL Design with Synopsys Tools

Create your Verilog HDL design, including a reference to the LPM function. You do not need to describe the functionality of the LPM function in the design because Synopsys tools treat this module as a black box and do not translate it. However, you should define the periphery of the LPM function in Verilog HDL (i.e., the inputs and outputs).

Figure 1 shows a sample Verilog HDL design that includes a reference (`mult_a`) to the `lpm_mult` function. The Verilog HDL design registers the two groups of inputs (`a` and `b`), feeds registered signals (`reg_a` and `reg_b`) to the inputs (`in_a` and `in_b`) of `mult_a`, and then registers the output of the black box (`m_out`) to the output (`c`).

2. Convert the Design to an EDIF Netlist File

Write a script that converts the design to an EDIF netlist file (see Figure 2). This script reads in the Verilog HDL file, builds a single, flattened EDIF netlist file for the entire design, and adds the **.edf** extension (which is necessary for the MAX+PLUS II software to read it) to the EDIF netlist file. If you want to write the EDIF netlist file hierarchically, you should set a `dont_touch` attribute to the `mult_a` module. Otherwise, you will receive an error when you bring the EDIF netlist file into the MAX+PLUS II software (e.g., `node` '*xxx*' `missing source`).

3. Create a MAX+PLUS II Design File Describing the LPM Function

Before opening the EDIF netlist file in the MAX+PLUS II software, you must create a design file describing the LPM function (i.e., the contents of the black box). You can create this design as a Graphic Design File (**.gdf**), a Text Design File (**.tdf**), or a VHDL Design File (**.vhd**). In this example, the LPM function is described in a TDF.

Ensure that the names of the black box and your LPM function design in the MAX+PLUS II software are consistent, i.e., the name of the design and the ports in the MAX+PLUS II software should be the same as the name of the black box module and the ports in the HDL file. If the names are not consistent, you must create a custom Library Mapping File (**.lmf**) to map the names of the black box to the MAX+PLUS II design.

The TDF in Figure 3—written in the Altera Hardware Description Language (AHDL)—defines the function of the `mult_a` module using the LPM multiplier `lpm_mult`. An Include Statement, shown in blue text, imports the contents of the Include File containing the Function Prototype Statement of the `lpm_mult` function. An Instance Declaration, shown in red text, implements an instance of the function.

You are now ready to bring your Verilog HDL design into the MAX+PLUS II software for compilation.

4. Compile the File Using the MAX+PLUS II Software

When compiling using the MAX+PLUS II software, ensure that the design file describing the functionality of the black box and the EDIF netlist file are in your current project directory. You can use the MAX+PLUS II Compiler to compile the design and write out an EDIF Output File (**.edo**), which you can then use in Synopsys tools for retargeting. You can also

Figure 1. Verilog HDL Source Code

```
  module example(a,b,c,clk,clr);
   input [3:0] a, b;
   input clk,clr;
   output [7:0] c;
   reg [3:0] reg_a, reg_b;
   reg [7:0] c;
   wire [7:0] mult_out;

/* Instantiate the black box module*/

mult_a u1 (.clk(clk), .in_a(reg_a),
   .in_b(reg_b), .m_out(mult_out));

/* Register the inputs and outputs of
the top level file */

always @ (posedge clk or negedge clr)
if (~clr)
     begin
       c = 0;
       reg_a = 0;
       reg_b = 0;
     end
else
     begin
       c = mult_out;
       reg_a = a;
       reg_b = b;
     end
endmodule

/* Create the black box */

module mult_a(clk, in_a, in_b, m_out);
     input clk;
     input [3:0] in_a, in_b;
     output [7:0] m_out;
endmodule
```

create a Verilog Output File (**.vo**) for a post-route simulation in a third-party simulation tool.

Before synthesizing the design, you may want to perform a functional simulation. You can check the EDIF web site at **http://www.edif.org** for functional simulation modules. For more information on MAX+PLUS II and Synopsys tools, refer to the *Synopsys & MAX+PLUS II Software Interface Guide*, which is available from Altera Literature Services or on the Altera world-wide web site.

Figure 2. Synopsys Script for use with Verilog HDL Design

```
read -f Verilog HDL example.v
   uniquify
   set_dont_touch u1
   compile -map_effort medium
   write -o example.db
   write -f edif -o example.edf
exit
```

Figure 3. AHDL Design Defining the Black Box

```
INCLUDE "lpm_mult.inc";

SUBDESIGN mult_a
(
   clk, in_a[3..0], in_b[3..0]: INPUT;
   m_out[7..0]                : OUTPUT;
)

VARIABLE
   m: lpm_mult WITH (LPM_WIDTHA=4,
      LPM_WIDTHB=4, LPM_WIDTHS=4,
      LPM_WIDTHP=8, LPM_PIPELINE=1);

BEGIN
   m.clock   = clk;
   m.dataa[] = in_a[];
   m.datab[] = in_b[];
   m_out[]   = m.result[];
END;
```

Altera Express to be Discontinued

Due to the rising popularity of Altera's world-wide web site and to ensure that Altera continues to provide the best possible support for literature distribution, Altera will focus future efforts on distribution via the internet. Therefore, the Altera Express fax-back service will be discontinued on June 30, 1997. To obtain the latest Altera literature—including all literature previously available on Altera Express—visit the Altera world-wide web site at **http://www.altera.com**.

Using Multiplied & Non-Multiplied Clocks in the Same Design

Altera FLEX 10K devices have optional ClockLock™ and ClockBoost™ features that incorporate a phase-locked loop (PLL), reducing clock delay and multiplying the system clock. These features give you the ability to use multiplied and non-multiplied versions of the same clock. For example, you can use a 30-MHz clock on your circuit board, while using 30-MHz and 60-MHz clocks in your FLEX 10K design. Both clocks in the FLEX 10K device are generated by the PLL, so both will have the same low delay and skew properties. Registers driven by either the multiplied or non-multiplied clock will show a lower clock-to-output delay than registers that are driven by other clocks.

To implement the ClockLock and ClockBoost features, you should use the `clklock` function provided with the MAX+PLUS II development system. This function has a parameter called `CLOCKBOOST` that controls the clock multiplication. To double the clock frequency, set the `CLOCKBOOST` parameter to 2. To use the multiplied and non-multiplied clocks simultaneously, instantiate two `clklock` functions, both driven by the same clock pin. The `INPUT_FREQUENCY` parameter is set to the system clock frequency and must be the same for each instance. Figure 1 shows an example of this technique.

You can also implement simultaneous multiplied and non-multiplied clocks in the Altera Hardware Description Language (AHDL) by instantiating two `clklock` functions, each of which drives the appropriate registers. When using a third-party synthesis tool such as Synopsys, use the **genclk** utility (supplied with the MAX+PLUS II software) to generate two `clklock` modules, then instantiate the modules into your design.

MAX+PLUS II version 8.0 adds a new timing analysis feature: it can analyze registered performance for a system with both a multiplied and a non-multiplied clock. See Figure 2.

The critical path of the design in Figure 2 occurs between a register clocked by the non-multiplied clock and a register clocked by the multiplied clock. Normally, paths between different clock domains are ignored by registered performance analysis tools because the analysis cannot be performed without knowing the relationship between the two clocks. In a design that uses the ClockLock and ClockBoost features, the MAX+PLUS II software knows that the two clocks are synchronous to each other, with little skew. Therefore, the Timing Analyzer is able to perform the correct analysis. In Figure 2, the critical path is the highlighted 20-ns delay. The delay must be doubled because the path drives a register that is clocked by a multiplied clock; only half of the system clock cycle is available to resolve the data signal. Therefore, in this example, the system speed is 25 MHz. The Timing Analyzer also doubles the delay if the critical path occurs in a register that is clocked by a multiplied clock driving a register clocked by a non-multiplied clock.

The combination of access to multiplied and non-multiplied versions of the same clock and timing analysis of clock-multiplied systems makes it easy to incorporate ClockLock and ClockBoost clock management into your design.

Figure 1. ClockLock & ClockBoost Features Used Simultaneously

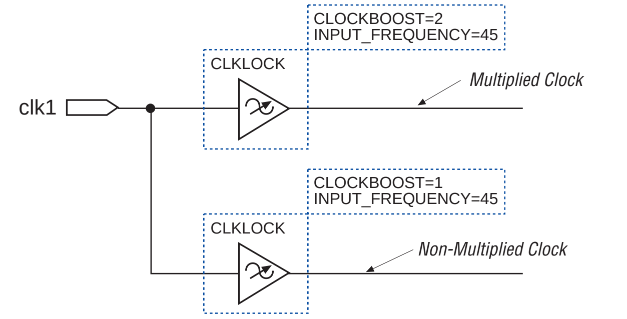
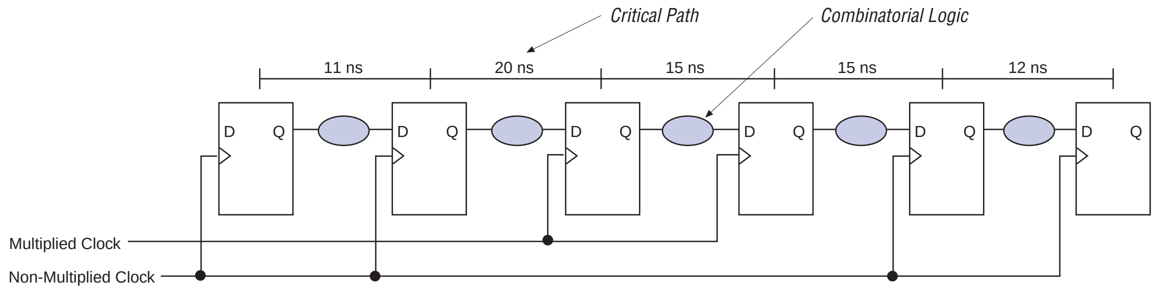


Figure 2. Critical Path Delays with Multiplied & Non-Multiplied Clocks



pci_a MegaCore Function

The PCI bus master/target MegaCore™ function (`pci_a`) offers a complete design solution that combines high performance and an integrated DMA engine with a prototyping board, software driver, test vectors, and hardware testing. The function can be easily "dropped in" to a design so that a designer can focus on integrating additional features. See Figure 1. The `pci_a` MegaCore function is optimized for the EPF10K30RC240-3 and EPF10K20RC240-3 devices. Future support is planned for FLEX 10KA devices.

The `pci_a` MegaCore function contains a DMA control engine that supports burst read and write data transfers. To transfer data on the PCI bus, the system software loads the internal DMA registers. The function is then ready to accept the local DMA request signal that enables the master to initiate data transfers on the bus.

For example, in a burst read, the master stores the read information in the RAM buffer from the PCI bus. After the burst transaction is completed, the `pci_a` MegaCore function indicates to the local side that it will transfer data from the RAM buffer to the local side memory. Similarly, in a burst write, the function indicates to the local side that it is ready to transfer data from the local side to the RAM buffer. When the RAM buffer is full, or the `pci_a` MegaCore function has the last data word, the function requests access to the PCI bus. After the arbiter grants the function access, the function will transfer all data from the RAM buffer to the PCI bus.

In the `pci_a` MegaCore function, the target capability is used for single data phase accesses. Target accesses are typically used to access configuration registers, internal DMA registers, and external target memory space.

The `pci_a` MegaCore function offers high data bandwidth and zero-wait state burst data transfers. The function can perform a zero-wait state PCI read with a bandwidth of 107 Mbytes/second and a zero-wait state PCI write at 102 Mbytes/second. It also supports a 256-byte, header type-0 configuration. Table 1 shows the key performance characteristics for the `pci_a` MegaCore function.

The `pci_a` MegaCore function uses less than 50% of the logic elements (LEs) available in an EPF10K30RC240-3 device. The remaining logic elements (LEs) can be used for user-defined local-side customization. Table 2 shows the typical device utilization for the `pci_a` MegaCore function in the EPF10K30RC240-3 device with 1,728 LEs available.

A PCI prototyping board is included with the function for implementing and testing PCI designs. The PCI prototype board contains an EPF10K30RC240-3 device that can be configured with a PCI design, a connector socket for the PCI bus interface, and other sockets for accessing the EPF10K30RC240-3 device I/O pins. The board also has 128 Kbytes of SRAM for the target address space and allows the local-side function to interface with a standard parallel or VGA port.

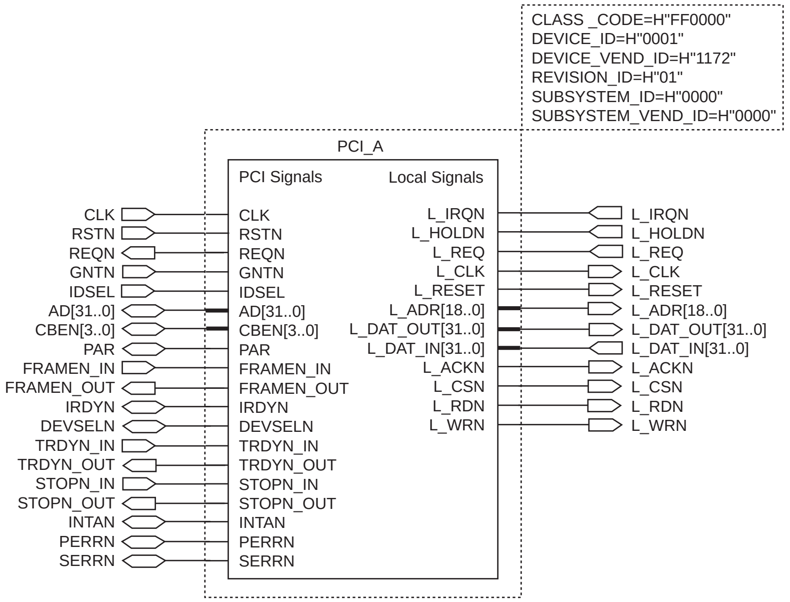
Table 1. pci_a MegaCore Function Performance Characteristics

Characteristic	Values
Clock rate	33 MHz
Read data burst transfer rate	107 Mbytes/second
Write data burst transfer rate	102 Mbytes/second

Table 2. Typical Device Utilization

Function	LEs
pci_a MegaCore function (includes a complete DMA circuit)	850
Local side with custom logic	878

Figure 1. pci_a Input & Output Pins



DSP Building Blocks in Programmable Logic

Historically, programmable logic devices (PLDs) have been used as “glue” logic to integrate and simplify board design. However, with today’s ever increasing device density, PLDs are useful for much more than just glue—they can now be used to implement complete systems, or at least major parts of a system. Altera FLEX 8000 and FLEX 10K devices are particularly well suited for implementing digital signal processing (DSP) functions because of their 4-input look-up table (LUT) architecture. These devices can be used in many DSP applications such as finite impulse response (FIR) filters, infinite impulse response (IIR) filters, digital modulators, demodulators, direct digital synthesis, video processing, and fast Fourier transform (FFT) functions. Not only do these devices occupy a small amount of board space, but they can also be much less expensive than other competing DSP processor or standard product solutions.

Basic DSP Building Blocks

Included with Altera’s MAX+PLUS II software are parameterized building blocks from the library of parameterized modules (LPM). The LPM is an architecture-independent electronic design interchange format (EDIF) standard for parameterized functions. The LPM functions that are directly applicable to DSP designs are: `lpm_mult`, `lpm_add_sub`, and `lpm_clshift`.

lpm_mult

The `lpm_mult` function is a parallel multiplier designed for high-speed operation, and has a built-in capability for constant-coefficient multiplication and signed or unsigned operation. A single function call to `lpm_mult` can implement any size multiplier. The `lpm_mult` function can efficiently multiply by a constant by taking advantage of the target architecture—a feature that is highly valuable for many DSP filtering algorithms.

lpm_add_sub

The `lpm_add_sub` function is a parameterized adder/subtractor. When compiled into FLEX 8000 and FLEX 10K devices, this function requires one logic element (LE) per bit of the adder. For example, a 16-bit adder will require 16 LEs. The smallest FLEX device currently available—the EPF8282—has 208 LEs, and the largest device currently shipping—the EPF10K100—has 4,992 LEs.

lpm_clshift

The `lpm_clshift` module is a parameterized bit-wise shifter. It can shift left or right by any number of bits and can perform arithmetic or logical shifts, or can perform a bit-wise rotation. It is useful for building DSP systems that have automatic gain control or any other system that requires run-time multiplication or division by any power of 2.

Advanced DSP Building Blocks

Using the basic building blocks, such as those discussed above, you can put together more complex DSP algorithms. One example of a complex DSP algorithm is the fast Fourier transform (FFT) MegaCore™ function. The FFT function is a fundamental building block in many DSP applications, including wireless communications, voice recognition, spectrum analysis, quadrature amplitude modulation (QAM), asymmetric digital subscriber line (ADSL), radar, and image manipulation.

The following steps explain how to instantiate and simulate the `fft_on_chip` reference design using the MAX+PLUS II development system for PCs. The `fft_on_chip` reference design implements an FFT function using the Altera `fft` MegaCore function. You must have MAX+PLUS II version 7.1 or higher in order to run this simulation.

Before Starting MAX+PLUS II

1. Create a directory named **fft** on your PC.
2. Download the self-extracting **fft02.exe** file from the Altera FTP site (**ftp.altera.com/pub/megacore/fft**), save it to your **fft** directory, and execute the file. Refer to the **readme** file on the FTP site for more information. This function may be revised or refined in the future, causing the file name to change. If you have problems getting the file, contact Altera Applications. You will receive a directory of files when you extract **fft02.exe**.
3. Copy the `fft_on_chip` Simulator Channel File, **fft_on_chip.scf,** from the **walkthru** directory to your **fft** working directory.
4. Generate the twiddle factors. In the MAX+PLUS II program directory (usually **c:\maxplus2**), type the following command at a DOS prompt:

```
twiddle 4 8↵
```

The **twiddle.exe** program, located in the MAX+PLUS□II directory, will automatically generate the file **tw4_8.mif** when you execute the above command.

5. Move the **tw4_8.mif** file to your **fft** directory. The **fft** directory should now contain the following two files: **fft_on_chip.scf** and **tw4_8.mif**.

In MAX+PLUS II

6. Choose **New** (File menu) to create a new Graphic Design File (**.gdf**) or use the **fft_test.gdf** file provided in the **walkthru** directory (if you use the file provided, you can skip steps 7 through 10).
7. Choose **Enter Symbol** (Symbol menu) and select the `fft_on_chip` symbol. Choose **Cancel** in the **Edit Ports/Parameters** dialog box; you will enter this information in a later step.
8. Enter, name, and connect the input and output pin symbols to the `fft_on_chip` symbol. For the simulation to work, the pin names must match the `fft_on_chip` symbol port names, as shown in Figure 1.
9. Save the file as **fft_test.gdf** in your **fft** directory.
10. Select the `fft_on_chip` symbol, then choose **Edit Ports/Parameters** (Symbol menu) and set the parameter values to those shown in Table 1. These are the values for the `fft_on_chip` design.

Table 1. Parameter Values

Parameter	Value
TWIDDLE_FILE	`"TW4_8.MIF"`
WIDTH_ADD	4
WIDTH_DATA	8
WIDTH_EXPONENT	4
WIDTH_TWIDDLE	8

You must use quotation marks around the name of the twiddle file.

11. Save your file and set your project to the current file by choosing **Project Set Project to Current File** (File menu).
12. Set the synthesis options for your design by choosing **Device** (Assign menu). Select *FLEX 10K* as the target device family and *AUTO* as the target device.
13. To obtain the best synthesis results, set the following options in the **Define Synthesis Style** dialog box, which is available from **Global Project Logic Synthesis** (Assign menu). Turn on the *Use LPM for AHDL Operators* option and choose *FAST* in the Style box.
14. Save and compile the design.

Simulating fft_on_chip

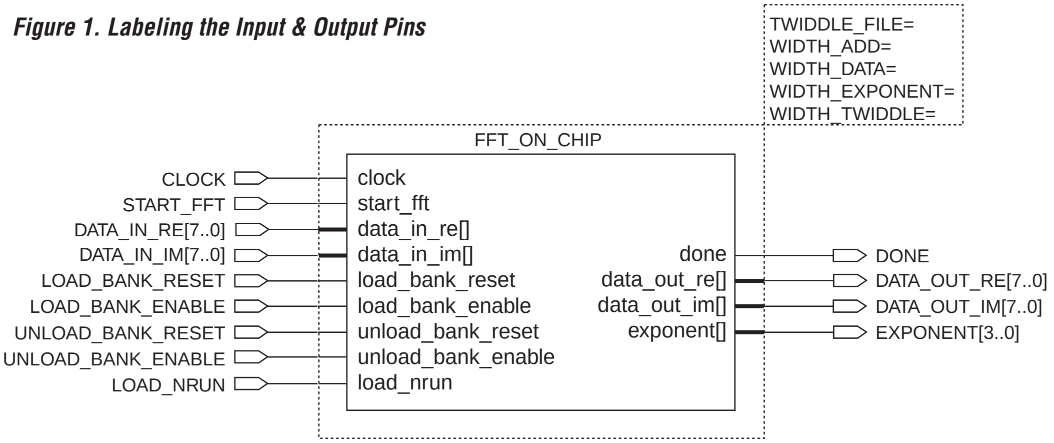
You can use the MAX+PLUS II Simulator to simulate the `fft_on_chip` reference design. You should turn on the *Check Outputs* option in the Simulator before starting simulation.

For more detailed information on how to implement the `fft` megafunction using the MAX+PLUS□II software, refer to MAX+PLUS□II Help.

Conclusion

Many common DSP functions have been successfully implemented in PLDs. The availability of pre-built functional blocks, such as the `fft` megafunction, facilitates a simple, component-like approach to design. The resulting designs are not only an order of magnitude faster than a DSP processor, but in many cases, they are also less expensive. The flexibility of programmable logic provides faster time-to-market, easier field upgrades and algorithm changes, and unparalleled performance. For more information on the `fft_on_chip` reference design, refer to *AN 84 (Implementing fft with On-Chip RAM in FLEX 10K Devices)* and *FS 7 (fft_on_chip Fast Fourier Transform)*.

Figure 1. Labeling the Input & Output Pins



Implementing a 100,000-Gate Gate Array Design in an EPF10K100

The EPF10K100 device is the largest programmable logic device (PLD) currently available. You can use the EPF10K100 to implement designs that historically have been implemented only in gate arrays (i.e., the EPF10K100 can be used for designs containing between 62,000 and 158,000 gates of logic and RAM). This article discusses the advantages of the FLEX 10K architecture for implementing a fast Fourier transform (FFT) function in an EPF10K100 device.

FFT functions are used to extract the component frequencies of a given signal. For instance, the FFT could be used to find that a given signal has component frequencies of 25 MHz, 31 MHz, and 36 MHz. Critical parameters for the FFT function are:

- Bit width, which determines the accuracy of the input and output data
- Number of points, which determines how many pieces of data are processed simultaneously

The EPF10K100 device can implement a 16-bit input/output, 256-point FFT function without using external memory. The design can operate at a clock frequency of 45 MHz (i.e., the design can perform the computation in 48 ms). This speed is comparable to dedicated off-the-shelf FFT devices. No other PLD can implement a larger FFT function. The FFT design takes advantage of the following FLEX 10K architectural features:

- The two 256 × 32 RAM blocks store data during computation.
- A 128 × 32 ROM block stores twiddle data, which is used during computation.
- The design is heavily-pipelined using FLEX logic element (LE) registers.
- Multipliers are implemented with the FLEX 10K carry chain.
- Functions from the library of parameterized modules (LPM) can be used to create efficient implementations of common functions such as adders and multipliers.
- The Altera Hardware Description Language (AHDL) gives you control over the design implementation.

Figure 1 shows a block diagram of the FFT function. The FFT function works as follows:

1. A RAM block is loaded with the data to be processed.
2. The FFT engine uses a decimation-in-frequency (DIF) FFT algorithm to compute the component frequencies of the incoming data.
3. While processing the data, the FFT function alternates reading and writing between two RAM blocks.
4. After processing is complete, the values for each component frequency are stored in the RAM, and can be unloaded into the host system.

The FFT design was compiled in the MAX+PLUS II software with an EPF10K100 device as the the target device. The same FFT design was also compiled using an LSI Logic LCA500K gate array library with industry-standard EDA tools (Synopsys Design Compiler and LSI Logic Memory Compiler). Table 1 compares the FFT design requirements for both the EPF10K100 device and LCA500K gate array. The LSI Logic LCA500K gate array required a total of 106,929 gates for the FFT design implementation.

Table 1. FFT Design Requirements for the EPF10K100 Device & LCA500K Gate Array

Design Component	EPF10K100 Device	LCA500K Gate Array
256 × 32 synchronous RAM	10 embedded array blocks (EABs)	57,936 gates
128 × 32 synchronous ROM	2 EABs	2,356 gates
Logic	3,458 LEs	46,637 gates

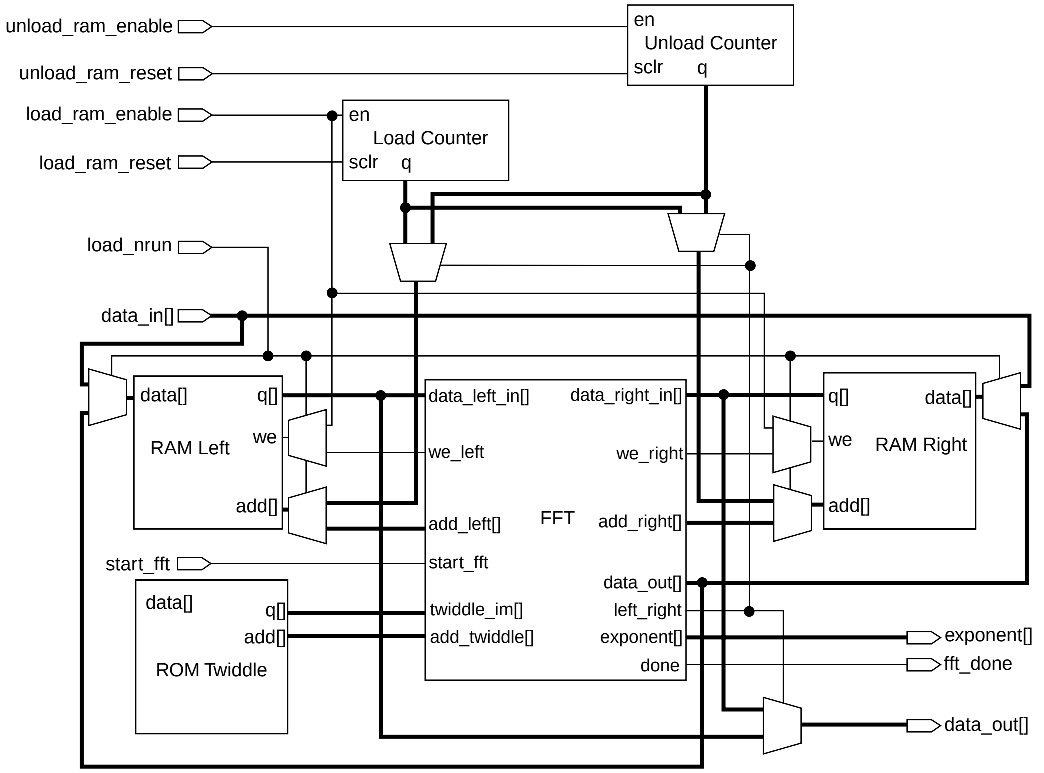
The EPF10K100 embedded array blocks (EABs) implement memory functions more efficiently than the LCA500K gate array. Implementing memory for the FFT design uses only 12 EABs, while the LCA500K uses over 60,000 gates for both RAM and ROM blocks.

To determine whether or not a particular gate array design will fit into a FLEX 10K device, you should investigate how to take advantage of the unique feature set of the FLEX 10K architecture. For example, the FLEX 10K architecture implements highly pipelined designs more efficiently than gate arrays because each FLEX logic element contains a register, while a gate array must assemble registers using NAND gates.

One final consideration is compilation time. In addition to flexibility and density, the faster compilation times available using the EPF10K100 device mean higher engineering productivity. Compilation of this FFT design for the LCA500K with Synopsys tools took one week on a Sun Ultra1 workstation. In contrast, MAX+PLUS II compilation completed in about 4 hours on a Pentium 90 PC with 64 Mbytes of RAM.

The FLEX 10K embedded array architecture provides designers with a programmable, high-density gate array alternative. For more information on implementing FFT functions, refer to the *fft Fast Fourier Transform Data Sheet* and *SB 12 (Fast Fourier Transform MegaCore Function)*.

Figure 1. FFT Block Diagram



New Altera Publications
continued from page 7

- **AN 82: Highly Optimized 2-D Convolvers in FLEX Devices** *A-AN-082-01*
  Describes how to improve the speed and efficiency of a conventional convolver by taking advantage of the numeric relationships between the coefficients.
- **AN 83: Binary Numbering Systems** *A-AN-083-01*
  Describes several binary numbering systems, including gray code, two's complement, unsigned integer, one's complement, floating point, and block floating point.
- **AN 85: In-System Programming Times for MAX 9000 & MAX 7000S Devices** *A-AN-085-01*
  Describes how to calculate in-system programming and verification times for MAX 9000 and MAX 7000S devices.
- **SB 16: Convolutional Interleaver Megafunction**
  A-SB-016-01
- **SB 18: Binary Pattern Correlator Megafunction**
  A-SB-018-01
- **SB 19: EC210 PCI Bus Master/Target Megafunction**
  A-SB-019-01
- **SB 20: PCI Bus Master/Target MegaCore Function**
  A-SB-020-01
- **SB 23: Microperipheral MegaCore Library**
  A-SB-023-01
- **SB 27: RGB2YCrCb & YCrCb2RGB Color Space Converter MegaCore Functions** *A-SB-027-01*
- **TB 14: Altera ISP-Based CPLDs & Concurrent Programming** *M-TB-014-01*
- **TB 15: Implementing a 100,000 Gate Gate Array Design in an EPF10K100 Device** *M-TB-015-01*
- **TB 19: Advantages of MAX 9000 & MAX 7000S Architectures** *M-TB-019-01*
- **TB 20: Advantages of MAX 7000S Devices**
  M-TB-020-01
- **TB 21: Benefits of Using MAX 9000 & MAX 7000S Devices** *M-TB-021-01*

Questions & ANSWERS

Q *What file contains the primitive mappings for Verilog Output Files?*

A Primitive mapping for Verilog Output Files (**.vo**) is located in the **alt_max.vo** file, which is created automatically by the MAX+PLUS II software.

Q *When I try to initialize a node in the MAX+PLUS II Simulator, the node does not retain its initial value. What should I do?*

A The MAX+PLUS II Simulator can initialize the state of your design, including input pins and registers. The value of combinatorial nodes and output pins is determined by the state of the input pins and registers. If an output pin and a register have the same name, the MAX+PLUS II software will give priority for initialization to the output pin. In this case, you can initialize the register by specifying the port as *<register name>***.q**.

Q *What should I do with the FLEX 10K and FLEX 10KA device* `DEV_CLRN` *and* `DEV_OE` *input pins during configuration?*

A During configuration, the state of the user-controlled `DEV_CLRN` and `DEV_OE` pins will not affect the operation of the device. After configuration and initialization are complete, the device will respond to these pins.

Q *Why would I use the Release Clears Before Tri-States device option?*

A The *Release Clears Before Tri-States* option affects the initialization process after the device has received all configuration data and before it has entered user mode. Two signals in the device are asserted during configuration; one tri-states all the pins (e.g., `DEV_OE`), and the other clears all the registers (e.g., `DEV_CLRn`). By default, `DEV_OE` is deasserted before `DEV_CLRn`. Therefore, for some period of time (approximately 100 to 250 ns) the output pins will drive out before the registers begin to operate. If registers directly drive pins, those pins will drive out a logic low, which may be appropriate for your design if, for example, the start-up state of the pins is critical. If you turn on the *Release Clears Before Tri-States* option, the `DEV_CLRn` signal is deasserted before `DEV_OE` is deasserted. In this case, the registers will begin to operate before the output pins are active.

Q *Why should I use dedicated clock pins instead of dedicated inputs on a FLEX 10K device?*

A On a FLEX 10K device, the two dedicated clock pins are better for distributing clocks than the four dedicated input pins for the following reasons:

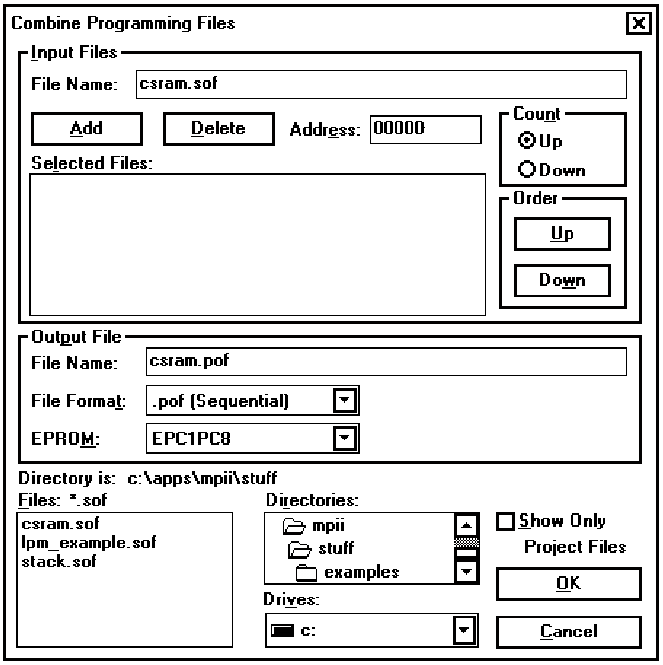
- Because dedicated clock pins cannot be used as global clears, loading on the line is reduced, which produces faster results.
- The delay from the dedicated clock pin to the logic element (LE) or I/O element (IOE) is less than the delay from the dedicated input to the LE or IOE.
- There is less skew on the dedicated clock interconnects than on the dedicated input interconnects.
- When a dedicated clock pin drives an IOE clock, the peripheral bus is not used. In contrast, when a dedicated input drives an IOE clock, one of the elements of the peripheral bus is taken, which can limit the number of output enables and clock enables that are available to the IOEs.
- Dedicated inputs can be used as clocks, but the clock-to-output times for registers fed by these clocks will be longer than the clock-to-output times of registers fed by dedicated clock pins.

The MAX+PLUS II software will show the differences between dedicated input and dedicated clock pins during simulation.

Q *How do I instruct the MAX+PLUS II software to use one EPC1 Configuration EPROM to configure an EPF81500 device?*

A When compiling for an EPF81500 device, the MAX+PLUS II Compiler automatically generates two Programmer Object Files (**.pof**) for two EPC1213 Configuration EPROM devices. If you want to use one EPC1 Configuration EPROM to configure an EPF81500 device, perform the following steps in the Programmer:

1. Choose **Combine Programming Files** (File menu). The following dialog box is displayed.



Combine Programming Files

Input Files

File Name: csram.sof

Address: 00000

Count: ● Up ○ Down

Selected Files:

Output File

File Name: csram.pof

File Format: .pof (Sequential)

EPROM: EPC1PC8

Directory is: c:\apps\mpii\stuff

Files: *.sof

csram.sof
lpm_example.sof
stack.sof

Directories:

mpii
stuff
examples

Drives: c:

□ Show Only Project Files

2. Choose the SRAM Object File (**.sof**) for your project in the *Files* box and choose **Add**.

3. Choose *.pof (Sequential)* in the *File Format* drop-down list box.

4. Choose *EPC1PC8* or *EPC1LC20* in the *EPROM* drop-down list box.

5. Enter the desired file name in the *File Name* field (*Output File* box). The default is <*project*□*name*>**.pof**. You may want to change the default name to differentiate the EPC1 POF from the EPC1213 POFs.

6. Choose **OK**.

The POF is generated and can be used to program EPC1 Configuration EPROM devices with Altera or third-party programmers.

Q *Why are some of the fields in the* ***Pin/Location/Chip*** *dialog box disabled?*

A If you have not chosen a specific device for your project and are relying on the MAX+PLUS II software to choose a device for you, you cannot make pin, location, or chip assignments, i.e., the boxes in the *Chip Resource* box in the **Pin/Location/Chip** dialog box (Assign menu) will be disabled. You can only make assignments if MAX+PLUS II knows which device and package you are using before compilation.

To select a specific device, perform the following steps in the **Pin/Location/Chip** dialog box:

1. Choose **Assign Device**.

2. Choose a target family in the *Device Family* drop-down list box.

3. Select the device you want to use in the *Devices* box.

4. Choose **OK**.

You can now adjust the settings in the *Chip Resource* box.

Q *What does* `reserved_clk_pin` *signify in the MAX+PLUS II Report File and Floorplan Editor?*

A If you use the ClockLock and ClockBoost features simultaneously, one dedicated clock pin brings the clock into the device. The other dedicated clock pin is not usable, because the interconnect that it drives is being used to distribute one of the ClockLock-generated clocks. To indicate that this pin cannot be used, the MAX+PLUS II software displays it as a `reserved_clk_pin` in the Report File and the Floorplan Editor. Your board design should drive this pin low.

For additional information on the ClockLock and ClockBoost features, refer to the *ClockLock & ClockBoost in FLEX 10K Devices Data Sheet Supplement*, which is available on Altera's world-wide web site.

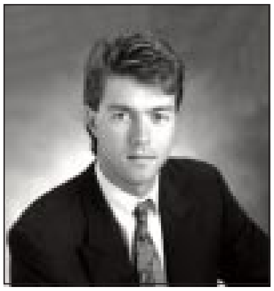
Q *Which version of the MAX+PLUS II software supports EPM7064S devices in 44-pin TQFP packages or EPM7128S devices in 100-pin TQFP packages?*

A EPM7064S devices in TQFP packages are fully supported by MAX+PLUS II version 8.0 and higher. EPM7128S devices in TQFP packages are supported by MAX+PLUS II version 7.2 and higher.

Altera

VIEWPOINT

Modern Design Methodology



Robert K. Beachler
Director, Development
Tools Marketing

Some designers believe that hardware description languages (HDLs) such as Verilog HDL and VHDL provide the highest level of electronic design. In fact, the terms "HDL-based design" and "high-level design" are often synonymous. The benefit of HDLs is increased designer productivity, and many designers feel that the transition to HDL-based design can dramatically increase productivity without any disadvantages. However, HDL-based design is just one part of the whole modern design methodology.

Programmable logic designers are in the midst of a transition from schematic and Boolean equation entry to standard HDLs, i.e., VHDL and Verilog HDL. But, designers who are using the Altera Hardware Description Language (AHDL) may be creating "high-level designs" without even knowing it. According to Altera Applications, AHDL is used by over 50% of the designers using Altera PLDs.

Although AHDL was developed prior to the use of VHDL and before Verilog HDL became an open standard, it has many features that these languages offer. However, AHDL was designed for one use only: creating textual designs for Altera PLDs. Because of this single purpose, AHDL designs are highly optimized for Altera PLDs, and allow a physical implementation to be easily specified. As a proprietary language, AHDL cannot be used for other devices and it is not supported by third-party EDA tools.

Using HDLs with Programmable Logic

VHDL and Verilog HDL and their associated tools were originally developed for gate arrays, not for programmable logic. Although any VHDL or Verilog□HDL design can be targeted and synthesized into programmable logic, the results are often far from optimal. While the overall "gate output" can be increased by targeting a gate array design for a PLD, these gates are not necessarily quality gates, i.e., they are not performance or area efficient. The architectural differences between gate arrays and programmable logic make the design transition difficult, which is why gate-array-based synthesis tools generally produce sub-optimal results for programmable logic.

Tools from Altera's ACCESS partners as well as Altera's VHDL and Verilog HDL synthesis capability provide significantly better results than gate array tools.

ACCESS™ PROGRAM

In contrast, synthesis engines designed specifically for programmable logic give better results. For example, tools from Altera's ACCESS partners—such as FPGA Express from Synopsys, Synplify from Synplicity, and Galileo and Leonardo from Exemplar Logic—as well as Altera's VHDL and Verilog HDL synthesis capability provide significantly better results than gate array tools. Even if you are already using specialized tools, area and performance optimization can still be improved using a modern design methodology. A modern design methodology uses HDLs with other entry methods and tools to provide a productive design environment that does not sacrifice area optimization or design performance.

Pre-Built Modules Optimize Results

Using pre-built modules, such as the library of parameterized modules (LPM) and the Synopsys DesignWare functions, provides the area and performance results of hand-crafted blocks, without the time spent learning programmable logic architectures. Functions such as adders, multipliers, and counters are optimally implemented using these module-generation techniques, producing better performance and area results than straight HDL implementations. The LPM is an industry standard, and EDA tool providers such as Synopsys, Exemplar Logic, Summit, and Synplicity recognize the benefits of these functions and support them in their tools. LPM and DesignWare functions can be used directly in VHDL or Verilog HDL designs, providing the benefit of optimization and the productivity of HDLs.

A Key Feature: Megafunctions

Megafunctions are another key feature in modern design methodology. Like LPM functions, megafunctions are blocks of logic that can be easily inserted into a design. The functionality of these blocks has been previously tested, so you only need to worry about system integration issues, not design creation. See Figure 1. Megafunctions from Altera, i.e., MegaCore functions, and those from Altera's AMPP partners have already been optimized for Altera devices, addressing area and performance requirements. Megafunctions range in size from small 1,000-gate functions, to some that can almost fill an entire 100,000-gate EPF10K100 device. Integrating megafunctions with your modern design environment can enhance your productivity. See Figure 2.

The Final Step: Design Verification

Design verification also plays a critical role in a modern design methodology. Behavioral simulation of HDL designs, including megafunctions and LPM functions, can help you determine the proper functionality of your design prior to synthesis and place-and-route. Behavioral simulators from Cadence, Mentor Graphics, Viewlogic, Synopsys, and Model Technology (all ACCESS partners) are a few of the tools available. Because synthesis and fitting does not alter the functionality of a design, secondary timing simulation may not be required. Instead, timing analysis using Altera's MAX+PLUS II software or other third-party software, such as Motive from Viewlogic, can be used to analyze design performance.

Conclusion

Modern design methodology encompasses much more than just HDL-based design. Module generation, megafunctions, and design verification are integral parts of today's design environment. With tools from Altera and Altera's ACCESS and AMPP partners, you can quickly create designs optimized for area efficiency and performance targetted for Altera's largest devices.

Megafunctions from Altera, i.e., MegaCore functions, and those from Altera's AMPP partners, have already been optimized for Altera devices.

Figure 1. Benefits of Using Megafunctions

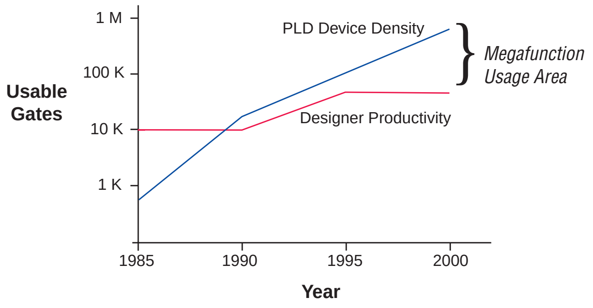
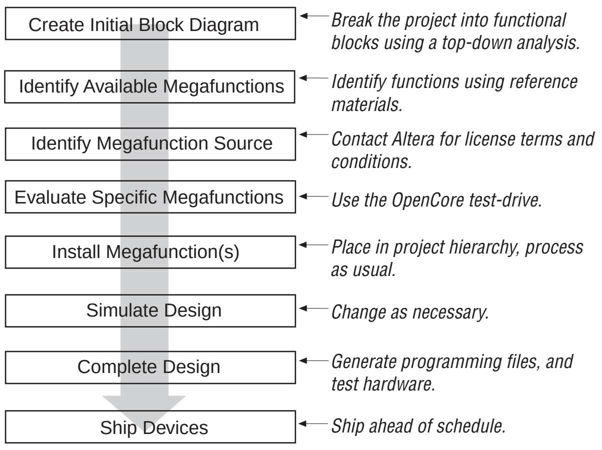


Figure 2. Megafunction Design Flow


Create Initial Block Diagram
Break the project into functional blocks using a top-down analysis.
Identify Available Megafunctions
Identify functions using reference materials.
Identify Megafunction Source
Contact Altera for license terms and conditions.
Evaluate Specific Megafunctions
Use the OpenCore test-drive.
Install Megafunction(s)
Place in project hierarchy, process as usual.
Simulate Design
Change as necessary.
Complete Design
Generate programming files, and test hardware.
Ship Devices
Ship ahead of schedule.

Implementing a $\pi/4$ DQPSK Modem with Megafunctions

Readily available, pre-tested megafunctions, such as those from the Altera Megafunction Partners Program (AMPP) provide several important advantages, including:

- System throughput is improved by implementing processing algorithms in hardware using parallelism and pipelining.
- Rapid prototyping can circumvent and complement long ASIC design cycles, improve time-to-market, and reduce technical risks.
- Megafunctions support custom designs that can be optimized for size, power, performance, and other features.
- Megafunctions can be re-used and applied to a broad range of products.

This article describes a modem application based on megafunctions available from Nova Engineering, Inc. (Nova). The transmitter portion uses finite impulse response (FIR) filters and a digital modulator. The receiver portion uses a quadrature down-converter, FIR filters, a complex multiplier, a pattern correlator, and a bit synchronizer.

Nova specializes in the design and development of leading-edge communications and signal processing systems. The company has comprehensive experience in military, government, and commercial electronics, with an emphasis on state-of-the-art digital communications systems. Many of the megafunctions described in this article have been successfully used in a variety of communication products.

The Function: $\pi/4$ DQPSK Digital Modulation

The United States and Japan have adopted a digital modulation technique—called $\pi/4$ offset, differential quadrature phase shift keying ($\pi/4$ DQPSK)—for digital cellular time division multiple access (TDMA) systems and personal communication systems (PCSs). Spectral efficiency is one of the most important requirements for PCSs and digital cellular systems. To meet this requirement, the DQPSK method transmits two bits per symbol in the same bandwidth, providing twice the bandwidth efficiency of binary phase shift keying (BPSK). In addition, DQPSK can be differentially encoded to provide the fast acquisition times necessary for burst transmissions in TDMA systems. Noncoherent receivers are preferred in the wireless communications industry because they are simple and inexpensive. The DQPSK function can be demodulated without a coherent reference at the receiver. The $\pi/4$ DQPSK offers a further advantage by allowing the use of cost- and power-efficient radio-frequency (RF) gain stages.

The Tools: AMPP Megafunctions

The $\pi/4$ DQPSK modulator and demodulator can be implemented digitally to provide a compact, low-power solution that supports a variety of network architectures. The complex logic functions necessary to implement the baseband portion of a $\pi/4$ DQPSK modem are available through AMPP. The ready-made, pre-tested megafunctions can be used with programmable logic devices (PLDs) to circumvent or complement long ASIC design cycles. Rapid prototyping with PLDs and megafunctions can accelerate time-to-market, while minimizing risk. Megafunctions can be tailored to incorporate unique features and to optimize power and performance.

The Implementation

Typical $\pi/4$ DQPSK modulator and demodulator implementations are illustrated in Figures 1 and 2, respectively. The signal mapping block converts the serial data stream into di-bit symbols, which are differentially encoded and mapped into a phase response. One of eight Cartesian pairs—$(0, \pm1)$ $(\pm1,0)$ $(\pm1/\sqrt{2}, \pm1/\sqrt{2})$—produces one of the four differential phase angles $(\pm\pi/4, \pm3\pi/4)$. Each new Cartesian pair—generated by the data stream—is passed through a square-root of raised cosine, linear phase filter, which limits the modulating signal bandwidth. The square-root raised cosine filter can be implemented using a FIR filter. Digital filters are immune to the parametric variations and degradations typically associated with analog filters.

The FIR filter output is applied to a digital modulator megafunction. The numerically controlled oscillator (NCO), contained within the digital modulator, is designed to generate the quadrature intermediate frequencies (IF). The quadrature outputs from the NCO are amplitude-modulated with the filtered, Cartesian pairs, creating the $\pi/4$ DQPSK waveform. Figure 3 illustrates the signal constellation diagram for a $\pi/4$ DQPSK signal, while Table 1 shows the instantaneous carrier phase shift generated by the di-bit symbols. The output of the digital modulator is converted to an analog signal by a digital to analog (D/A) converter. The analog signal is further filtered, amplified, and up-converted to the desired radio frequency.

The RF receiver provides the tuning, amplification, and mixing necessary to recover the signal to an intermediate frequency. Then, the A/D converter generates digital samples of the waveform for subsequent digital processing.

A digital modulator function is used in the receiver to convert the IF signal to baseband and to recover the quadrature signals. If coherent demodulation is desired, the quadrature down-converter—with a carrier-phase detector and phase locked loop (PLL)—can also track the carrier phase. The outputs of the quadrature down-converter are processed with a FIR filter that applies the same square root of raised cosine function as the filter in the transmitter.

The transmit data is differentially detected by a delay-and-mix process. The phase difference between consecutive symbols is derived by performing a complex multiplication of the quadrature values of the current symbol and the conjugate of the previous symbol. The extraction of the phase information is shown in the following calculations:

In-phase component $(t_0) = A\cos(\omega t_0 + \theta_0)$
Quadrature phase $(t_0) \quad = B\sin(\omega t0 + \theta_0)$

In-phase component $(t_1) = A\cos(\omega t_1 + \theta_1)$
Quadrature phase $(t_1) \quad = B\sin(\omega t_1 + \theta_1)$

Complex notation, assuming $A^2 + B^2 = 1$ (hard limiting) gives the following equations:

$s(t_1) = e^{j(\omega t_1 + \theta_1)}$
$s(t_0) = e^{j(\omega t_0 + \theta_0)}$

Taking the complex conjugate of $s(t_0)$ and expanding:

$s(t_1) \quad = (e^{j\omega t_1} * e^{j\theta_1})$
$s^*(t_0) = (e^{-j\omega t_0} * e^{-j\theta_0})$

Multiplication results in:

$e^{j(\theta_1 - \theta_0)} * e^{j\omega(t_1 - t_0)}$

$\omega(t_1 - t_0)$ can be assigned as $N2\pi$, making $e^{j\omega(t_1 - t_0)} = 1$. Then, $e^{j(\theta_1 - \theta_0)} * e^{j\omega(t_1 - t_0)}$ can be rewritten as:

Inphase $\quad = \cos(\theta_1 - \theta_0)$
Quadrature $= \sin(\theta_1 - \theta_0)$

This format shows that the result of the delay and complex mix is a constant, $\cos(\theta_1 - \theta_0)$ or $\sin(\theta_1 - \theta_0)$.

Figure 1. π/4 DQPSK Transmitter

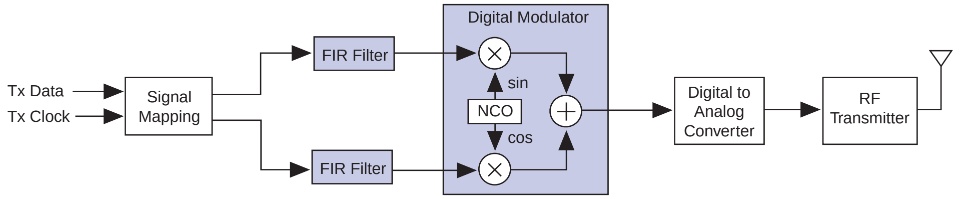


Figure 2. π/4 DQPSK Receiver

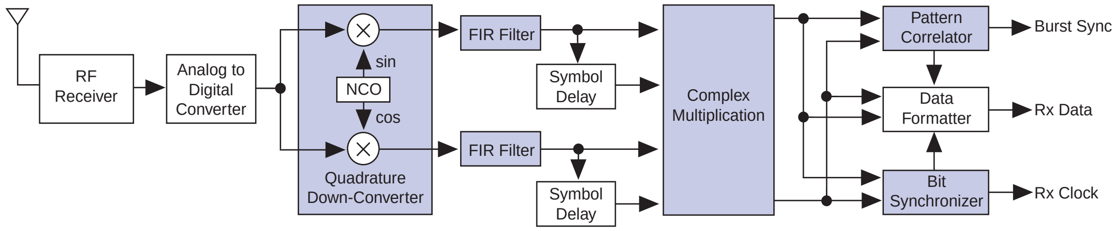


Figure 3. π/4 DQPSK Signal Constellation

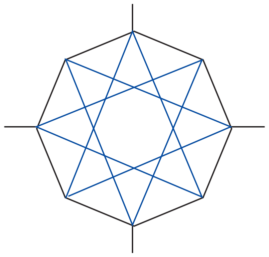


Table 1. Instantaneous Carrier Phase for π/4 DQPSK Symbols

I	Q	Change in θ
0	0	π/4
0	1	3π/4
1	0	–π/4
1	1	–3π/4

continued on page 22

Implementing a π/4 DQPSK Modem with Megafucntions continued from page 19

The constants are one of the five possible values encoded in the transmitter (i.e., $0, \pm 1, \pm 1/\sqrt{2}$) because the modulator encoding guarantees that $(\theta_1 - \theta_0) = N\pi/4$, where $N = \pm 1$ or ± 3.

A bit-synchronizer megafunction recovers timing information from the demodulated data and provides a phase lock between the local clock and the received data stream. The early/late-gate synchronizer is a closed-loop data synchronizer that performs two separate integrations for each symbol interval. The integrators accumulate the value of the symbol detected at the sample clock rising edge for each half of the symbol period. The difference between the integrations is a measure of the local clock timing error; the symbol integration for the first half matches the symbol integration of the second half when the local data clock is aligned to the incoming data. The synchronizer includes a filter for controlling the timing loop response. The filter can be programmed to increase the loop bandwidth, providing a faster transient response during acquisition mode, or to reduce the loop bandwidth, providing better performance during tracking mode.

A pattern correlator megafunction is used to identify a unique word, signifying the beginning of a burst message. The unique word should have good auto-correlation properties and is programmed into the correlator's reference pattern register. The pattern correlator's programmable threshold can be adjusted to control the probability of detection and the probability of a false alarm. Additionally, the pattern correlator can provide an initial timing estimate; if the timing reference drift between the transmitter and receiver is small over a burst period, the timing reference established by the pattern correlator may eliminate the need for a bit synchronizer.

The start of message indication and the recovered clock are used by the data formatter to format the recovered data. The data can be hard-limited by combining the most significant bits (MSB) of the recovered I and Q data streams. Alternative processing can be used by providing soft decision data—consisting of the *n* MSBs of the recovered I and Q data—to a Viterbi decoder.

References

For more information on the functions used in the π/4 DQPSK modem, refer to the following documents:

- *SB 5 (Complex Multiplier/Mixer Megafunction)*
- *SB 10 (Digital Modulator Megafunction)*
- *SB 18 (Binary Pattern Correlator Megafunction)*
- *AN 73 (Implementing FIR Filters in FLEX Devices)*

For availability information of the functions described in this article, contact Nova Engineering at:

5 Circle Freeway Drive
Cincinnati, OH 45246-1105
Tel. (513) 860-3456
Fax (513) 860-3535
info@nova-eng.com
http://www.nova-eng.com

For more information on DSP and wireless communications, refer to the following documents:

- Sklar, Bernard. ***Digital Communications: Fundamentals and Applications***. New Jersey: Prentice Hall, 1988.
- Feher, Kamilo. ***Wireless Digital Communications: Modulation and Spread Spectrum Applications***. New Jersey: Prentice Hall, 1995.
- Rappaport, Theodore S. ***Wireless Communications: Principles and Practice***. New Jersey: Prentice Hall, 1996.

Altera Announces MAX Roadmap with 3.3-V, ISP-Capable Michelangelo Family, continued from page 1

and 3.3-V devices. The central interface device—often a high-density PLD—must be able to connect with these different devices. MultiVolt-enhanced Michelangelo devices will have a core supply voltage of 3.3 V, and will have I/O pins that are compatible with 5.0-V, 3.3-V, or 2.5-V logic levels.

Altera expects the Michelangelo family to contain 32 to 1,008 macrocells in a variety of packages, including plastic J-lead chip carrier (PLCC), pin-grid array (PGA), quad flat pack (QFP), and ball-grid array (BGA) packages. Michelangelo device will be supported by Altera's industry-leading MAX+PLUS® II development system. Currently in the design phase, the first Michelangelo devices are planned for the first half of 1998. For more information on the Michelangelo family, contact your local Altera representative.

RGB2YCrCb & YCrCb2RGB Color Space Converters

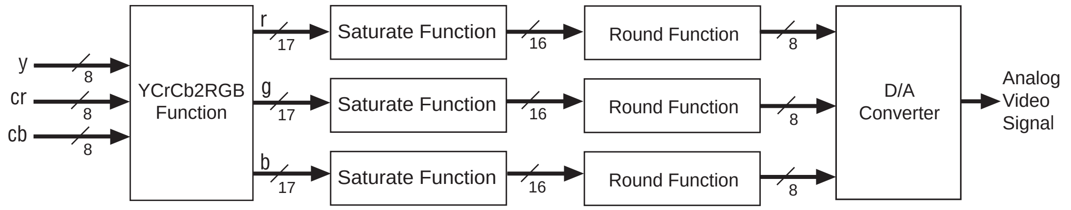
The RGB2YCrCb and YCrCb2RGB color space converter MegaCore functions convert digital video colors to television broadcast signal colors and vice-versa. The functions are useful for a number of image processing and filtering operations. The RGB2YCrCb function converts red-green-blue (RGB) color space to the YCrCb (intensity-color red-color blue) color space; the YCrCb2RGB function performs the inverse operation.

Because the inputs are multiplied by constant values, the look-up table (LUT) architecture of FLEX 10K and FLEX 8000 devices is ideal for efficiently performing the conversion equations.

Figure 1 shows a simplified illustration of an 8-bit color channel in a digital video system. In this example, color space conversion is performed using the YCrCb2RGB function. Depending on the input values to the YCrCb2RGB function, the multiplication used in the color conversion may result in 17-bit data words that roll over. To avoid data word roll over, the 17-bit digital video signals are fed to the `saturate` function, where they are saturated to 16-bit words. The signals are then fed to the `round` function, where they are rounded to 8-bit words. At this point, the data words are ready for conversion to an analog video signal.

For more information on the RGB2YCrCb and YCrCb2RGB functions, refer to the *RGB2YCrCb & YCrCb2RGB Color Space Converters Data Sheet*; for more details on data word roll over and saturating, refer to *Functional Specification 6 (saturate Data Word Saturator)*; for more information on rounding, refer to *Functional Specification 5 (round Data Word Rounder)*.

Figure 1. Digital Video Channel



Altera Provides a Complete PCI Solution continued from page 3

widest range of PCI-compliant architectures, including devices from the FLEX□10K, FLEX 8000, MAX 9000, and MAX 7000 families. You can use FLEX 8000 devices to implement cost-effective PCI target interface solutions; FLEX 10K and FLEX 10KA devices are ideal PCI master/target interfaces.

OpenCore via the World-Wide Web

Altera's OpenCore™ feature allows you to "test drive" MegaCore functions before you purchase them. With the OpenCore feature, MegaCore functions can be instantiated in your design, and then compiled and simulated using the MAX+PLUS II development system, giving you a preview of exactly how the function will fit into an Altera device. When you are ready to program a device, you must license the MegaCore function. To test-drive the PCI master/target MegaCore function using the OpenCore feature, simply download the function from Altera's world-wide web site (**http://www.altera.com**) and try it in your design.

Conclusion

The wait for a complete PCI solution is over. Altera's device architectures combined with the `pci_a` MegaCore function provide the winning combination. For more information about Altera's PCI solution, refer to the *PCI Master/Target MegaCore Function with DMA Data Sheet* or contact your local Altera sales representative.

AMPP Provides Tools for Designers

The successsful development of megafunctions requires close cooperation between the intellectual property developer and the PLD vendor. The Altera Megafunction Partners Program (AMPP) identifies megafunction developers, trains them on the Altera device architectures and tools, and promotes their resulting megafunctions. Altera works closely with AMPP partners to establish relationships with customers and to provide timely resources during megafunction evaluation and implementation. These megafunctions are the tools that designers require to increase productivity and keep pace with the increasing capacity of PLDs.

The vastly expanded product offerings described in the ***AMPP Catalog*** version 2 include information on the smallest target device that can implement a particular megafunction, as well as details on density, speed grade, and performance. Newly available AMPP megafunctions targeted for Altera devices are shown below:

- IEEE 1284 Parallel Interface
- C2910/C2910A Controller
- C49410 Controller
- 6502 Processor
- Z80 Processor
- 8031/8051 Processor
- 8032/8052 Processor
- Adaptive Filters
- Biorthogonal Wavelet Filter
- Generic UART
- Complex Mixer/Multiplier
- Convolutional Interleaver
- Decimating Filter
- Digital Modulator
- Discrete Cosine Transform
- FIR Filter Library
- Floating-Point Adder
- Floating-Point Divider
- Floating-Point Multiplier
- IIR Filter Library
- Image Processing Library
- Integer Divider
- Linear Feedback Shift Register
- Median Filter Library
- Multi-Standard ADPCM
- Numerically Controlled Oscillator
- PCI Master/Target (2 versions)
- PCI Target (1 version)
- PowerPC Bus Arbiter
- PowerPC Bus Master
- PowerPC Bus Slave
- Rank Order Filter Library
- Reed-Solomon Encoder
- Speedbridge FIFO
- Square Root Operator
- XMIDI Library
- XMIDI UART

For an AMPP information packet, call Altera Literature Services; for a copy of the ***AMPP Catalog***, contact your local Altera sales representative or go to the Altera world-wide web site at **http://www.altera.com**.

ISP CD-ROM & ISP Selector Guide Now Available

For more information on how Altera supports in-system programmability (ISP), refer to the ***In-System Programmability CD-ROM*** and the ***In-System Programmability Selector Guide***.

The Altera ***In-System Programmability CD-ROM*** contains electronic versions of all currently available literature on Altera ISP. The CD-ROM contains information on Altera ISP-capable devices, the ByteBlaster, BitBlaster, automated test equipment (ATE) vendor support, and much more. The CD-ROM also contains several technical briefs and a cost/benefit analysis illustrating the advantages of using Altera ISP-capable devices.

The Altera ***In-System Programmability Selector Guide*** can help you choose the appropriate Altera device family for your designs. This guide contains information on MAX 9000A and MAX□7000S devices, including pin and package options, speed grades, temperature grades, densities, and I/O pin offerings. The guide also offers a general overview of ISP and highlights the advantages of using ISP during development, production, and in-field upgrades.

Altera Target Applications

Altera Target Applications provide total solutions for the application-specific needs of designers. Target Applications use megafunctions from both the Altera Megafunctions Partners Program (AMPP) and the Altera MegaCore program to create integrated solutions that deliver significant time-to-market benefits. Target Applications provides technical documentation to ensure a smooth transition from design to implementation and focuses on markets such as digital signal processing (DSP), peripheral component interconnect (PCI), and wireless and broadband communications.

DSP Imaging

Altera's DSP imaging solutions provide the functional blocks necessary for high-performance DSP-based imaging systems. Combining megafunctions from the AMPP and MegaCore programs, Altera provides functional blocks for convolution, compression, and filtering applications. See Table 1.

All DSP imaging solutions employ the latest, cutting-edge technology. For example, compression support involves discrete cosine transform and JPEG megafunctions, which are ideally implemented in FLEX 10K embedded array blocks. Filtering support involves decimation and biorthogonal wavelet filters.

The new color space converter (RGB2YCrCb and YCrCb2RGB) MegaCore functions, which are available as MAX+PLUS II migration products, have full precision outputs and are optimized for the FLEX□10K and FLEX 8000 device architectures.

Target Applications CD-ROM & Selector Guide

For more information on Target Applications products, contact Altera Literature Services for a copy of the *Target Applications CD-ROM* and *Target Applications Selector Guide*. The selector guide provides you with a complete listing of megafunctions, reference designs, and technical documentation. The CD-ROM provides details about these applications, and includes reference designs and a variety of technical literature.

Table 1. DSP Imaging Functions

Function	Source
Discrete cosine transform	Integrated Silicon Systems
Image processing library	Integrated Silicon Systems
JPEG decoder	Integrated Silicon Systems
JPEG encoder	Integrated Silicon Systems
Parameterized decimator	FASTMAN
Biorthogonal wavelet filter	FASTMAN
Color-space converters	Altera MegaCore function
Video convolver	Altera reference design

Altera Supports Mixed-Voltage Systems with MultiVolt Interface

Although the 5.0-V interface has been a standard for decades, the move towards deep sub-micron architectures has required a shift to lower voltage levels. For example, a 3.3-V power supply is required for devices fabricated on 0.35-micron processes, and many of these devices require a 3.3-V interface instead of a 5.0-V interface. In the future, even lower voltage levels will be required for smaller geometry processes. In today's printed circuit boards (PCBs), designers mix 5.0-V and 3.3-V devices. The central nucleus of these systems must interface with devices of different voltages.

Altera's MultiVolt™ interface meets the demand for voltage compatibility. The MultiVolt interface separates the power supply voltage from the output voltage, enabling Altera devices powered at a specific core voltage level to interface with devices using different voltage levels. For example, Altera's FLEX 10KA devices contain patent-pending circuitry that isolates and protects the 3.3-V core from the potentially damaging 5.0-V interface levels on the I/O pins.

Altera's MAX and FLEX devices support a mixed voltage interface. Altera's 5.0-V devices—MAX 7000, MAX 9000, FLEX 8000, and FLEX 10K devices—support interfaces to 3.3-V devices. Altera's 3.3-V FLEX□10KA family supports 5.0-V, 3.3-V, and 2.5-V levels. The following table shows the core and interface voltage levels planned for Altera devices.

MultiVolt Interface Voltage Levels

Core Supply Voltage	Interface Levels Supported
5.0 V	5.0 V, 3.3 V
3.3 V	5.0 V, 3.3 V, 2.5 V
2.5 V	3.3 V, 2.5 V, 1.8 V
1.8 V	3.3 V, 1.8 V, future levels

Details of the MultiVolt interface will appear in a paper published at the 1997 Custom Integrated Circuits Conference (CICC) entitled "A 3.3-V Programmable Logic Device that Addresses Low Power Supply and Interface Trends."

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	✓	✓
EPM7032	✓	✓
EPM7064	✓	✓
EPM7096	✓	✓
EPM7128E	✓	✓
EPM7128S	✓	✓
EPM7160E	✓	✓
EPM7192E	✓	✓
EPM7192S	✓	✓
EPM7256E	✓	✓
EPM7256S	✓	✓
EPM9320	✓	*Note (3)*
EPM9400	✓	
EPM9480	✓	
EPM9560	✓	

Notes to tables:

(1) These devices are supported by Data I/O 2900 version 5.4, 3900 version 5.4, and UniSite version 5.4 programmers.

(2) These devices are supported by BP Microsystems programmers version 3.24.

(3) BP Microsystems plans to support MAX 9000 devices in the future. Contact your local your local Altera representative or BP Microsystems for more information.

Current Software Versions

The latest versions of Altera software products are shown below:

- MAX+PLUS II version 8.0, available June 1997 (PC, Sun SPARCstation, HP 9000 Series 700, and IBM□RISC System/6000 platforms)

Altera Programming Hardware Support

The following tables contain the latest programming hardware information for Altera devices. To ensure correct programming, you should always use the software version shown in “Current Software Versions” on page 26. PLM-prefix adapters can be used only with the Master Programming Unit (MPU). Table□1 shows Altera programming adapters.

Table 1. Altera Programming Adapters, *Note (1)*

Device	Package	Adapter
EPC1064, EPC1064V, EPC1213 (all FLEX 8000 devices)	DIP, J-lead TQFP	PLMJ1213 PLMT1064
EPC1 (FLEX 10K and FLEX 8000 devices)	DIP J-lead	PLMJ1213 PLMJ1213
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
EPM9560	PGA RQFP (208-pin) RQFP (240-pin) RQFP (304-pin)	PLMG9000-280 PLMR9000-208 PLMR9000-240 PLMR9000-304
MAX 7000S *(2)*	PQFP (100-pin) TQFP (100-pin)	PLMQ7000-100NC PLMT7000-100NC
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
EPM7128S, EPM7160S *(2)*	PQFP (160-pin)	PLMQ7128/160-160NC
EPM7128, EPM7128E	J-lead (84-pin) PQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100 PLMQ7128/7160-160
EPM7160, EPM7160E	J-lead PQFP (100-pin) PQFP (160-pin)	PLMJ7000-84 PLMQ7000-100 PLMQ7128/7160-160
EPM7192S, EPM7256S *(2)*	PQFP (160-pin)	PLMQ7192/256-160NC
EPM7192, EPM7192E	PGA PQFP	PLMG7192-160 PLMQ7192/7256-160
EPM7256S *(2)*	RQFP (208-pin)	PLMQ7256-208NC
EPM7256E	PGA MQFP, RQFP PQFP	PLMG7256-192 PLMR7256-208 PLMQ7192/7256-160

Table 2 provides programming information for the BitBaster™ and ByteBlaster™ download cables.

Table 2. Programming with the BitBlaster & ByteBlaster

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

Notes to tables:

(1) Refer to the Altera ***1996 Data Book*** for device adapter information for MAX 5000 and Classic devices. Altera offers an adapter exchange program for 0.8-micron EPM5032, EPM5064, and EPM5130 programming adapters. See “Product Transitions” on page 5 of this newsletter for more information.

(2) These devices are not shipped in carriers.

Software Utilities

eau000.exe	Overview of electronic 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

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

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 located in the Altera ***1996 Data Book***. Contact Altera or your local sales office for current product availability.

FLEX 10K Devices

DEVICE *(1)*	GATES	PIN/PACKAGE OPTIONS	I/O PINS	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
EPF10K10A		144-Pin TQFP, 208-Pin QFP	107, 134	I	-4			
EPF10K20	20,000	144-Pin TQFP, 208-Pin QFP, 240-Pin QFP	107, 147, 189	C	-3, -4	1,344	1,152	12,228
EPF10K30	30,000	144-Pin TQFP *(2)*, 208-Pin QFP, 240-Pin QFP, 356-Pin BGA	107, 147, 189, 246	C	-3, -4	1,968	1,728	12,228
EPF10K30A		240-Pin QFP	189	I	-4			
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 *(3)*	189, 274, 310	C	-3, -4	3,184	2,880	20,480
EPF10K50V		240-Pin QFP	189	I	-4			
EPF10K50A								
EPF10K70	70,000	240-Pin QFP, 503-Pin PGA	189, 358	C	-3, -4	4,096	3,744	18,432
EPF10K100	100,000	240-Pin QFP *(2)*, 356-Pin BGA *(2)*, 503-Pin PGA, 600-Pin BGA *(2)*	189, 274, 406, 406	C	-3, -4	5,392	4,992	24,576
EPF10K100A								
EPF10K130V	130,000	600-Pin BGA, 599-Pin PGA	470	C	-3, -4	7,120	6,656	32,768
EPF10K130A								
EPF10K250A	250,000	600-Pin BGA, 599-Pin PGA	470	C	-3, -4	12,624	12,160	40,960

Notes:

(1) Not all devices are currently available. Contact Altera for FLEX 10KA device availability.
(2) Available in FLEX 10KA devices only.
(3) Not available in FLEX 10KA devices.

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 7000 Devices

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	-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 *(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
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
EPM7096S	96	84-Pin PLCC, 100-Pin PQFP/TQFP	52, 64, 76	C	-6	6	150
EPM7096, EPM7096S	96	68-Pin PLCC *(2)*, 84-Pin PLCC, 100-Pin PQFP/TQFP *(1)*	52, 64, 76	C	-7	7.5	125
EPM7096, EPM7096S	96	68-Pin PLCC *(2)*, 84-Pin PLCC, 100-Pin PQFP/TQFP *(1)*	52, 64, 76	C, I *(1)*	-10	10	100
EPM7096	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C	-12 *(2)*	12	90.9
EPM7096	96	68-Pin PLCC, 84-Pin PLCC, 100-Pin PQFP	52, 64, 76	C, I	-15 *(2)*	15	76.9
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP/TQFP *(1)*, 160-Pin PQFP	68, 84, 100	C	-7	7.5	125
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP/TQFP *(1)*, 160-Pin PQFP	68, 84, 100	C, I *(1)*	-10(P)	10	100
EPM7128E	128	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C	-12 *(2)*	12	90.9
EPM7128E, EPM7128S	128	84-Pin PLCC, 100-Pin PQFP/TQFP *(1)*, 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 *(2)*	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 *(1)*, 160-Pin PQFP	64, 84, 104	C, I *(1)*	-10(P)	10	100
EPM7160E	160	84-Pin PLCC, 100-Pin PQFP, 160-Pin PQFP	68, 84, 100	C	-12 *(2)*	12	90.9
EPM7160E, EPM7160S	160	84-Pin PLCC, 100-Pin PQFP/TQFP *(1)*, 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 *(2)*	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 *(2)*	124	C, I *(1)*	-15	15	76.9
EPM7192E	192	160-Pin PQFP/PGA	124	C, I	-20 *(2)*	20	62.5
EPM7256S	256	208-Pin RQFP	132, 164	C	-7	7.5	125
EPM7256S	256	208-Pin RQFP	132, 164	C	-10	10	100
EPM7256E, EPM7256S	256	160-Pin PQFP, 192-Pin PGA *(2)*, 208-Pin RQFP	132, 164	C	-12(P)	12	90.9
EPM7256E, EPM7256S	256	160-Pin PQFP, 192-Pin PGA *(2)*, 208-Pin RQFP	132, 164	C, I *(1)*	-15	15	76.9
EPM7256E	256	160-Pin PQFP, 192-Pin PGA, 208-Pin RQFP	132, 164	C, I	-20 *(2)*	20	62.5

Notes:

(1) Available in MAX 7000S devices only.

(2) Not available in MAX 7000S devices.

continued on page 30

Altera Device Selection Guide

continued from page 29

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	-7, -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
EPM9400A	400	84-Pin PLCC, 208-Pin RQFP, 240-Pin RQFP	59, 139, 159	C	-10, -12
	400	84-Pin PLCC, 208-Pin RQFP, 240-Pin RQFP	59, 139, 159	C, I	-15
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, 304-Pin RQFP, 356-Pin BGA	153, 191, 216	C	-10, -12
	560	208-Pin RQFP, 240-Pin RQFP, 304-Pin RQFP, 356-Pin BGA	153, 191, 216	C,I	-15

Notes:

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

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:

Altera Contact Information

Information Type	Access	U.S. & Canada	All Other Locations
Literature *(2)*	Altera Express	*Note (1)*	*Note (1)*
	Altera Literature Services	(888) 3-ALTERA lit_req@altera.com	(408) 894-7144 *(3)* 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) 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 *(3)*
	Fax	(408) 954-0348	(408) 954-0348 *(3)*
	Bulletin Board Service	(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 *(3)*
	World-Wide Web	http://www.altera.com	http://www.altera.com

Notes:

(1) Due to the rising popularity of Altera's world-wide web site and to ensure that Altera continues to provide the best possible support for literature distribution, Altera will focus future efforts on distribution via the internet. Therefore, the Altera Express fax-back service will be discontinued on June 30, 1997. To obtain the latest Altera literature—including all literature previously available on Altera Express—visit the Altera world-wide web site at **http://www.altera.com**.

(2) Literature Services does not distribute MAX+PLUS II software manuals. Contact Altera Customer Service or your local distributor for a copy.

(3) You can also contact your local Altera sales office or sales representative. See the Altera ***1996 Data Book*** for a list of sales offices and representatives.

ALTERA®

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
2610 Orchard Parkway
San Jose, CA 95134-2020
Fax: (408) 894-7809
E-mail: n_v@altera.com