News & Views

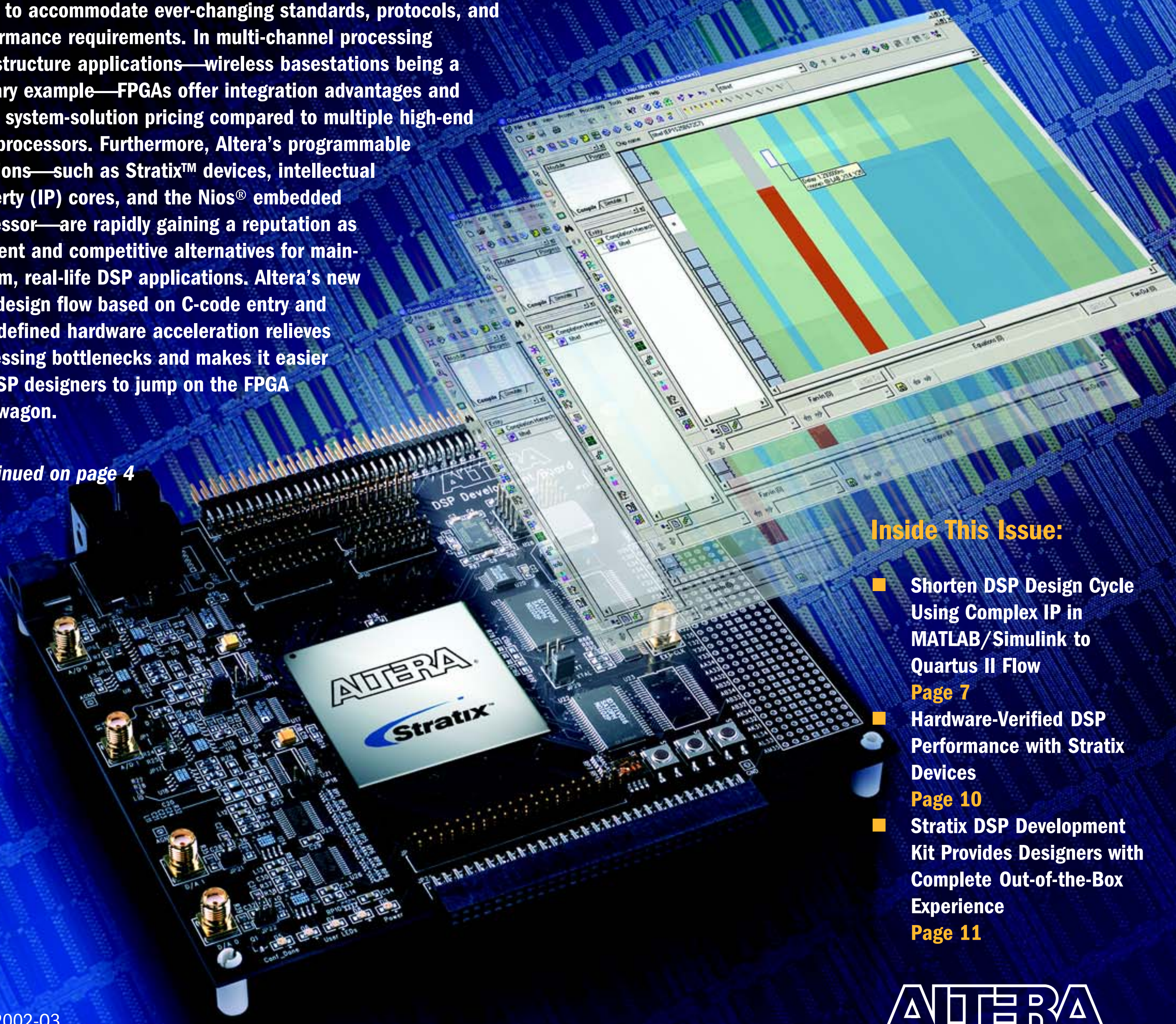
Third Quarter 2002

Newsletter for Altera Customers

DSP System Designers: Relax, It's Programmable

No longer just a novel idea, FPGAs today are a powerful solution for a broad range of digital signal processing (DSP) applications. Due to the inherent flexibility of a programmable solution, DSP system designers can be confident that their designs will scale to accommodate ever-changing standards, protocols, and performance requirements. In multi-channel processing infrastructure applications—wireless basestations being a primary example—FPGAs offer integration advantages and lower system-solution pricing compared to multiple high-end DSP processors. Furthermore, Altera's programmable solutions—such as Stratix™ devices, intellectual property (IP) cores, and the Nios® embedded processor—are rapidly gaining a reputation as efficient and competitive alternatives for mainstream, real-life DSP applications. Altera's new DSP design flow based on C-code entry and user-defined hardware acceleration relieves processing bottlenecks and makes it easier for DSP designers to jump on the FPGA bandwagon.

continued on page 4



Inside This Issue:

NV-2002-Q3

Message from *Marketing*

Programmable Logic Goes Mainstream in the DSP World

You've heard the saying: The world is analog. While that may be true in a pure sense, it is digital signal processing (DSP) that increasingly allows most technology to touch and impact our daily lives. From personal entertainment systems, to wireless communications, to remote medical diagnosis, DSP technology is driving the most significant advances in real-world applications.

FPGAs have a long history of providing hardware acceleration in DSP applications and are now breaking into the mainstream. Already at the heart of many mobile phone systems, FPGAs provide the processing power for hundreds of individual simultaneous voice and data channels. Now, FPGAs can be found providing signal processing power in a broad array of applications such as studio editing equipment, high-definition televisions, global positioning systems, medical imaging, encryption applications, and many more. The reason is simple; for many algorithm-intensive tasks, FPGAs can frequently provide better performance at a lower cost when compared to a standard DSP processor.

Meeting this increasing demand for FPGA-based DSP solutions, Altera has significantly increased its DSP arsenal, while maintaining a focus on ease-of-use. Over the last year, enhanced IP functions, improvements in our software tools, new development kits, and better interfaces to industry-standard tool flows reflect just some of our DSP-centric investments. A recent and more strategic addition is the inclusion of dedicated DSP blocks in Stratix™ devices, which are shipping to customers today. (Stratix devices are breaking new ground. In fact, customers who have Stratix devices today inform us that the DSP performance they are witnessing in the core logic and dedicated DSP blocks is higher than in any other programmable device on the market.) All these elements are designed to provide you with the competitive edge that you need to win in your markets.

In this issue, we will introduce the different DSP solutions that Altera has available today. And here is the best part for DSP system designers: you no longer have to be experts in HDL design to implement DSP functions on Altera FPGAs. Altera provides a C-code-based flow combined with custom DSP hardware acceleration designed in the familiar MATLAB/Simulink tool domain. From Stratix embedded DSP blocks to our Nios® embedded processor and intellectual property (IP) accelerator functions, Altera offers an effective, competitive, and easy-to-use alternative to fixed DSP processors with all the benefits of programmable logic. You can develop a complete solution integrating the data path with the control path on a single FPGA. There are no limits to how far you can take your application—changing protocols and standards, exploding DSP performance requirements, and ever-changing product features pose no threat to a programmable-logic-implemented design. Given all these ingredients, the possibilities for FPGAs in the DSP world are endless.

Erik Cleage,
Senior Vice President of Marketing

Table of Contents

Features

Devices & Tools

Contributed Articles

Technical Articles

In Every Issue

Altera, ACAP, ACCESS, ACEX, ACEX 1K, AMPP, APEX, APEX 20K, APEX 20KC, APEX 20KE, APEX II, Atlantic, Avalon, BitBlaster, ByteBlaster, ByteBlasterMV, Classic, ClockBoost, ClockLock, ClockShift, CoreSyn, DirectDrive, E+MAX, Excalibur, FastLUT, FastTrack, FineLine BGA, FLEX, FLEX 10K, FLEX 10KE, FLEX 10KA, FLEX 8000, FLEX 6000, FLEX 6000A, Flexible-LVDS, HardCopy, IP MegaStore, Jam, LogicLock, MasterBlaster, MAX, MAX 9000, MAX 9000A, MAX 7000, MAX 7000E, MAX 7000S, MAX 7000A, MAX 7000AE, MAX 7000B, MAX 3000, MAX 3000A, MAX+PLUS, MAX+PLUS II, MegaCore, MegaLAB, MegaWizard, Mercury, MultiCore, MultiVolt, MultiTrack, NativeLink, Nios, nSTEP, OpenCore, OptiFLEX, PowerFit, PowerGauge, Quartus, Quartus II, RapidLAB, SignalCore, SignalProbe, SignalTap, SignalTap Plus, SignalTap II, SoftMode, Stratix, Terminator, The Programmable Solutions Company, TriMatrix, True-LVDS, 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: Adobe and Acrobat are registered trademarks of Adobe Systems Incorporated. ARM is a registered trademark and ARM922T and ETM9 are trademarks of ARM limited. HP-UX is a trademark of Hewlett-Packard Company. HyperTransport is a trademark of HyperTransport Consortium. Mentor Graphics is a registered trademark and Exemplar, LeonardoSpectrum, and ModelSim are trademarks of Mentor Graphics Corporation. Microsoft, Windows, Windows 98, and Windows NT are registered trademarks of Microsoft Corporation. PALACE is a trademark of Aplus Design Technologies, Inc. RapidIO is a trademark of RapidIO Trade Association. Rochester Electronics is a registered trademark of Rochester Electronics, Inc. Sun is a registered trademark and Solaris is a trademark of Sun Microsystems, Inc. Synplicity, Synplify, Synplify Pro are registered trademarks of Synplicity, Inc. 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. The actual availability of Altera's products and features could differ from those projected in this publication and are provided solely as an estimate to the reader.

Copyright © 2002 Altera Corporation. All rights reserved.

♻ Printed on recycled paper.

I.S. EN ISO 9001

John Panattoni,
Publisher
Justin Bennett,
Technical Editor
Paula Cruz,
Cover Layout
101 Innovation Drive
San Jose, CA 95134
Tel: (408) 544-7000
Fax: (408) 544-7809
n_v@altera.com

DSP System Designers: Relax, It's Programmable, continued from page 1

DSP Processor Bottlenecks

DSP software developers have grown accustomed over the last decade to leveraging hardware acceleration techniques employed by off-the-shelf DSP processors. These techniques include adding special custom instructions and dedicated co-processors. For example, Texas Instruments (TI) TMS320C54x and Motorola Starcore SC110/140 have "custom" instructions (capable of providing up to 5× improvements in efficiency over pure software implementations) to accelerate Viterbi add-compare-select operations used in wireless forward error correction (FEC). Although this approach has proven to be quite successful for the targeted applications, it does have several critical limitations. For example, in TI's latest C6000 processor family, there are only two fixed co-processors and eight custom integrated instructions. Most of the DSP applications do not cover all the algorithms, and therefore cannot take full advantage of the vendors' predefined hardware acceleration capability. In addition, the hardware accelerators can quickly fall out of date for today's rapidly evolving standards. Another drawback is in the area of on-chip bus bandwidth. When selecting an off-the-shelf DSP processor, designers have to make compromises between size and performance that determine the number of data buses on the device. For heavily accelerated tasks, these often become the primary performance limitation.

Earlier this year at the Embedded Systems Conference, an EE Times study revealed that the Nios embedded processor is one of the fastest growing 16- or 32-bit processors in the market today.

Table 1. DSP Hardware Acceleration Availability

TI C6000	Altera	
Viterbi	Reed Solomon *(1)*	Numerically Controlled Oscillator (NCO) *(1)*
Galois Field Multiply	Viterbi	Constellation Mapper *(1)*
Turbo	Turbo	Correlator
	Cyclic Redundancy Code (CRC)	G.711
	Interleaver *(1)*	G.726
	Finite Impulse Response (FIR) *(1)*	MPEG4
	Infinite Impulse Response (IIR) *(1)*	MPEG2
	Fast Fourier Transform (FFT) *(1)*	Discrete Cosine Transform (DCT)
	IDCT	Motion JPEG
	Wavelet	Color Space Converter
	Advanced Encryption Standard (AES)	DES/3DES
	SHA-1	SHA-2

Note to Table 1:

(1) DSP Builder-Ready Certified.

Relieving the Bus Bottlenecks with Nios

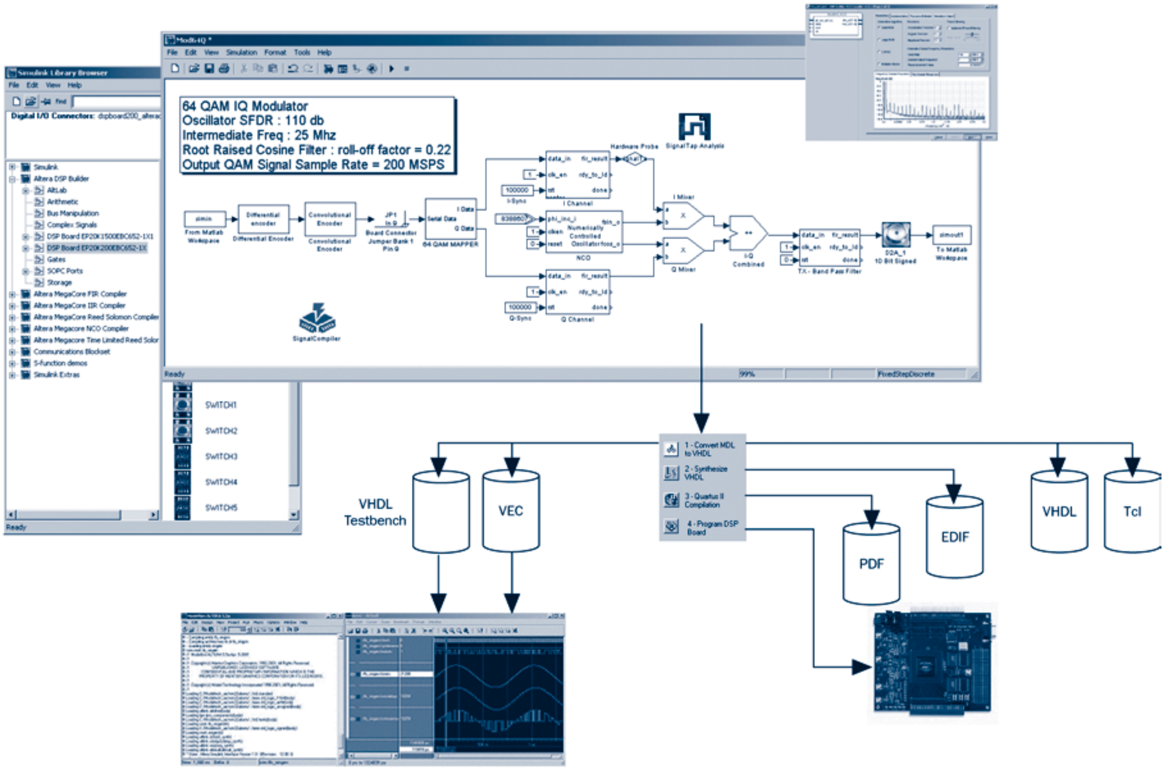
Altera's Nios embedded processor was designed to provide the maximum performance (over 125 MHz in Stratix devices) at the lowest cost by giving the user flexibility to execute instructions in hardware or software. With the Nios embedded processor, automated hardware and software compiler support for custom instructions and custom peripherals is built right into the SOPC Builder tool, delivered in the Altera® Quartus® II software version 2.1. Table 1 shows DSP hardware acceleration availability. As a key element of DSP applications, the Nios embedded processor utilizes a multi-master bus architecture that removes the bus bandwidth performance bottleneck found in DSP processors. Using multi-master buses, designers can define as many buses and as much performance as needed for a particular application.

Earlier this year at the Embedded Systems Conference, an EE Times study revealed that the Nios embedded processor is one of the fastest growing 16- or 32-bit processors in the market today. In fact, since the product was released just over a year ago, Altera has recorded over 6,000 Nios embedded processor users.

C-Based DSP Design Flow

Altera provides the broadest and most cost-effective solutions for a wide variety of DSP applications by leveraging Stratix devices, implementing the highly successful Nios embedded processor, and including the Altera suite of parameterizable DSP IP. Designers can accomplish this flow using traditional C-code-based development; therefore, software engineers can design their DSP systems without becoming HDL experts. First, the software developer profiles the algorithm in C and identifies those functions which consume the highest percentage of processor cycles. Then DSP Builder version 2.0, a tool that leverages the capabilities of MATLAB/Simulink and Altera DSP IP, allows developers to verify their system design in Simulink and go directly to a logic implementation (see Figure 1). Integration of the DSP IP and custom instructions, along with the processor, buses, memory interfaces, and communications peripherals, is seamlessly handled by SOPC Builder. The control code, along with any DSP code that is left, can then run efficiently on the Nios processor. The Nios processor combined with hardware acceleration offers the ultimate

Figure 1. DSP Builder Design Flow



flexibility, performance, and cost effectiveness in a development flow that is familiar to software developers (see Figure 2).

Stratix Dedicated DSP Blocks

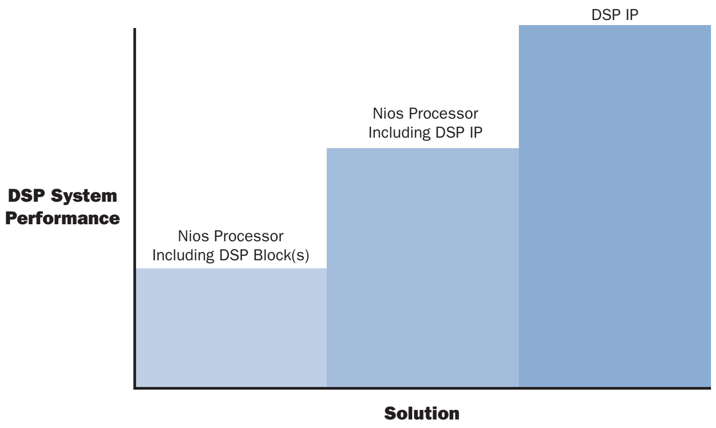
Flexible, efficient, and optimized for a variety of applications that require high data throughput, the DSP blocks in Stratix devices contain a variety of typical DSP functions. These blocks, available on every member of the Stratix device family, consist of hardware multipliers, adders, subtractors, accumulators, and pipeline registers (see *AN 214: Using the DSP Blocks in Stratix Devices*). Stratix DSP blocks can run at 250 MHz to provide data throughput performance of up to 2.0 giga multiply accumulate operations per second (GMACs) per DSP block. Because designers can implement Stratix DSP blocks with dedicated circuitry, they offer optimal performance.

Advantages of Altera's MegaCore Functions

DSP IP is ideal for emerging applications such as third-generation (3G) wireless, digital audio and video imaging processing and broadcast, multichannel multi-point distribution services

continued on page 6

Figure 2. DSP Solutions



DSP System Designers: Relax, It's Programmable, continued from page 5

(MMDS), and orthogonal frequency division multiplexing (OFDM) systems. The flexibility of programmable logic and soft IP cores enables companies to quickly adapt their designs to new standards such as the Wireless 802.11a, Wireless Broadband Working Group 802.16, and HiperLAN/2.

IP cores are statically parameterized so that the most efficient hardware is produced for a given set of parameters, using the MegaWizard Plug-In Manager.

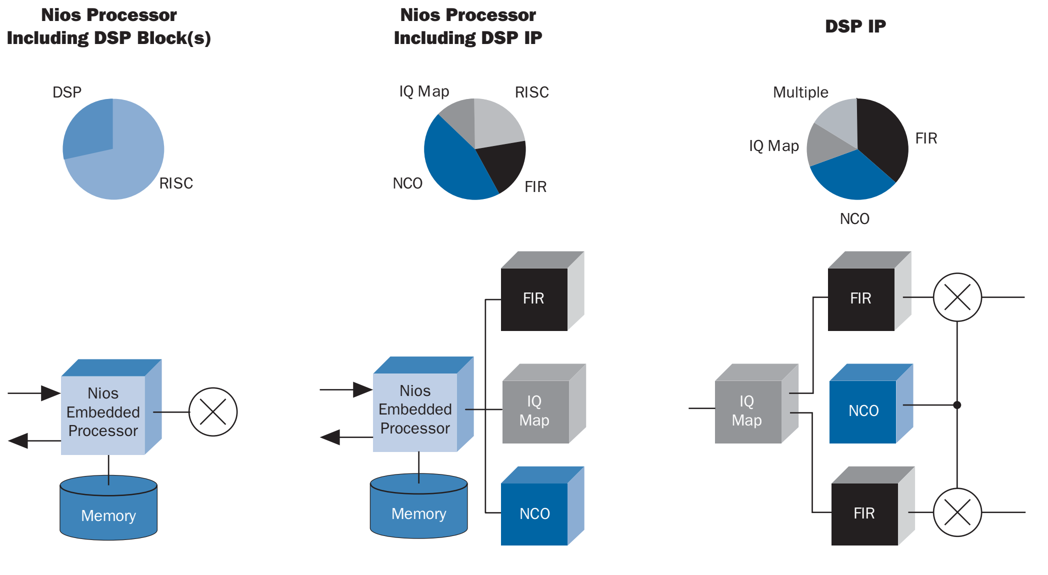
Altera's suite of DSP IP, which includes standard DSP functions (i.e., Viterbi and turbo decoders), offers several significant advantages. IP cores are statically parameterized so that the most efficient hardware is produced for a given set of parameters, using the MegaWizard® Plug-In Manager. These plug-ins provide maximum flexibility, allowing users to customize IP without changing design source code. If parameters change, Altera can update the IP in a matter of days or weeks, as opposed to DSP processors that may take years to incorporate updated versions. Also, soft IP can immediately be ported to new Altera FPGA device families providing an obsolescence-proof solution with cost and performance advantages. Design and manufacturing departments now have a solution they can both agree on.

Conclusion

Both hardware and software designers now have easy access to a variety of Altera solutions for DSP applications, complete with all the benefits of programmable logic. With the advent of new FPGA architectures such as the Stratix device family and the growing number of DSP IP cores and tools such as DSP Builder version 2.0, programmable logic's adoption for DSP applications continues to increase. Control code and DSP algorithms integrate with the Nios processor to provide a performance and cost-optimized FPGA solution that has a significant edge over generic DSP processor devices (see Figure 3). When navigating through feature and performance requirements, which change as rapidly as standards and protocols, programmable solutions deliver a priceless benefit—peace of mind.

DSP IP cores and the integration of third-party DSP system-level tools will truly put programmable logic in the forefront of DSP applications.

Figure 3. Altera DSP Technology



Shorten DSP Design Cycle Using Complex IP in MATLAB/Simulink to Quartus II Flow

The use of complex digital signal processing (DSP) intellectual property (IP), such as forward error correction (FEC) functions like Reed-Solomon, or filtering functions like finite impluse response (FIR) or infinite impulse response (IIR), is becoming standard for system-on-a-programmable-chip (SOPC) design methodology. Support for encryption and parameterization schemes is a significant factor in the success of IP adoption in SOPC development. These two key features are essential to enable complete IP evaluation in a complex custom system. For example, there could be different levels of parameterization, or a filter can require system-specific bit widths or even system-specific architecture such as direct form, cascaded, or parallel. Also, different levels of encryption can exist.

Although using IP speeds up DSP SOPC design when compared to traditional register transfer level (RTL) entry, there is strong motivation to extend the use of the hardware IP one level above traditional RTL tools.

Traditional hardware data analysis, such as waveform display or data tables, is somewhat limited when exploring design behavior in the frequency (or “Z”) domain. The typical analysis of complex modulation architectures requires multi-dimensional plotting capabilities to display specific constellation schemes. In addition to the lack of visual sinks, typical RTL tools do not always offer appropriate sources for system-level analysis, such as channel models or noise generators.

The Simulink tool from The Mathworks offers very powerful system analysis that you can use in conjunction with hardware implementation to address many aspects of IP integration. DSP Builder combines the Altera® Quartus® II development tool and MATLAB/Simulink interface to extend the hardware capabilities of Simulink to RTL designs for Altera FPGAs. Simulink IP integration via DSP Builder supports all the existing parameterization and encryption structures in the RTL domain.

This article discusses how The MathWorks’ MATLAB and Simulink tools speed up IP’s integration in an SOPC design. The Altera DSP Builder tool enables this flow by providing an automated translation of a bit-and-cycle-accurate Simulink model into a VHDL hardware implementation.

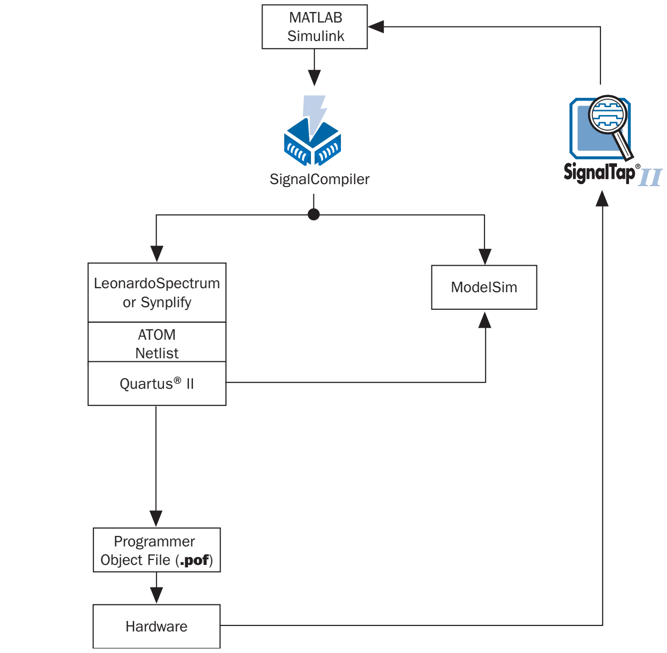
DSP Builder Overview

An overview of the DSP Builder design flow is necessary prior to describing the implementation details of DSP IP cores in Simulink. DSP Builder contains bit- and cycle-accurate Simulink blocks that cover basic operations such as arithmetic or storage functions as well as more complex functions such as FEC and filtering from the IP libraries.

As shown in Figure 1, the overall design flow starts by converting Simulink model files (**.mdl**)

continued on page 8

Figure 1. DSP Builder Design Flow



Shorten DSP Design Cycle Using Complex IP in MATLAB/Simulink to Quartus II Flow, continued from page 7

that are built using DSP Builder blocks into a RTL VHDL representation and tool command language (Tcl) scripts. After synthesis, you can compile a gate-level netlist into a binary file format called a POF that is downloaded into the target FPGA device. Figure 2 shows the RTL representation, automatically generated by the SignalCompiler feature in the DSP Builder. Figure 3 shows an amplitude modulation design modeled in Simulink.

Figure 2. RTL Representation

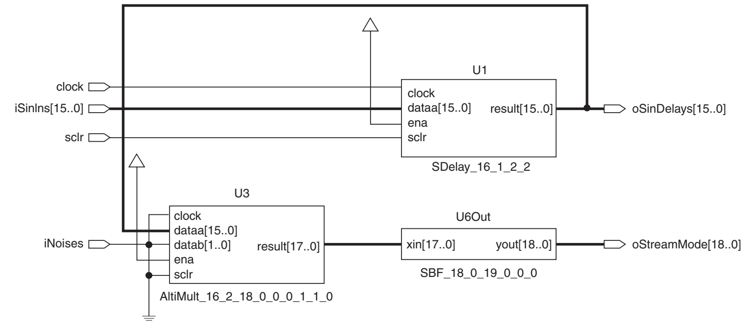
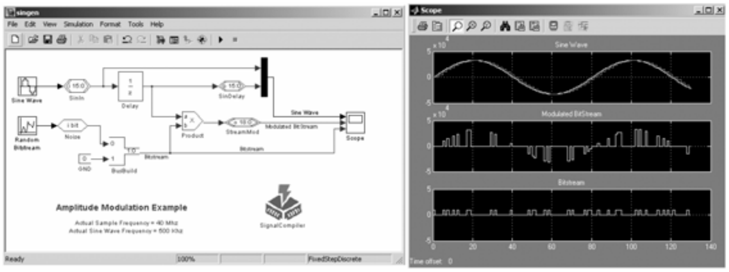


Figure 3. Simulink Model



This conversion is based on a structural analysis of the Simulink model. Each DSP Builder block is optimized to take advantage of the dedicated hardware resources, which may vary from one device family to another. For instance, an accumulator block generally uses a dedicated DSP block when designed for Stratix™ devices, but will use logic elements (LEs) when you compile the design for APEX™ II devices.

To achieve efficient hardware implementation, you should have some understanding of how the DSP Builder tool operates and the correspondence between Simulink and VHDL simulation with respect to signals and clock domains. Simulink signals, typically of type double, are mapped to VHDL buses. The DSP Builder provides several casting schemes to map Simulink double signals into hardware buses covering variable bit widths as well as multiple bus types, such as signed integer, unsigned integer, or signed binary fractional.

To simplify the design entry, the SignalCompiler block propagates bit width information across an entire data path from the source to the destination through all intermediate blocks. For example, in the amplitude modulation design (see Figure 3), the SinIn and SinDelay blocks have a bit width of 16. Therefore, SignalCompiler automatically assigns a bit width of 16 to the Intermediate Delay block. The bit width growth rule of each DSP Builder block is described in the DSP Builder User Guide, available in the Literature section of the Altera web site at **http://www.altera.com**.

The next step in the conversion is to take a close look at the clock domain correspondence between the Simulink and VHDL representations. The Simulink register transfer conversion is based on synchronous design rules. For digital designs implemented in programmable logic, the model needs to work with the Simulink fixed-step discrete solver. Therefore, all DSP Builder synchronous blocks, such as the Delay block, operate on the positive edge of the single clock domain, which is running at the system-sampling frequency. The clock pin is not graphically displayed in the block symbol in Simulink; however, when SignalCompiler converts the design to VHDL, it automatically connects the clock pin of the synchronous blocks (i.e., the Delay block) to the clock domain of the system.

By default, Simulink does not graphically display the clock enable and synchronous reset input pins of the DSP Builder's synchronous blocks. When SignalCompiler converts the design to VHDL, it connects these pins to V_{CC} and the system reset, respectively. If you turn on the **Use Control Inputs** parameter in the **Parameter** dialog box of each of the DSP Builder synchronous blocks, you can access and drive the clock enable and synchronous reset input pins graphically in the Simulink software. Most DSP systems operate with a normalized frequency with respect to the Nyquist rate. By default, DSP Builder works with a Nyquist rate of 1.0. All discrete sampling blocks operate at this normalized sampling frequency.

A Complete Flow

The Altera Stratix DSP development kit, which combines the DSP Builder and the Stratix DSP development board, enables the design of a complete system from the Simulink cockpit, giving you the tools to:

- Design compilation from Simulink to the Quartus II software
- Download the resulting POF into the DSP board
- Analyze in MATLAB the internal hardware node using the SignalTap® II logic analyzer

Although system-level simulations such as Simulink offer a speed advantage over traditional RTL simulation, there is still a strong need in DSP system analysis to extend simulation coverage time in real hardware. This coverage time applies to a wide variety of systems, such as adaptive filtering or bit error rate analysis, where hours of real-time simulation are required to trigger probabilistic blocks.

The Quartus II software includes the SignalTap II logic analyzer, with which you can set up event triggers, configure memory, and display captured waveforms. Samples are saved to internal embedded RAM blocks when the logic analyzer is triggered, and are subsequently streamed off chip via the Joint Test Action Group (JTAG) port using the ByteBlasterMV™ or MasterBlaster™ download cable. The captured data is then stored in a text file, displayed as a waveform in a MATLAB plot, and transferred to the MATLAB work space as a global variable.

IP Design Flow in Simulink

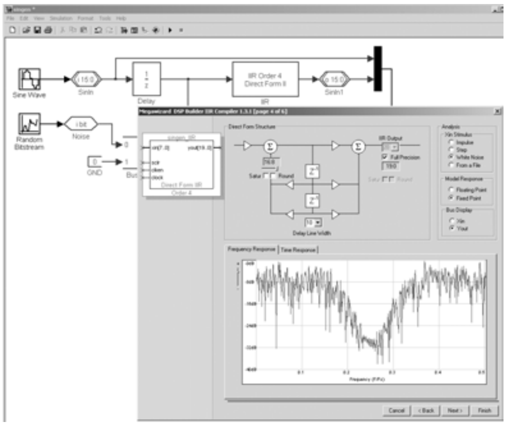
From a user standpoint, hardware IP cores in Simulink are presented as discrete block library elements. DSP Builder uses the Altera MegaWizard® Plug-In Manager technology to configure the IP within Simulink. The MegaWizard Plug-In Manager is a stand-alone graphical user interface (GUI) that you can use to pass static parameters to complex IP in an HDL design flow. This technology, which manipulates encrypted IP, is extended to pass parameters to a C++ model.

The Simulink simulation engine is state based. Simulink supports the C++ model via S-function callback methods. Each discrete block contains a set of methods per state. The simulator schedules these functions during initialization, output result, and register update events.

A VHDL IP model contains a generic section for static parameters like bit width or architecture type, a port section for the input and output port signal, and an architecture section to describe the function. The task of translating a C++ model to a VHDL model consists of mapping correctly those three sections with respect to the simulator states.

Figure 4 illustrates how a single user interface (the IIR MegaCore® function) is deployed to configure in Simulink both the C++ model and VHDL representation. At this point, the conversion to hardware is closely tied to the coding rules used for the Simulink C++ model and VHDL module.

Figure 4. IIR MegaCore Function



To maintain IP protection, the models are deployed as compiled dynamic linked libraries in the Simulink domain. In the VHDL domain, the encryption is based on Altera's OpenCore® or OpenCore Plus evaluation tools. The OpenCore evaluation feature lets you test-drive IP cores for free using the Quartus II software. The OpenCore Plus evaluation feature enhances the OpenCore feature by supporting free hardware evaluation. This feature allows you to generate

continued on page 10

Shorten DSP Design Cycle Using Complex IP in MATLAB/Simulink to Quartus II Flow, continued from page 9

time-limited programming files for a design that includes Altera MegaCore functions. With these files, you can perform board-level design verification before deciding to purchase licenses for the MegaCore functions.

With additional development, you can make these C++ IP models compatible with the The MathWorks Real Time Workshop. This powerful feature can offer a single DSP design representation, which not only links system-level features with implementation features, but also facilitates hardware and software partitioning.

Conclusion

This article covers the needs of both hardware engineering and system-level engineering. DSP Builder offers an extension of RTL and debug capabilities to system-level tools and enables IP evaluation at the system level. For a system-level engineer, DSP Builder is a rapid prototyping vehicle, which requires minimal FPGA expertise and extends the system analysis from floating point to fixed point.

Hardware-Verified DSP Performance with Stratix Devices

Stratix™ devices feature the versatile digital signal processing (DSP) blocks along with dedicated functional blocks and routing, delivering high performance. Table 1 summarizes the performance from the DSP blocks in *Stratix Programmable Logic Device Family Data Sheet.*

Table 1. Stratix Data Sheet Performance Results for DSP Blocks

Definition	Speed Grade			Unit
	-5	-6	-7	
Data Width = (9×9)				
Simple Multiplier	336	294	256	MHz
Two-Multiply Adder	336	294	256	MHz
Four-Multiply Adder	336	294	256	MHz
Data Width = (18×18)				
Simple Multiplier	279	237	206	MHz
Multiply Accumulator	279	237	206	MHz
Two-Multiply Adder	279	237	206	MHz
Four-Multiply Adder	279	237	206	MHz
Data Width = (36×36)				
Single Multiplier	279	237	206	MHz

Due to characterization results that showed silicon is faster than expected, Altera was able to improve the specifications in the *Stratix Programmable Logic Device Family Data Sheet*. The data was taken under worse-case operating conditions.

Test Conditions

The following is a list of test conditions:

- Testing performed on typical -6ES silicon
- Ambient temperature = 85° C
- V_{CCINT} (-5%) = 1.425 V

Altera performs hardware characterization testing on all FPGAs to verify timing specifications. Characterization data with system noise and corner silicon is required before the preliminary specification in the Quartus II software is finalized. The Quartus II software version 2.1 service pack 1 will reflect the new data sheet specifications.

The DSP block supports operation in eight modes. Table 2 shows the preliminary Quartus II software specification and the measured characterization data results.

Table 2. DSP Block Support for the Stratix EP1S25 Device (–6 Speed Grade)

DSP Block Operation	Definition	Quartus II Version 2.1	-6ES Silicon Data	Unit
Mode 1	18×18-bit Multipliers	232	290	MHz
Mode 2	18×18-bit Multiply-Accumulator	232	284	MHz
Mode 3	Sum of two 18×18-bit Multipliers	232	281	MHz
Mode 4	Sum of four 18×18-bit Multipliers	232	288	MHz
Mode 5	9×9-bit Multipliers	281	376	MHz
Mode 6	Sum of two 9×9-bit Multipliers	281	339	MHz
Mode 7	Sum of four 18×18-bit Multipliers	281	339	MHz
Mode 8	36×36-bit Multiplier	232	282	MHz

Stratix DSP Development Kit Provides Designers with Complete Out-of-the-Box Experience

Many digital signal processing (DSP) algorithms can run faster in programmable logic and offer significant performance advantages over conventional DSP processors. The challenge is how to prove this enhanced capability in real-system designs. Whether you are looking to increase performance, re-partition your system, or evaluate some of Altera's intellectual property (IP) MegaCore® functions, the Stratix™ DSP development kit can help. The core of this kit is the DSP Builder, which is a tool that interfaces the industry-leading MATLAB/Simulink environment from The MathWorks to the Altera® Quartus® II software development flow. DSP Builder allows you to target a design from the Simulink environment to an Altera Stratix DSP development board, without extensive programmable logic expertise.

The Stratix DSP development board has all the necessary analog connectivity to plug into an existing design or test equipment. Additionally, Altera's unique OpenCore® Plus technology allows users to evaluate DSP MegaCore functions in hardware as well as within the simulation environment of Simulink. The Stratix DSP development kit contains everything that you need to start converting Simulink designs to Altera programmable logic.

Development Kit Contents

The items included in the kit provide you with a complete out-of-the-box experience. In addition to power supplies, cables, and documentation, the Stratix DSP development kit includes the following items.

DSP Development Board

The DSP development board is a hardware platform that provides DSP system designers with a solution for prototyping and verifying signal processing designs. Some of the key features on the board include:

- Stratix EP1S25F780 device
- Analog I/O pins
  - Two-channel, 12-bit, 125 million samples per second (MSPS) analog-to-digital (A/D)
  - Two-channel, 14-bit, 165 MSPS digital-to-analog (D/A)
- Digital I/O pins
  - 40-pin connectors for Analog Devices A/D converter evaluation boards
  - Connector for TI TMS320 cross-platform daughter card
  - Expansion/prototype headers
  - RS232 serial port
- Two Mbytes of synchronous SRAM

The Stratix DSP development board has all the necessary analog connectivity to plug into an existing design or test equipment.

DSP Builder

DSP Builder is a digital processing tool that interfaces the Simulink system-level tool with Altera's industry-leading Quartus II development software. DSP Builder provides a seamless design flow, where you can perform algorithmic design in the MATLAB software and system integration in the Simulink software. You can then port the design to HDL files to use in the Quartus II software.

Quartus II Development Software

The Quartus II development software provides a comprehensive environment for designing and prototyping systems with the Altera DSP development kit.

continued on page 12

Stratix DSP Development Kit Provides Designers with Complete Out-of-the-Box Experience, continued from page 11

30-Day MATLAB/Simulink Evaluation Software

MATLAB/Simulink is an industry-leading, system-level tool from The MathWorks. You can obtain a 30-day evaluation license from The MathWorks web site at **http://www.mathworks.com**.

Evaluation MegaCore IP Functions

The kit contains evaluation versions of Altera DSP MegaCore functions. You can evaluate the functions using the OpenCore feature, which lets you quickly and easily verify the functionality and performance of an IP function. Additionally, these functions support the OpenCore Plus hardware evaluation feature, which allows you to generate time-limited programming files for prototyping and board-level simulation.

System Reference Designs/Labs

Included with the kit are several system-level reference designs and labs that help you get started with the kit.

The ordering code and pricing for the Stratix DSP development kit is DSP-BOARD/S25 and $1,995, respectively. The kit will be shipping in early Q4 2002. To learn more about Altera's DSP kit offerings and discover a new way to implement your DSP systems, go to the Altera web site at **http://www.altera.com** under Products and then Development Kits.

DSP-Certified, Third-Party Design Services Partners Reduce Time-to-Market with FPGA-Based DSP Solutions

ACAP®
Altera Consultants Alliance Program

Altera's comprehensive digital signal processing (DSP) solutions, consisting of a complete design software environment, performance-optimized devices, and DSP-related intellectual property (IP), give you the opportunity to integrate entire DSP systems on a programmable chip. If you do not have the resources to take advantage of Altera's solutions, DSP-certified members of the Altera Consultants Alliance Program (ACAP®) partnership can assist you in your DSP design.

DSP-certified ACAP partners are industry-leading design service providers that use their FPGA and DSP knowledge to create system-on-a-programmable-chip (SOPC) solutions for Altera's customers.

Serving customers in North America, DSP Chapter members include Adaptive Micro-Ware, CEPD, Nuvation, PTG, Synopsys Professional Services, and Vanteon; Alcahest, El Camino, Plextek, and Synopsys Professional Services provide service to customers throughout Europe; and Dexcel Electronics Designs service customers in the Asia Pacific and India. Along with regional diversity, the selected partners have a broad range of DSP expertise.

All partners are trained and certified by Altera and have a strong relationship with the Altera® field sales force. By using these partners for DSP services, Altera customers gain access to engineers with a much-needed combination of Altera and traditional DSP expertise, thus easing the transition to an FPGA-based DSP solution. Collaboration between Altera and ACAP partners increases the quality of services and support for the customer. Using Altera® devices, tools, and IP, Altera's partners help shorten design cycles and decrease development costs and risk. Ultimately, as with any FPGA-based SOPC solution, the bottom line is reduced time-to-market.

For more information, or to engage with DSP-certified ACAP partners, contact your local Altera sales representative or visit the Altera DSP Solutions Center at **http://www.altera.com**.

Efficient Channel Coding Builds iPSTAR Satellite Modem with Altera DSP

Efficient Channel Coding, Inc. (ECC) helped to launch the first satellite broadband system in Thailand and was tasked with evaluating ways to efficiently deploy modem and transport functions for Shin Satellite's iPSTAR two-way digital satellite radio system.

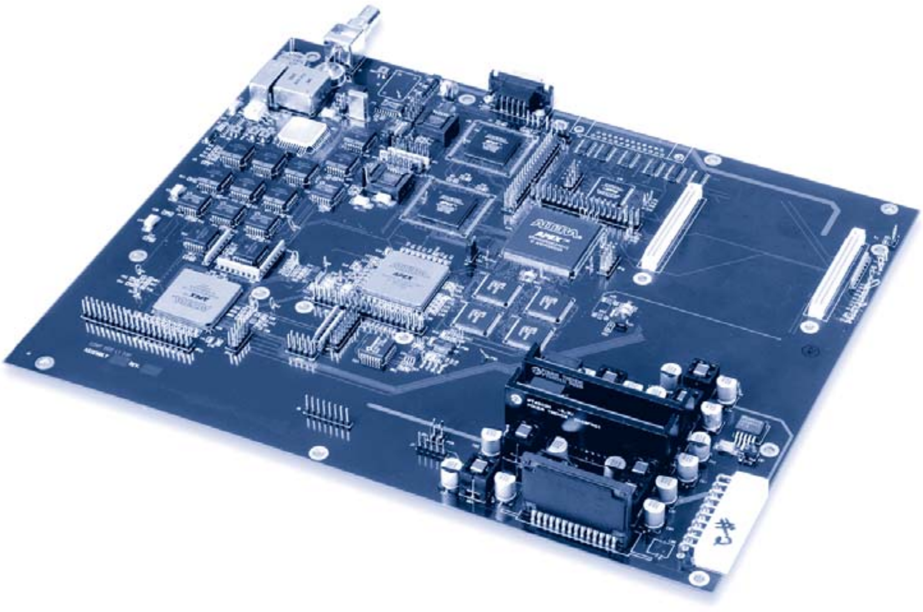
For the design, ECC integrated its high-speed modem intellectual property (IP) with several digital signal processing (DSP) functions, including advanced forward error correction (FEC), finite impulse response (FIR) filter, and fast Fourier transform (FFT) IP targeting the design for Altera's high-density programmable logic devices (PLDs).

The parallel processing capabilities of programmable logic offered an ideal solution for implementation of the iPSTAR system, delivering additional benefits over stand-alone DSP processors. ECC chose Altera's APEX™ 20K, APEX 20KE, and MAX® 3000 devices to implement the system in customer premise equipment (CPE) and transmitter and receiver head-end carrier class equipment. See Figure 1.

Altera's devices allow the head-end equipment to be updated with custom instructions to add billing and other features, delivering tangible benefits to service providers.

For further information on the ECC design and the ShinSat iPSTAR system, visit the Altera web site at **http://www.altera.com/ecc**.

Figure 1. ECC Transmitter/Receiver Head-End



Stratix

New Stratix -5 Speed Grade Released in Quartus II Version 2.1

Altera introduces a faster -5 speed grade for Stratix™ devices (see Table 1), enabling further performance improvements to the industry's fastest FPGA. Through enhancements both to silicon and software, Stratix devices provide up to a 20% design performance boost over the Quartus® II software version 2.0, and provide over 60% performance improvements over previous architectures on older software platforms. Support for Stratix devices is currently available in version 2.1.

Table 1. Stratix Device Speed Grade Availability

Device	Speed Grade
EP1S10	-5, -6, -7
EP1S20	-5, -6, -7
EP1S25	-5, -6, -7
EP1S30	-5, -6, -7
EP1S40	-5, -6, -7
EP1S60	-6, -7
EP1S80	-6, -7
EP1S120	TBD

See Tables 2 and 3 for Stratix availability schedules and software support.

Table 2. Stratix Device Availability

Device	Device Availability	Pin-Out Availability
EP1S10	Now	Now
EP1S20	Q4 2002	Now
EP1S25	Now	Now
EP1S30	Q4 2002	Now
EP1S40	Q4 2002	Now
EP1S60	First half of 2003	Now
EP1S80	September 2002	Now
EP1S120	First half of 2003	TBD

Table 3. Stratix Devices & Quartus II Software Advanced Support Availability

Device	Package	Quartus II Software Advanced Support Availability
EP1S10	672-pin BGA *(1)*	Now
	672-pin FineLine BGA®	Now
	780-pin FineLine BGA	Now
EP1S20	672-pin BGA	Now
	672-pin FineLine BGA	Now
	780-pin FineLine BGA	Now
EP1S25	672-pin BGA	Now
	672-pin FineLine BGA	Now
	780-pin FineLine BGA	Now
	1,020-pin FineLine BGA	Now
EP1S30	780-pin FineLine BGA	Now
	956-pin BGA	Now
	1,020-pin FineLine BGA	Now
EP1S40	956-pin BGA	Now
	1,020-pin FineLine BGA	Now
	1,508-pin FineLine BGA	Now
EP1S60	956-pin BGA	Now
	1,020-pin FineLine BGA	Now
	1,508-pin FineLine BGA	Now
EP1S80	956-pin BGA	Now
	1,508-pin FineLine BGA	Now
	1,923-pin FineLine BGA	TBD
EP1S120	1,923-pin FineLine BGA	TBD

Note to Table 3:

(1) BGA: Ball-grid array.

EP1S10 Devices Now Shipping

The second member of the Stratix device family, the EP1S10 device, is now shipping. Following the timely rollout of the Stratix EP1S25 device, the EP1S10 now extends Altera's product leadership in performance and features.

EXCALIBUR

Targeting applications requiring the functionality of microprocessors and the flexibility of FPGA designs, Altera's Excalibur™ device family of embedded processor FPGA solutions provides a platform to integrate these requirements on a single device. Excalibur devices tightly integrate Altera's FPGA architecture with an ARM922T™ processor subsystem (stripe). This implementation optimizes the performance of the processor subsystem and enhances those applications that require high-speed computing capabilities alongside advanced, high-performance programmable logic.

Excalibur solutions also enable stand-alone processing and programming of the device without external influence. You are free to focus on the custom logic and intellectual property (IP) blocks required to complete your system-on-a-programmable-chip (SOPC) designs. IP cores, such as the 10/100 Ethernet MAC, double data rate (DDR) SDRAM controllers, universal asynchronous receiver/transmitters (UARTs), and peripheral component interconnect (PCI) interfaces from Altera's IP MegaStore™ web site and third-party IP partners, further facilitate the design process.

Excalibur embedded processor solutions are successfully designed into industrial automation systems, test equipment, medical equipment, avionic products, networking systems, and printer products. The Excalibur device family is well suited for both ASIC prototyping quantities and low to moderate volume production.

Processor Subsystem

The processor subsystem (stripe) includes the following:

- 133-, 166-, or 200-MHz ARM922T processor
- 8-Kbyte instruction and 8-Kbyte data caches
- Memory management unit (MMU) for real-time operating system (RTOS) support
- 32-, 128-, 256-Kbyte SRAM memory
- 16-, 64-, 128-Kbyte dual-port SRAM memory
- Interrupt controller
- Single data rate (SDR) and DDR SDRAM and flash controllers
- EBI controller (flash, SRAM)
- UART
- Joint Test Action Group (JTAG) debug and ETM9™ trace support
- Stripe-to-FPGA logic interface
- Choice of development tools, operating systems, and applications software IP from third-party companies

FPGA Architecture

The FPGA architecture includes the following:

- 1.8-V, 0.18-µm SRAM process
- 100,000 to 1,000,000 gates
  - EPXA1 device: 4,160 logic elements (LEs)
  - EPXA4 device: 16,640 LEs
  - EPXA10 device: 38,400 LEs
- MultiCore™ architecture: dual-port RAM, content addressable memory (CAM), product-term architecture
- Multiple phase-locked loops (PLLs)
- Advanced I/O support: PCI, PCI-X, LVDS, SSTL-3, GTL+, AGP, LVPECL, HSTL, LVTTL, and LVCMOS
- Altera® and third-party IP

Excalibur devices are supported by a versatile selection of Altera and third-party tools such as EDA software, compiler, assembler, synthesis, verification, debug, and real-time trace. All Excalibur development kits include the development board, documentation, connection cables, power supply, Quartus II software, including SOPC Builder, GNUPro tools, and an Excalibur Solutions Pack containing third-party demonstration and evaluation tools.

Ordering Information

The following is a list of Excalibur device ordering codes:

- EPXA1 (EPXA1F484Cx, EPXA1F484I2, EPXA1F672Cx)
- EPXA4 (EPXA4F672Cx, EPXA4F672I2, EPXA4F1020Cx)
- EPXA10 (EPXA10F1020Cx)

For more information, visit the Altera web site at **http://www.altera.com** or contact your local distributor or sales office.

continued on page 16

Devices & Tools, continued from page 15

GNUPro Development Tools

Red Hat's GNUPro Toolkit is now included with Altera software subscriptions and provides Excalibur and Nios® developers with industry-standard embedded software development tools. The GNUPro compiler and debugger is an open-source C/C++ development tool suite that is optimized for the Excalibur and Nios embedded processors. It delivers an environment familiar to design engineers, including:

- GNU C compiler (gcc) and GNU C++ compiler (g++)
- GNU debugger (gdb) source and assembly-level debugger
- GNU assembler (gas)
- GNU linker (ld)
- Insight interface for GNU debugger
- GNU software code profiler (gprof)

Some of the many benefits to using the GNUPro tools include:

- Complete set of tools needed to design and debug software applications running on Excalibur devices and/or Nios embedded processors
- No new tool chain to learn (industry standard)
- Portability within Altera embedded processor solutions
- Open source code (GNUPro tools available with complete source code)
- No license fees (chain included with the development kits)

GNU Compilers

GNUPro includes a compiler suite consisting of the optimized ANSI-C compiler (gcc) and the C++ compiler (g++). The toolkit also includes additional tools and utilities, including the GNU assembler (gas), GNU linker (ld), and many Nios and ARM® processor-specific binary utilities.

GNU Debugger

The GNUPro Toolkit provides all the tools necessary for effective software debugging. The command-line (gdb) debugger provides complete access to program state, including source and assembly level, variables, registers, and memory. Additionally, the gdb debugger easily handles large, complex programs.

Red Hat's Insight tool is a graphical user interface to the gdb debugger. Insight contains all of the expected features of a source-level debugger; it is capable of displaying mixed mode/split screen source and assembly-level code, with "watch points" and "breakpoints" windows and a graphical stack display for quick navigation.

RedBoot

Included with the GNUPro tools is RedBoot for Excalibur devices. RedBoot is Red Hat's standard embedded system debugging environment that supports embedded Linux on Excalibur devices. It includes facilities such as network downloading and debugging, and a simple flash file system for boot images.

GNUPro Tool Availability

GNUPro tools are included in the SOPC Builder version 2.6 CD that is shipping with the Quartus II software version 2.1. The GNUPro toolkit is also included in the Nios Development Kit.

SOPC Builder

SOPC Builder Version 2.6 Now Shipping

Altera is now shipping the SOPC Builder system development tool version 2.6 along with the Quartus II software version 2.1. All Altera software subscription customers can now take advantage of the industry's first complete SOPC integration tool.

The SOPC Builder system development tool is an Altera-pioneered technology that dramatically increases designer productivity, slashing time-to-market while enabling the creation of high-bandwidth systems. Version 2.6 of this development tool introduces advanced support for the Excalibur device family, with their associated ARM922T processing cores. You can now use SOPC Builder to configure the processor stripe of Excalibur devices, add soft peripherals to the advanced high-performance bus (AHB), and generate a custom software development environment for each unique processor configuration. Additionally, with the availability of an Avalon™-to-AHB bridge, you can connect all of Altera's Avalon interface peripherals to the ARM922T processor, giving you flexibility beyond bus interface standards. For more information on SOPC Builder, see **http://www.altera.com/sopcbuilder**.

Nios Processor

Test Drive the Nios Embedded Processor

Included with SOPC Builder is an OpenCore® Plus version of the Nios embedded processor. The OpenCore Plus Nios processor evaluation provides a free test drive of the processor prior to purchase. With this version, you can generate, simulate, compile, and download to Nios processor-based hardware systems, creating time-limited hardware evaluation designs. Customers can upgrade to a fully supported Nios product by purchasing the Nios Development Kit for $995. The full product provides you with 12 months of product upgrades as well as technical support. For more details, go to **http://www.altera.com/nios**.

Nios Subscription Renewal

Annual subscription renewals are available for the Nios embedded processor. With the Nios Embedded Processor Subscription Program, you will receive automatic updates to the Nios processor, peripherals, SOPC Builder, and other related development tools for 12 months. Visit **http://www.altera.com/niosrenewal** for more details.

APEX II

APEX II Availability

All members of the APEX™ II device family are shipping. APEX II devices range in density from 16,640 to 67,200 logic elements (LEs) and are memory-rich, offering 4 Kbits of memory per embedded system block (ESB), with total device memory ranging from 416 Kbits to 1.1 Mbits.

The APEX II device family supports high-speed data transfers through a wide range of high-speed I/O standards such as LVDS, PCML, LVPECL, HSTL, SSTL, and HyperTransport™ technology. With True-LVDS™ circuitry, APEX II devices can achieve data transfer rates of up to 1 gigabit per second (Gbps) per channel. With these I/O features, you can use APEX II devices in the following applications:

- PHY-link layer interface applications (POS-PHY, Flexbus, and UTOPIA)
- Host-processor interface applications (HyperTransport technology, peripheral component interconnect (PCI), and PCI-X)
- Switch fabric interfaces (CSIX and LCS)
- External memory interfaces (double data rate (DDR), zero bus turnaround (ZBT), and quad data rate (QDR) memory devices)

See Tables 4 and 5 for availability and software support for APEX II devices.

Table 4. APEX II Device Availability

Device	Production Availability
EP2A15	Now
EP2A25	Now
EP2A40	Now
EP2A70	Now

Table 5. APEX II Devices & Quartus II Software Support

Device	Package	Software Support
EP2A15	672-pin FineLine BGA 724-pin BGA	Now
EP2A25	672-pin FineLine BGA 724-pin BGA	Now
EP2A40	672-pin FineLine BGA 724-pin BGA 1,020-pin FineLine BGA	Now
EP2A70	724-pin BGA 1,508-pin FineLine BGA	Now

continued on page 18

Devices & Tools, continued from page 17

APEX II HardCopy Solution

Altera offers a migration option from APEX II to HardCopy™ devices for system designers who need a low-risk, cost-reduction solution for high-volume production. You can prototype time-sensitive applications using APEX II devices and migrate the design to HardCopy devices for high-volume production. HardCopy devices preserve the functionality and timing of the design and allow you to improve time-to-market at the lowest cost.

APEX II Industrial Offerings

All the industrial-grade devices for the APEX II device family are now available to further compress design cycles for the fastest possible time-to-market. Industrial-grade production versions of the device offerings are available in a -8 speed grade. Table 6 shows the availability for industrial-grade offerings.

Table 6. APEX II Industrial Device Offering

Device	Package	Production Availability
EP2A15	672-pin FineLine BGA	Now
EP2A25	672-pin FineLine BGA 724-pin BGA	Now Now
EP2A40	724-pin BGA 1,020-pin FineLine BGA	Now Now

MERCURY

Mercury Silicon Available in Production Mode

All devices and all speed grades of the Mercury™ device family are shipping in production mode, including industrial-temperature offerings in both product lines (see Table 7). High-speed 1.25-gigabits per second (Gbps) serial links featuring clock data recovery (CDR) circuitry and an embedded serializer/deserializer (SERDES) make these devices ideal for serial backplane applications.

Table 7. Mercury Device Availability

Device	Package	Temperature Grade	Production Availability
EP1M120	484-pin FineLine BGA	Commercial in -5, -6, -7 speed grade	Now
		Industrial in -6 speed grade	Now
EP1M350	780-pin FineLine BGA	Commercial in -5, -6, -7 speed grade	Now
		Industrial in -6 speed grade	Now

APEX

APEX 20KC Available in Production Mode

All APEX 20KC devices are available with all parts and packages shipping in full production mode. Table 8 shows the availability schedule for APEX 20KC devices.

Table 8. APEX 20KC Device Availability

Device	Production Availability
EP20K200C	Now
EP20K400C	Now
EP20K600C	Now
EP20K1000C	Now

Industrial-Grade APEX Offerings

Industrial-grade APEX devices are now available in a wide variety of package offerings. Refer to Tables 9, 10, and 11.

Table 9. APEX 20KC Device Industrial Offering

Device	Package	Speed Grade
EP20K200C	484-pin FineLine BGA	-8
EP20K400C	652-pin BGA 672-pin FineLine BGA	-8
EP20K600C	652-pin BGA 672-pin FineLine BGA	-8
EP20K1000C	1,020-pin FineLine BGA	-8

Table 10. APEX 20KE Device Industrial Offering

Device	Package	Speed Grade
EP20K30E	144-pin FineLine BGA	-2X *(1)*
EP20K60E	144-pin FineLine BGA 208-pin PQFP *(2)* 324-pin FineLine BGA	-2X *(1)*
EP20K100E	144-pin FineLine BGA 240-pin PQFP 324-pin FineLine BGA 356-pin BGA	-2X *(1)*
EP20K160E	484-pin FineLine BGA	-2X *(1)*
EP20K200E	240-pin PQFP 356-pin BGA 484-pin FineLine BGA 672-pin FineLine BGA	-2X *(1)*
EP20K300E	240-pin PQFP 652-pin BGA 672-pin FineLine BGA	-2X *(1)*
EP20K400E	652-pin BGA 672-pin FineLine BGA	-2X *(1)*
EP20K600E	652-pin BGA 672-pin FineLine BGA	-2X *(1)*
EP20K1000E	652-pin BGA 672-pin FineLine BGA	-2X *(1)*

Notes to Table 10:

(1) The “X” denotes PLL and LVDS support.

(2) PQFP: Plastic quad flat pack.

Table 11. APEX 20K Device Industrial Offering

Device	Package	Speed Grade
EP20K100	208-pin PQFP 240-pin PQFP	-2V (1)
EP20K200	240-pin PQFP 484-pin FineLine BGA	-2V (1)
EP20K400	652-pin BGA 672-pin FineLine BGA	-2V (1)

Note to Table 11:

(1) The “V” denotes 5.0-V tolerant I/O interfaces.

ACEX

ACEX 1K Availability

ACEX® 1K devices are available in quad flat pack (QFP) and FineLine BGA packages in 576-, 1,728-, 2,880-, and 4,992-LE densities. These cost-optimized devices are specially suited for low-cost, high-volume applications.

Altera continues to provide the lowest-cost solution in the industry.

Free software support for all ACEX 1K devices is available in the Quartus II Web Edition software version 2.1, which is available for download at **http://www.altera.com**.

MAX

Improved MAX Design Performance in the MAX+PLUS II & Quartus II Software

MAX devices are now supported in the Quartus II software version 2.1, including the free downloadable Web Edition. The software integrates the support of both complex programmable logic devices (CPLDs) and FPGAs into a single platform, and significantly improves second-time fitting for MAX designs. See Table 12.

Table 12. Software Support for MAX Devices

Device Family	Quartus II Support	MAX+PLUS II Support
MAX 7000		✓
MAX 7000E		✓
MAX 7000S		✓
MAX 7000AE	✓	✓
MAX 7000B	✓	✓
MAX 3000A	✓	✓

MAX® device performance has also improved in the MAX+PLUS® II software version 10.2, which includes the PALACE™ Compiler software from Altera’s partner Aplus Design Technologies, Inc. (ADT). The PALACE software complements the MAX+PLUS II software by using advanced physical re-synthesis algorithms and timing constraints to increase the design’s performance. ADT’s software can increase the design performance of MAX devices by up to 50%.

For more information on the improved MAX design performance, contact your local Altera sales representative. Your Altera sales representative can also help you choose the optimal software support for your design needs.

continued on page 20

Devices & Tools, continued from page 19

EPM3512A Device Availability

Altera now extends the popular 3.3-V MAX 3000A family to include 512 macrocell devices. With the introduction of the EPM3512A device, the MAX 3000A family now offers a complete high-volume, low-cost solution with densities ranging from 32 to 512 macrocells, offering customers an ideal pairing of cost and density. Fabricated on a 0.3-µm, four-layer-metal process, the EPM3512A device provides 7.5-ns pin-to-pin performance and frequency counter speeds of up to 115 MHz. As with previous MAX 3000A device offerings, the EPM3512A device has an enhanced in-system programmability (ISP) feature set, supports both Jam™ standard test and programming language (STAPL), MultiVolt™ I/O (2.5-, 3.3-, and 5.0-V) operation, and is IEEE 1532 compliant. Table 13 shows the packages and speed grades available for the EPM3512A device. Production quantities of the EPM3512A device are now available.

Table 13. EPM3512A Devices

Package	Speed Grade
208-pin PQFP	-7, -10
256-pin FineLine BGA	-7, -10

Software Support

The EPM3512A device is supported by both the Quartus II software version 2.1 and the MAX+PLUS II software version 10.2. If you want to evaluate the new device, download the software from the Altera web site at **http://www.altera.com**.

MAX Process Transition

Altera has migrated all existing MAX 7000AE and MAX 3000A devices from 0.35-µm process to an advanced 0.3-µm process. All timing and reliability characteristics for the devices remain within the data sheet specifications. Altera provides a comprehensive Process Transition Report (also referred to as the data pack) for each device that compares the key characterization and reliability data for the two processes.

CONFIGURATION

Enhanced Configuration Devices

Altera's enhanced configuration devices enable remote system upgrades. Coupled with dedicated circuitry in Stratix devices, remote configuration allows you to get your products to market in the fastest time possible, while ensuring the capability to keep up with evolving standards and protocols.

Enhanced configuration devices provide a complete single-device solution for a wide range of density requirements. Vertical migration capability allows you to easily migrate from the EPC4 to the EPC8 to the EPC16 device in the same package without having to change the board layout. Commercial and industrial grade EPC4, EPC8, and EPC16 devices are all now available.

Enhanced configuration devices offer ISP through a built-in IEEE standard for boundary-scan-based, in-system configuration standard of programmable devices (IEEE 1532). The inclusion of ISP and re-programmability provides a significant advantage over one-time programmable solutions by introducing flexibility and reusability to the configuration process.

Altera's enhanced configuration devices also introduce numerous features for specialized configuration needs. These features include an external flash interface that allows unused portions of the flash memory to be used as general-purpose memory, parallel configuration capability to accelerate configuration times, a new page mode that allows you to store multiple configurations, block protection for partial reprogramming support, and full clocking flexibility through the programmable clock and external clock features. This advanced feature set enhances the overall programmable logic device (PLD) design experience.

Design Software

Quartus II Version 2.1 Includes New Timing Closure Flow

The Quartus II software version 2.1 is now shipping, and includes a new timing closure methodology facilitated by a suite of tools that allows a hardware design to achieve timing requirements on all clocks signals with fewer iterations. For more information, see "Quartus II & the New Timing Closure Methodology" on page 35.

In addition to integration to third-party synthesis software from Mentor Graphics®, Synopsys, and Synplicity®, the Quartus II design software includes integrated VHDL and Verilog HDL synthesis technology. For more information, refer to "Quartus II Verilog HDL & VHDL Integrated Synthesis" on page 33 and *AN 238: Using Integrated RTL Synthesis in the Quartus II Software.*

Additionally, version 2.1 features enhancements and improvements in FPGA design, embedded software design, the LogicLock™ block-based design flow, place-and-route, verification, and device support. Refer to Table 14 and the Design Software section of the Altera web site for more information.

continued on page 22

Table 14. New Features in Quartus II Version 2.1

Design Flow Area	New Features
Design	■ Device and EDA tool settings can now be assigned in the New Project wizard ■ Device migration is now supported for APEX II, APEX 20K, APEX 20KE, APEX 20KC, and MAX 7000 device families ■ HardCopy device's Design Assistant design rule checker checks designs to make sure they can easily be migrated to low-cost, high-volume HardCopy devices ■ Tool command language (Tcl) scripts can be assigned to the new Tcl toolbar buttons
Embedded Software Design	■ The SOPC Builder system development tool is integrated into the Quartus II user interface and now supports Excalibur devices ■ The Red Hat GNUPro C/C++ development tools are provided with Altera subscriptions
LogicLock Block-Based Design Flow	■ With the new Reserve nodes feature, only nodes assigned to a region are allowed in the region; these nodes are used to reserve areas of a device ■ Support for virtual I/O pins allows users to specify which LogicLock region I/O pins will be internal nodes in the top-level design ■ Soft region support allows nodes to be placed outside of a LogicLock region to meet timing requirements ■ Resource entity usage information is added to the Project Navigator
Place-and-Route	■ New timing closure flow increases designer productivity; new features include the Timing Closure Floorplan Editor, ability to make path-based assignments, netlist optimizations, and incremental, block-based placement ■ Initial placement configuration option is available to improve performance by an extra 5% ■ Quality-of-results enhancements are added for the Stratix devices
Verification	■ Introduction of the SignalTap® II Embedded Logic Analyzer ■ Introduction of the Excalibur Stripe Simulator (ESS), to simulate the ARM embedded processor stripe ■ Improved SignalProbe™ routing success rate for APEX 20K, APEX II, and Excalibur device families ■ PowerGauge™ support for APEX II and Excalibur embedded processors, updated support for Mercury devices ■ Support for Verplex Conformal LEC software formal verification flow for APEX 20KE, APEX 20KC, and APEX II designs ■ Support for Atrenta Spyglass and Synopsys LEDA design-rule checking tools
OS Support	■ Added support for Windows XP

Devices & Tools, continued from page 21

Support for Faster -5 Speed Grade Stratix Devices

Full compilation, simulation, timing analysis, and programming support is available in the Quartus II software version 2.1 for the new devices and device packages located in Table 15. The Quartus II software version 2.1 also includes support for faster -5 speed grade Stratix devices.

Table 15. New Quartus II Device Support

Device Family	Device	Package
Stratix	EP1S25	672-pin BGA
		672-pin FineLine BGA
		780-pin FineLine BGA
		1,020-pin FineLine BGA
MAX 3000A	EPM3512A	208-pin PQFP
		256-pin FineLine BGA

Quartus II Web Edition Device Support Expanded

The Quartus II Web Edition software is an entry-level version of the Quartus II design software supporting selected Stratix, APEX II, APEX 20KE, Excalibur, MAX 7000, MAX 3000, FLEX 10KE, ACEX 1K, and FLEX 6000 devices. With PowerFit™ place-and-route technology, the Quartus II Web Edition software lets you experience the performance and compile time benefits of the Quartus II software.

The Quartus II Web Edition software includes a complete environment for PLD design including schematic- and text-based design entry, HDL synthesis, place-and-route, verification, and programming. The Quartus II Web Edition software removes all barriers for designing for Altera device architectures for high-performance applications.

The Quartus II Web Edition software version 2.1 now supports devices from all mainstream Altera device families, including:

- Stratix (EP1S10) New
- APEX II (EP2A15) New
- Excalibur (EPXA1) New
- APEX 20KE (EP20K30E, EP20K60E, EP20K100E, EP20K160E)
- ACEX 1K (all devices)
- FLEX 10KE (EPF10K30E, EPF10K50S, EPF10K100E, EPF10K130E, EPF10K200S)
- FLEX 6000 (all devices)
- MAX 7000AE (all devices) New
- MAX 7000B (all devices) New
- MAX 3000A (all devices) New

The Quartus II Web Edition software now includes integrated VHDL and Verilog HDL synthesis. The Mentor Graphics® LeonardoSpectrum™-Altera synthesis tool is also available from the Altera web site to use in conjunction with the Quartus II Web Edition software. Quartus II Web Edition users can also use third-party synthesis tools from Mentor Graphics, Synopsys, and Synplicity.

Quartus II Web Edition software version 2.1 is available now for downloading from the Altera web site and is also included on the Altera Digital Library (ADL) CD-ROM.

MAX+PLUS II Version 10.2 Improves MAX Design Performance

The MAX+PLUS II and MAX+PLUS II BASELINE software version 10.2 include the Aplus Design Technologies (ADT) PALACE Compiler option, which can increase performance by up to 50% for MAX 7000 and MAX 3000 designs. The ADT PALACE software optimally compiles designs using pin and timing constraints and advanced physical synthesis algorithms to improve performance and place-and-route results.

Other improvements to the MAX+PLUS II software include support for the Windows XP operating system and resolution of nearly 50 customer issues.

The Quartus II software version 2.1 upgrade shipments include the MAX+PLUS II version 10.2 CD-ROM and ADT documentation describing the benefits of enabling the ADT PALACE Compiler option in the MAX+PLUS II software (available in PC and Solaris versions only). MAX+PLUS II BASELINE version 10.2 is now available for downloading from the Altera web site and is included on the ADL CD-ROM.

The E+MAX™ software will not be updated to version 10.2; E+MAX software version 10.1 will remain on the Altera web site until the end of December 2002. E+MAX licenses will not be issued after March 31, 2003.

Stratix DSP Accelerates Medical Imaging

Michael Worry
CEO, Nuvation Engineering
Altera Certified Design Center

Recently Nuvation encountered a medical imaging application that required algorithm acceleration to both increase image resolution and reduce processing time. As an Altera® Certified Design Center (CDC), Nuvation often encounters conflicting goals that can only be solved with the use of innovative technologies. The digital signal processing (DSP) capabilities of Altera's Stratix™ devices provide the architecture to solve this problem.

Computationally-Intensive Design

The customer's process required matching and indexing subsequent captured images, which is accomplished by an algorithm known as normalized cross-correlation. This algorithm requires computationally-intensive convolution of images, which is primarily implemented in multiply accumulators. A further functional requirement is that the image that can grow in either size or resolution for future products, resulting in a exponential increase in required processing power.

Traditionally, a design like this would be implemented in a digital signal processor. A high-end digital signal processor can produce roughly a half dozen billion multiply-accumulates per second (GMACS). As a comparison, each DSP block in an Altera Stratix FPGA can produce two GMACS. The number of DSP blocks ranges from 6 with the entry product to 28. This means Stratix devices are capable of producing several multiples higher DSP throughput, or 228 parallel multiplications and a combined throughput of 56 GMACS. If further processing is required, you can instantiate additional multiplier-accumulators (MACs) in the distributed logic elements (LEs) and memory. Coupling this with the ability to drop in larger footprint-compatible devices allows you to start with a cost-effective FPGA and scale as needed without a board spin, which enables future functionality and enhanced algorithms.

Memory Flexibility

As a further advantage, the initial target image fits nicely into Stratix's industry-leading TriMatrix™ memory architecture. The different-sized memories allow the customer to make architecture tradeoffs between increasing the number of data ports for overall bandwidth (up to 12 terabits per second), or storing the whole image in a unit of an M-RAM blocks (up to 10 Mbits) to avoid swapping to an external memory.

Small Form Factor

A traditional design would need an FPGA for some data flow massaging, a DSP for processing, RAM, and likely a phase-locked loop (PLL). With Stratix devices, all that discrete functionality is neatly tucked into a single device.

Remote System Configuration

The desire for a standard product with future feature scalability is becoming increasingly necessary. This flexibility allows in-field devices to support new algorithms and standards, feature integration, and revenue streams. The Stratix device family supports this scalability with dedicated support for remote system configuration. A complete solution is achieved by combining Stratix devices with a device to manage the upgrade, such as Blue Iguana's in-system programming gateway. Blue Iguana's firmware code could be loaded onto a Nios® embedded processor inside the Stratix device, and hooked back to Blue Iguana's Secure Service Hub software. Remote system configuration provides enhanced design flexibility and an extended product life.

Conclusion

Altera's Stratix devices allowed Nuvation to meet the tough requirements of this medical imaging application. A combination of the GMACs throughput, future scalability, memory flexibility, system integration, and remote system configuration support distinguish Stratix devices as a true programmable logic powerhouse.

continued on page 24

Stratix DSP Accelerates Medical Imaging, continued from page 23

About Nuvation

Nuvation is a leading electronics design services (EDS) firm that provides ASIC, FPGA, board, and firmware design services for the DSP and embedded systems industry. Nuvation is a member of the Altera CDC and is a certified high-speed design center. Nuvation is an Altera Megafunction Partners Program (AMPP^{SM}) partner with a 10-channel Gigabit Ethernet over OC-192 IP core (GEOS-10). Nuvation's design experience spans the Communications, Consumer Electronics, Medical, and Security/Defense industries. Founded in 1997 and based in San Jose, California, Nuvation's customers include a number of top-tier companies such as Altera, Lucent, Hewlett Packard, JDS Uniphase, Infineon, LeapFrog, SanDisk, Finisar, and Trimble Navigation. Nuvation is a wholly owned subsidiary of IAR Systems. For more information, visit the Nuvation web site at **http://www.nuvation.com**.

Instantiating Memory Using Synopsys FPGA Compiler II Saves Coding Time & Area

The Stratix™ device family is based on a 1.5-V, 0.13-µm, all-layer-copper process technology and offers up to 114,140 logic elements (LEs), 10 Mbits of embedded memory, optimized digital signal processing (DSP) blocks, and high-performance I/O capabilities. Because these devices offer a less expensive and more flexible alternative to ASICs, design engineers are choosing them for prototyping and for use in complex, high-performance systems. As a result, designers are turning to ASIC-like tools and techniques to fully take advantage of all that Stratix devices have to offer.

The preferred method of instantiating Altera memories is to use the MegaWizard Plug-In Manager in the Quartus II software to set up and parameterize a memory megafunction.

Design engineers who use Synopsys' FPGA Compiler II in their design flows have the option of instantiating memory in Stratix devices, in contrast to the more common inference method used by other synthesis tools on the market. With the inference method, designers create a one- or two-dimensional array, and then put values in and out of the array variable. These values will be automatically mapped to memory.

With instantiation, you can indicate that a specific portion of the HDL code will be used for memory, which is then "wired in" as though you are constructing a gate-level design. You might think that instantiation means more work and that it is easier to just code an array. However, inferencing involves more coding because you have to reference the array each time you use memory. In instantiation, Altera supplies the whole declaration, which means that you simply insert the declaration and connect to it.

There are several reasons to instantiate. First, memory allocation is optimized by hand, saving device area. When targeting Stratix devices, it is impossible for the tool to automatically allocate the correct type of memory (M4K, M512, or M-RAM blocks) for the application. Second, memory instantiation allows an easier technology transfer to or from an ASIC. Additionally, there is less typing when instantiating versus memory inference (e.g., when entering VHDL or Verilog HDL).

Using Memory Elements with Altera Devices

The preferred method of instantiating Altera memories is to use the MegaWizard® Plug-In Manager in the Quartus® II software to set up and parameterize a memory megafunction. In the case of Stratix devices, the `altsyncram` megafunction is instantiated by performing the following steps:

1. Use the MegaWizard Plug-In Manager to configure the memory according to your design specifications. The wizard creates three files: a wrapper file that instantiates the megafunction, a file that contains the module declaration, and a file that contains the instantiation declaration. Refer to Quartus II Help for detailed instructions on how to create and instantiate a megafunction for FPGA Compiler II.

2. Create a design file that black-boxes the wrapper module and synthesize it using FPGA Compiler II. Figure 1 shows a Verilog example. In this case, the user file is called **memory_inst.v**. The files created by the MegaWizard Plug-In Manager are called **mymem.v**, **mymem_bb.v**, and **mymem_inst.v**. You can copy and paste the module declaration and instantiation from the **mymem_bb.v** and **mymem_inst.v** files to create the user design file.
3. Synthesize the file you created (**memory_inst.v**) along with the rest of your design in FPGA Compiler II and export the EDIF Output File (**.edf**) after synthesis.
4. Place-and-route the EDIF file in the Quartus II software. Ensure that the wizard-created file and the user-created wrapper exist in the same directory.

Conclusion

Design engineers who follow these simple steps will quickly realize the advantages of instantiation in terms of time, effort, and efficient use of Stratix resources. For more information, visit the Synopsys web site at **http://www.synopsys.com**.

Figure 1. Verilog Example of a User Design File Instantiation & a Wizard-Created Wrapper

User design file: memory_inst.v

```
module memory_inst (
        data_sig,
        wren_sig,
        wraddress_sig,
        rdaddress_sig,
        clock_sig,
        q_sig);

        input   [7:0] data_sig;
        input     wren_sig;
        input   [4:0]  wraddress_sig;
        input   [4:0]  rdaddress_sig;
        input     clock_sig;
        output  [7:0]  q_sig;

        mymem   mymem_inst (
        .data ( data_sig ),
        .wren ( wren_sig ),
        .wraddress ( wraddress_sig ),
        .rdaddress ( rdaddress_sig ),
        .clock ( clock_sig ),
        .q ( q_sig )
        );

        endmodule


        module mymem (
        data,
        wren,
        wraddress,
        rdaddress,
        clock,
        q);

        input   [7:0]  data;
        input    wren;
        input   [4:0]  wraddress;
        input   [4:0]  rdaddress;
        input    clock;
        output  [7:0]  q;

endmodule
```

Megawizard created file (edited for space): mymem.v

```
module mymem (
        data,
        wren,
        wraddress,
        rdaddress,
        clock,
        q);
        input   [7:0]  data;
        input     wren;
        input   [4:0]  wraddress;
        input   [4:0]  rdaddress;
        input     clock;
        output  [7:0]  q;
        wire [7:0] sub_wire0;
        wire [7:0] q = sub_wire0[7:0];
        altsyncram   altsyncram_component (
                     .wren_a (wren),
                     .clock0 (clock),
                     .address_a (wraddress),
                     .address_b (rdaddress),
                     .data_a (data),
                     .q_b (sub_wire0));
        defparam
          altsyncram_component.operation_mode = "DUAL_PORT",
          altsyncram_component.width_a = 8,
          altsyncram_component.widthad_a = 5,
          altsyncram_component.numwords_a = 32,
          altsyncram_component.width_b = 8,
          altsyncram_component.widthad_b = 5,
          altsyncram_component.numwords_b = 32,
          altsyncram_component.lpm_type = "altsyncram",
          altsyncram_component.width_byteena_a = 1,
          altsyncram_component.outdata_reg_b = "UNREGISTERED",
          altsyncram_component.indata_aclr_a = "NONE",
          altsyncram_component.wrcontrol_aclr_a = "NONE",
          altsyncram_component.address_aclr_a = "NONE",
          altsyncram_component.address_reg_b = "CLOCK0",
          altsyncram_component.address_aclr_b = "NONE",
          altsyncram_component.outdata_aclr_b = "NONE",
          altsyncram_component.read_during_write_mode_mixed_ports = "DONT_CARE",
          altsyncram_component.ram_block_type = "AUTO",
          altsyncram_component.intended_device_family = "Stratix";
endmodule
```

Controlling I/O Register Mapping For Altera Devices

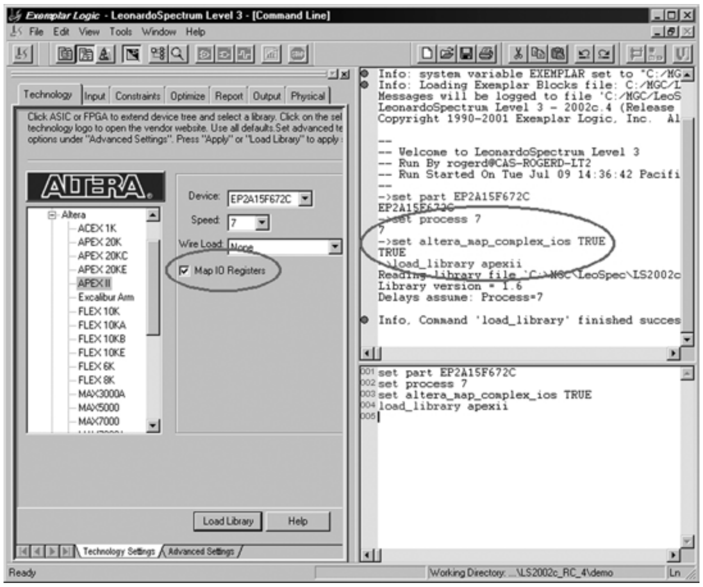
by Roger Do,
Mentor Graphics

Altera's look-up table (LUT)-based series of devices contain I/O structures that include input, output, and control signal registers. To adequately meet timing requirements on and off chip, these registers must be implemented. Mentor Graphics® synthesis software fully supports the use of all the I/O registers for Stratix™, APEX™, and Mercury™ devices. APEX 20K and APEX 20KE devices have only a single I/O register while the Stratix, APEX II, and Mercury devices have input, output, and control signal registers.

How Synthesis Works

The default configuration for Precision Synthesis™ and LeonardoSpectrum™ does not map to any I/O registers. For mapping I/O registers, you can implement I/O registers globally or individually. When you globally implement I/O registers, all input and output registers are mapped to the I/O atom. For APEX 20K and APEX 20KE devices, if a bidirectional I/O pin has both input and output registers, the input register is mapped to the I/O atom and the output register is mapped to an internal LCELL atom.

Figure 1. Global I/O Register Mapping



Exemplar Logic - LeonardoSpectrum Level 3 - [Command Line]

Click ASIC or FPGA to extend device tree and select a library. Click on the sel technology logo to open the vendor website. Use all defaults.Set advanced te options under "Advanced Settings". Press "Apply" or "Load Library" to apply :

- Altera
  - ACEX 1K
  - APEX 20K
  - APEX 20KC
  - APEX 20KE
  - APEX II
  - Excalibur Arm
  - FLEX 10K
  - FLEX 10KA
  - FLEX 10KB
  - FLEX 10KE
  - FLEX 6K
  - FLEX 8K
  - MAX3000A
  - MAX5000
  - MAX7000

Device: EP2A15F672C
Speed: 7
Wire Load: None
☑ Map IO Registers

```
Info: system variable EXEMPLAR set to 'C:/MG
Info: Loading Exemplar Blocks file: C:/MGC/L
Messages will be logged to file 'C:/MGC/LeoS
LeonardoSpectrum Level 3 - 2002c.4 (Release
Copyright 1990-2001 Exemplar Logic, Inc.  Al

--
-- Welcome to LeonardoSpectrum Level 3
-- Run By rogerd@CAS-ROGERD-LT2
-- Run Started On Tue Jul 09 14:36:42 Pacifi
--
->set part EP2A15F672C
EP2A15F672C
->set process 7
7
->set altera_map_complex_ios TRUE
TRUE
->load_library apexii
Reading library file 'C:\MGC\LeoSpec\LS2002c
Library version = 1.6
Delays assume: Process=7

Info, Command 'load_library' finished succes
```

```
set part EP2A15F672C
set process 7
set altera_map_complex_ios TRUE
load_library apexii
```

Working Directory: ...\LS2002c_RC_4\demo

Considerations Before Synthesis

Deciding to globally map to I/O registers affects the overall design performance. Since the I/O registers are moved to the peripheral of the device, these registers may or may not be placed in close proximity. Therefore, the internal register-to-register performance of the design can suffer because of the placement of the I/O registers with respect to the logic that it is driving. Altera recommends you only use I/O register mapping when external I/O timing requirements dictate I/O register use.

Controlling I/O Mapping Globally

You can perform global I/O register mapping through either the graphical user interface (GUI) or an attribute. To select global I/O register mapping from the LeonardoSpectrum GUI, go to the **Technology Flow** tab and click **Map IO Registers**. When the Altera library is loaded, the attribute `altera_map_complex_ios` will be set to `TRUE`. This attribute can also be set to `TRUE` at the command line or from tool command language (Tcl) scripts anytime before optimization to enable global I/O register mapping. See Figure 1.

The following is a command-line example:

```
set altera_map_complex_ios TRUE
```

Mapping Individual I/O Registers

You can control individual I/O register mapping through attributes placed on individual signals either through the HDL code or at the command line. The attribute that controls I/O register mapping is `map_complex`. The `map_complex` attribute overrides the mapping directive that is set globally with the `altera_map_complex_ios` variable. In one situation, you might want to set the `map_complex` attribute to `TRUE` on a few I/O ports that you want to map to the I/O atom. In another situation, you might want to turn on global mapping for all I/O ports by setting altera_map_complex_ios to TRUE, then turn off complex mapping for selected ports by setting the map_complex attribute to FALSE. Figure 2 shows a command-line example.

Figure 2. Mapping Individual I/O Registers Example

```
usage: set_attribute -<obj_type> <obj_name> -name <attribute_name> -value <attribute_value>
set_attribute -port inp -name map_complex -value TRUE
```

VHDL Example:

```
entity example is
port (inp, clk : in std_logic;
outp : out std_logic);

--usage:
--The following syntax can be used for the declaration of a VHDL attribute:
--attribute <attribute_name> : <attribute_type>;
--The following syntax describes how to set an attribute on a VHDL component.
--Set on component that corresponds to view.
--attribute <attribute_name> of <object_name> : component is
--<attribute_value>
--Set on a label that corresponds to an instance.
--attribute <attribute_name> of <object_name> : label is
--<attribute_value>

attribute map_complex: string ;
attribute map_complex of inp:signal is "TRUE";

end example;
```

Verilog Example:

```
//Usage:
//synthesis attribute <object_name> <attribute_name> <attribute_value>

//example
module example (inp, clk, outp);
input inp, clk;
output outp;
.
.
.
// synthesis attribute inp map_complex TRUE
endmodule
```

El Camino's Design Services Provide Solutions for Incremental Encoder Emulation

Albert Lindmeier,
Director of Engineering,
El Camino GmbH

With a strong background in digital signal processing (DSP) designs, El Camino contributed its highly qualified design knowledge to Baumueller Anlagen-Systemtechnik, a company that specializes in distributed drive technology, specifically for publishing and printing applications. For their new generation of drive controls, El Camino focused its resources on providing a solution which includes an Altera® ACEX® 1K EP1K30 device.

Incremental Encoders

For their new generation of drive controls, El Camino focused its resources on providing a solution which includes an Altera ACEX 1K EP1K30 device.

Contemporary motor drives for printing machines often use the concept of virtual axes, in which remote drives are synchronized to each other by means of a control algorithm (behaving as if they were connected physically). Usually, you can achieve this synchronization by using a real-time bus system and a digital signal processor built into the drives. The tracking performance depends mainly on the update rate of the position data and the algorithm.

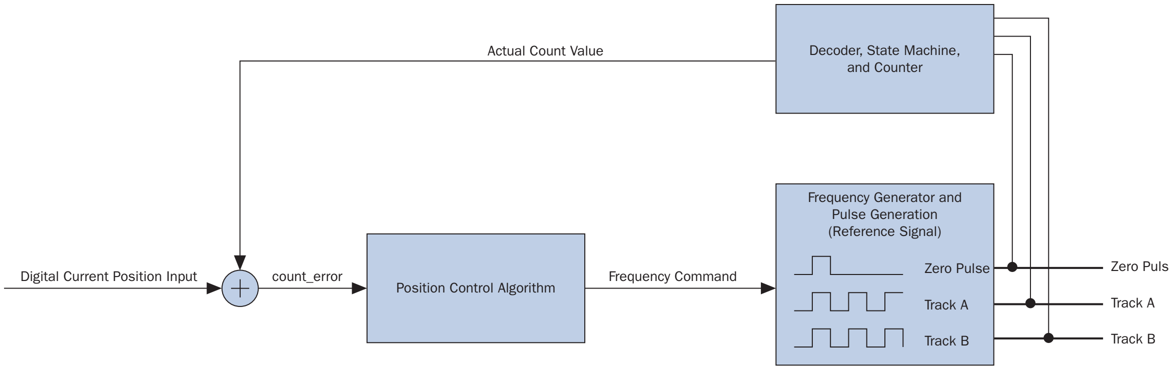
However, there is still a lot of auxiliary equipment (e.g., paper cutters or paper-folding equipment) that must receive the input signals from a traditional incremental encoder. Incremental encoders have two tracks (A and B) that have a phase shift of 90 degrees and one reference pulse (zero-pulse) to achieve the proper start condition. You can derive the direction of the rotation from the time relation of track A to track B. For example, if track A is leading track B, the rotation is clockwise and vice versa.

The most obvious way to generate such signals is to add an incremental encoder to the axis, but this is not always possible or desirable because of mounting problems, reliability, and cost.

Existing Solution: Incremental Encoder Emulation

Running an incremental encoder emulation on a digital signal processor is the existing solution. This solution, as in most cases, is the same digital signal processor as the drive control with additional external counters for speed considerations. A position control loop within the digital signal processor controls the frequency of the oscillator. You can generate the signals for tracks A and B out of this oscillator frequency and the direction information. A state machine and a counter evaluates these track signals and the resulting counter value, which represents the actual generated number of output pulses (which can differ from the actual change in position due to response time or frequency errors). This value is another input to the position control loop. An error signal is generated out of the counter value and the actual position value, which in turn controls the oscillator to minimize the error.

Figure 1. Tracking Loop Basic Block Diagram


Actual Count Value
Decoder, State Machine, and Counter
Frequency Generator and Pulse Generation (Reference Signal)
Digital Current Position Input
count_error
Position Control Algorithm
Frequency Command
Zero Pulse
Track A
Track B
Zero Pulse
Track A
Track B

The type of emulation described above is known as incremental encoder emulation with tracking loop. Figure 1 shows a basic block diagram of this solution.

Despite the fact that the control algorithm is refined by adding features like position prediction or similar logic, it is common that the actual number of track pulses A and B does not always represent the actual position. This fact is especially true if the drive changes its speed (acceleration or deceleration). The speed of the control loop is also limited by the available computation power of the digital signal processor.

New Solution: Using Altera FPGAs

The new solution does not use a tracking loop. The track pulses A and B are directly synthesized from the incoming position information. You can view this solution as a virtual axis with a high-resolution angle accumulator. From this angle accumulator, you can derive the signals for the tracks and the zero pulse.

However, due to the high resolution of the counter and the desired rotation speed of the drive, this accumulation must execute very quickly (in this application, 160 ns). You could not achieve this quick execution with a digital signal processor. Furthermore, you have to perform several parallel tasks within that 160-ns cycle time. Therefore, an Altera ACEX EP1K30 device was selected to accomplish the task. Using an Altera FPGA, you are able to vary your program to fit different solutions in terms of pulses per revolution. This programmability allows you to emulate a whole range of incremental encoders.

The digital signal processor writes a new current position value to the FPGA every 0.2 to 2 ms, depending on the application.

The algorithm calculates the difference of the incoming current position value and the internal virtual position value. The resulting difference is divided by *k* (the ratio of sample time [T_a] to updated time [T_k]) and split into an integer and the remainder part. By integration of the integer part of Δl, you get a high-resolution representation of the current position, updated every cycle time (in this case, 160 ns), including an accumulated remainder. Every time it exceeds one, the one is added to the integer part, resulting in no pulse loss.

In contrast to the tracking solution, this new approach makes a linear interpolation of the angle values, which immediately follow changes in the input value, and not after the latency time T_a as in the tracking solution.

The high-angular resolution of the virtual axis allows a more accurate setting of the zero-reference pulse. The track pulses A and B are directly derived from the virtual axis value. Depending on the register-settable number of pulses per turn (lines per revolution of a real incremental encoder), the track pulses A and B are generated at given positions of the virtual axis.

When there is no movement of the axis, this technique also suppresses any high-frequency toggling of the track pulses, which would be a problem with the tracking solution.

The FPGA contains multipliers, accumulators, limiters, a divider, and additional logic. Most of the processing must be done in parallel, which is not possible in a DSP solution. Therefore, the Altera EP1K30 device made it possible to use a high-performance solution, which would otherwise not be possible.

Using an Altera FPGA, you are able to vary your program to fit different solutions in terms of pulses per revolution. This programmability allows you to emulate a whole range of incremental encoders.

About El Camino

El Camino GmbH was founded in 1999 and is a fast-growing competence center focusing on design, training, and consultation related to PLDs. With its background in DSP design, El Camino GmbH is a key partner in implementing innovative DSP designs in programmable logic. For more information on El Camino GmbH and its products and services, visit their web site at **http://www.elca.de**.

About Baumueller Anlagen-Systemtechnik GmbH & Company

Baumueller Anlagen Systemtechnik are experts in distributed drive technology and offer complete solutions for publishing and printing, and advertising and package printing. Baumueller Anlagen Systemtechnik is a reliable partner with many years of experience in the fields of process engineering, automation, drive technology, and open- and closed-loop control engineering.

Effective Programmable Logic Architecture for CDMA/W-CDMA Matched Filter

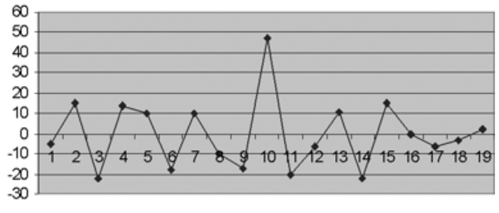
Code division multiple access (CDMA) and the newer wideband code division multiple access (W-CDMA) are wireless communication standards. In both standards, different users are using the same frequency spectrum and, to distinguish one user from the other, a different code sequence is used for each user. You can use matched filters to identify the different code sequences. The matched filter is a correlator structure. You can use similar correlator structures for random access channel (RACH) detectors. Matched filters or RACH detectors typically require high-speed computation power. Depending on the parameters, the matched filter or the RACH receiver can require execution of tens of billions or hundreds of billions of operations per second. This high-speed computational power requirement is clearly beyond the capability of a single digital signal processor. FPGAs can achieve this level of computational power and still maintain the high level of flexibility required to support different variants of these applications.

FPGA architecture based on distributed memory is used in some of those applications to create a high-speed parallel processing architecture. Distributed memory architecture is based on logic cell RAM elements that can store 16 bits each and can be used as a 16-bit shift register. This article describes another FPGA architecture based on (parallel samples, parallel coefficients, and time division multiplexing (TDM)) (PPT) array calculations and memory blocks that can achieve even higher levels of cost efficiency for those applications.

Correlator Function

Matched filters and RACH detectors are implemented as correlators. The correlator searches for a code sequence (sometimes multiple code sequences) embedded in the received signal by comparing the received signal with a copy of the code sequence. The code sequence is a sequence of +1 or -1 coefficients. Figure 1 shows a correlator example.

Figure 1. Correlator Example Graph



Received Samples	5	2	-1	3	6	-6	3	-3	5	9	-8	-8	7	-6	0	3	-2	-6	-5	4
Code Sequence	1	-1	-1	1	-1	1				x										
Correlation Output	-5	15	-22	14	10	-18	10	-10	-17	47	-20	-6	11	-22	2					

The correlator slides the code sequence to the right of the received samples and searches for one of the correlation points that has the maximum correlation value. The correlation value is calculated as a sum of multiplication coefficients and samples similar to a finite impulse response (FIR) filter structure. The maximum correlation result 47 is found in point number 10, as shown in Figure 1.

Parallel Samples, Parallel Coefficients & (PPT) Correlator Architecture

In each clock, an array of *n*d* (*n* = number of samples processed together, *d* = number of correlation points calculated together) correlation nodes is processed by the PPT correlator. In the next clock, a subsequent array of *n*d* correlation nodes (to the right of the current *n*d* correlation nodes in Figure 3) is processed. After *L*/*n* clocks the correlation calculation, the current *d* starting nodes is completed. Later, a new correlation calculation for shifted correlation sequences is started *d* nodes to the right of the previous starting point. Each one of the horizontal lines in Figure 3 represents a shifted sequence of *L* samples. The number of samples processed together (*n*) of coefficients (FFs) are driving in parallel the array of correlation nodes. For (*n*+*d*-1) samples, each number of samples (*b* bits in width) are driving in parallel the array of correlation nodes.

The samples and coefficients are coming from internal block memories (embedded system blocks (ESBs) for APEX™ devices or M4K blocks for Stratix™ devices). They are not shifted into the array, and there is no latency issue for context switching. It is possible to switch to the next *d*-shifted, correlation sequence calculation, or switch to another code sequence calculation immediately to the next clock. Once the PPT correlator completes processing (a group of shifted correlation sequences), it loads the output matched results to an output shift register, and can start processing the next group of shifted correlation sequences immediately to the next clock.

Each group of *n* multiplier (implemented as an exclusive or gate of *b* samples and coefficients plus the sign bit for two's compliment representation) outputs feed the adder tree, as shown in Figure 4. On each clock, a new group of *n* nodes along the horizontal line of Figure 3 feeds the inputs of the same adder tree. The last adder of the adder tree sums all the intermediate results. After the (*L*/*n*) clock sequence, the final correlation results are latched in the output-matched result register. The next clock after the final correlation results is latched into the matched result register, a new set of *d* correlation points begins calculation. In parallel, the matched result registers of all the *d* adder trees (connected to each other as a shift register) will shift the results clock after the clock to the correlator block output.

The PPT correlator has some features of traditional FIR correlators (e.g., adder trees for the multiplication results), and has some features of inverse FIR correlators (e.g., data samples driving multiple nodes in parallel). It is significantly different from both of them since the samples and the coefficients are driving a two-dimensional array of nodes and are stored in a block memory.

To estimate the PPT correlator size, see equation (1):

$$N_{\text{LEs}} = n + b*(n+d\text{-}1) + d*(2*(\log_2 L + b) + \sum_{i=1}^{\log_2 n} \frac{M}{2^i}(b+i)) \quad (1)$$

Figure 3. PPT Correlator Architecture

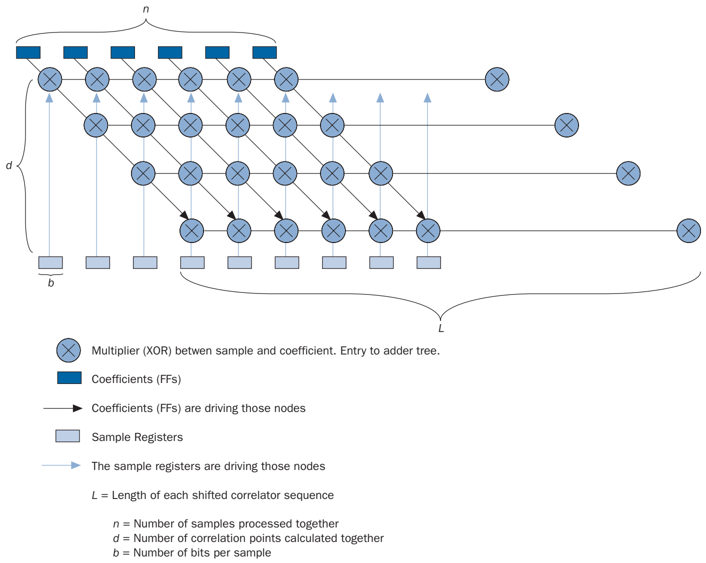
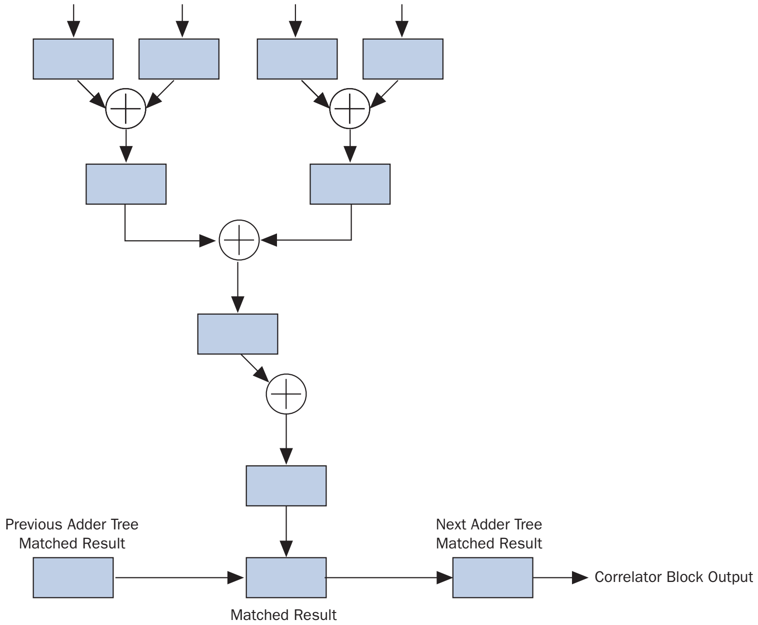

n
d
b
L
Multiplier (XOR) betwen sample and coefficient. Entry to adder tree.
Coefficients (FFs)
Coefficients (FFs) are driving those nodes
Sample Registers
The sample registers are driving those nodes
L = Length of each shifted correlator sequence
n = Number of samples processed together
d = Number of correlation points calculated together
b = Number of bits per sample

Figure 4. PPT Correlator Adder Tree



To estimate the distributed memory correlator size based on the traditional FIR architecture, see equation (2):

$$N_{\text{LEs}} = \text{L}*(b+1)/16 + \sum_{i=1}^{\log_2 n} \frac{M}{2^i}(b+i) \quad (2)$$

M = L/16 for clock rate/chip rate = 16
M = L/32 for clock rate/chip rate = 32

continued on page 34

Effective Programmable Logic Architecture for CDMA/W-CDMA Matched Filter, continued from page 31

This size estimation (equations 1 and 2) does not include the control logic for the memory and the timing logic for the correlator. It also does not include variations in size when you use different synthesis tools. You can assume that the control and timing logic is significantly smaller than the correlator logic.

To calculate the memory size, see equation (3):

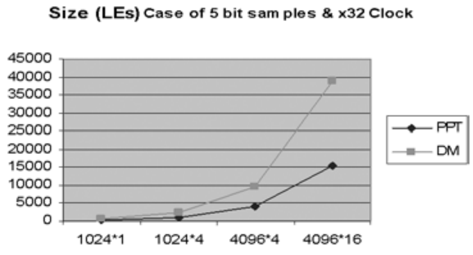
$$Memory_{bits} = L*(b+1)*\text{Number of Channels}*\text{Oversampling} \quad (3)$$

Benchmark PPT vs. Distributed Memory

The Altera PPT correlator architecture is more cost effective (same functionality and performance with smaller size) than the competition's distributed memory architecture.

Figures 5 and 6 plot the resource usage efficiency of Altera's PPT architecture versus the competition's distributed memory architecture.

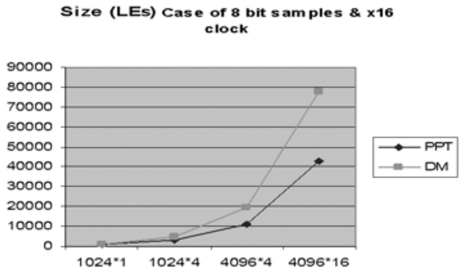
Figure 5. Benchmark 1 (LEs only) *(1)*



Note to Figure 5:

(1) Sample (5 bits), clock (122.88 MHz), sequence length (1024, 4096), number of channels (1, 4, 16)

Figure 6. Benchmark 2 (LEs only) *(1)*



Note to Figure 6:

(1) Sample (8 bits), clock (61.44 MHz), sequence length (1024, 4096), number of channels (1 , 4, 16)

In all the data points covered by this benchmarking, the PPT architecture shows better results (smaller size for same functionality and performance).

Figures 5 and 6 display a clear trend. The PPT architectural advantage increases when there is the following:

- Higher computation load [(computation load = sequence length)*(number of bits per sample)*(oversampling)*(number of channels (user codes))*(number of antennas)].
- Higher ratio of clock-to-chip rate.

Figures 5 and 6 show that the clock rates are (16*chip rate) and (32*chip rate). Those comparison points are sweet spots for the competition's distributed memory architecture, which compresses 16 bits of shift register into a single logic cell. If the maximum possible clock frequency is not an integer multiplication of (16*chip rate), the comparison point will be more friendly to the Altera PPT architecture which is more flexible and takes advantage of a higher clock frequencies.

Conclusion

The Altera PPT correlator architecture is more cost effective (same functionality and performance with smaller size) than the competition's distributed memory architecture. Techniques of parallel processing, TDM, and efficient data flow are used to achieve this level of cost efficiency.

Quartus II Verilog HDL & VHDL Integrated Synthesis

In addition to supporting third-party synthesis flows from Mentor Graphics®, Synopsys, and Synplicity®, the Altera® Quartus® II software version 2.1 introduces improved integrated synthesis that fully supports the Verilog HDL and VHDL languages, and provides synthesis options to control the synthesis process. Integrated synthesis is also included in the Quartus II Web Edition software. With this synthesis support, the Quartus II software provides a complete, easy-to-use, standalone solution for system-on-a-programmable-chip (SOPC) designs.

Improved Verilog HDL & VHDL Support

The Quartus II software's integrated synthesis now fully supports Verilog HDL and VHDL synthesizable language features, as well as certain compiler directives. In addition to the Altera schematic Block Design File (**.bdf**) and AHDL Text Design File (**.tdf**) formats, the Quartus II Compiler's Logic Synthesizer module supports the following language standards:

- Verilog-1995 (IEEE Std. 1364-1995)
- Verilog-2001 (IEEE Std. 1364-2001) (new, partial support)
- VHDL 1987 (IEEE Std. 1076-1987)
- VHDL 1993 (IEEE Std. 1076-1993)

You can select which standard to use with an option in the **Verilog HDL Input** or **VHDL Input** tab of the **General Settings** dialog box (Project menu). The Compiler uses Verilog-2001 and VHDL 1993 by default. For full information on specific syntax features and language constructs, refer to the "Quartus II Verilog HDL Support" and "Quartus II VHDL Support" topics in Quartus II Help.

New Features in Verilog HDL Synthesis

Verilog-2001 support is a new feature in the Quartus II software version 2.1 and higher. Examples of supported constructs include:

- Generate statements
- Preprocessor statements such as `‘elsif`, `‘line`, and `‘file`
- Signed declarations for all variables
- Operators such as `**` and `<<<`

In version 2.1 and higher, the Quartus II software supports case-sensitive Verilog HDL code in accordance with the Verilog HDL standard. Older versions of the Quartus II software did not support case-sensitive module names.

New Features in VHDL Synthesis

The Quartus II software version 2.1 and later supports VHDL libraries differently from previous versions of the Quartus II or MAX+PLUS® II software, and now conforms to the VHDL standard.

In version 2.1 and later, you can call standard IEEE and vendor VHDL libraries and packages from VHDL code from within the Quartus II software. For a complete listing of library and package support, refer to the "Using Quartus II Packages" topic in Quartus II Help. The IEEE library includes the standard VHDL packages **std_logic_1164**, **numeric_std**, and **numeric_bit**. The STD library is part of the VHDL language standard and includes **standard** packages (included in every project by default) and **textio**. For compatibility with older designs, the Quartus II software also supports vendor-specific packages and libraries, including:

- Synopsys packages such as **std_logic_arith** and **std_logic_unsigned** in the IEEE library
- Mentor Graphics packages such as **std_logic_arith** in the ARITHMETIC library
- Altera packages such as **maxplus2**, **altera_mf_components**, and **lpm_components** in the ALTERA library

In previous versions of the Quartus II software, you declared precompiled libraries in the **VHDL Input** tab of the **General Settings** dialog box (Project menu). To call a user-defined VHDL package in version 2.1 and later, indicate the library and package name using the `LIBRARY` and `USE` commands. You can use any name for your library, including work; therefore, you can use version 2.1 and later to compile projects that you created in earlier versions of Altera software that used precompiled user-defined libraries without any code modifications. To do so, simply

continued on page 34

Quartus II Verilog HDL & VHDL Integrated Synthesis, continued from page 33

include your VHDL package in your Quartus II project using the **Add Files** tab of the **General Settings** dialog box (Project menu). The package must be listed before other files that use the package because it must be analyzed first by the Quartus II Compiler.

Compiler Directives

The Quartus II software now supports certain Compiler directives, also called pragmas or attributes. The user includes Compiler directives in Verilog HDL or VHDL code as comments. These directives are not Verilog HDL or VHDL commands; however, synthesis tools use the commands to drive the synthesis process in a particular manner.

The `translate_on` and `translate_off` directives indicate whether the Quartus II software or a third-party synthesis tool should compile a portion of the design file. For example, you might use this directive to indicate a portion of code that is intended for simulation only. The synthesis tool reads synthesis-specific directives and processes them during synthesis; however, third-party simulation tools read them as comments and ignore them. The `translate_off` directive indicates the beginning of code that the synthesis tool should ignore; the `translate_on` directive indicates that synthesis should resume.

Figure 1 shows the different translate compiler directives that are supported in the Quartus II software.

Figure 1. Translate Compiler Directives

Verilog HDL:

```
// synthesis translate_off
// synthesis translate_on
// synopsys translate_off
// synopsys translate_on
// exemplar translate_off
// exemplar  translate_on
```

VHDL:

```
-- pragma translate_off
-- pragma translate_on
```

Quartus II Synthesis Options

The Quartus II software provides a number of logic options that control synthesis. Refer to Quartus II Help for more information on how to use logic options and which options are available. You can use the following logic options to change how the compiler synthesizes and optimizes your design:

- *Optimization Technique*: This option specifies the overall goal for logic optimization, (i.e., whether to attempt to achieve maximum speed performance or minimum area usage during compilation). The default setting varies by target device family, and is generally optimized to get the best area/speed trade-off.
- *Preserve Hierarchical Boundary*: This option determines how strictly the hierarchical boundaries between design entities should be maintained during logic synthesis.
- *Power-Up Don't Care*: This option causes registers to power up with a "don't care" logic level (X), or the logic level most appropriate for the design. You can use this option to allow the Compiler to change the power-up level of a register to minimize your design's area usage.
- *Remove Duplicate Logic* or *Remove Duplicate Registers*: If you turn on one of these two options, the Compiler removes logic or registers respectively if they are identical to others in the design, minimizing area utilization. If two functions generate the same logic or registers, the Compiler removes the second one and the first one fans out to the second one's destinations.

Conclusion

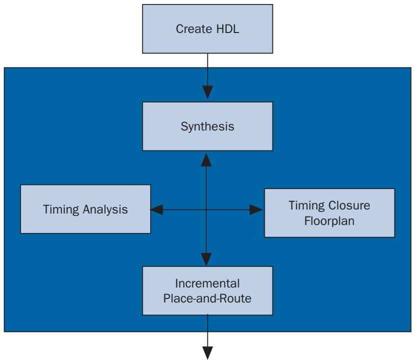
The Quartus II software includes full Verilog HDL and VHDL synthesis support, making it a complete easy-to-use stand-alone solution for SOPC designs. For more information on Quartus II integrated synthesis, refer to *AN 238: Using Quartus II Verilog HDL & VHDL Integrated Synthesis* (available in the Literature section of the Altera web site at **http://www.altera.com**).

Quartus II & the New Timing Closure Methodology

Designers now face the difficult task of achieving timing closure in designs while still maintaining the significant time-to-market advantage that FPGAs offer. The Altera® Quartus® II software version 2.1 introduces a new timing closure methodology to address this new dilemma with FPGA designs.

Altera delivers a timing closure methodology as an integrated part of its existing tool suite at no additional cost. As shown in Figure 1, the Quartus II timing closure methodology enables close interaction between synthesis, timing analysis, floorplan editing, and place-and-route processes to easily meet complex timing requirements. The Quartus II software now provides designers with a new timing closure flow and new timing closure features, including netlist optimization options and a new Timing Closure floorplan. This new floorplan allows designers to achieve timing closure on designs faster and with fewer iterations.

Figure 1. Timing Closure Methodology



Timing Closure Flow

A traditional flow for designs using FPGA tools is to enter constraints, synthesize the design, and then perform place-and-route. The Quartus II software introduces features that allow you to more effectively close timing, including netlist optimization, a new Timing Closure floorplan, and more powerful user assignments. The Quartus II timing closure flow also offers more control over the synthesis and place-and-route process. You can now use fitter information for more efficient synthesis. Figure 2 on page 36 shows the Quartus II timing closure flow diagram.

Netlist Optimization

The Quartus II software now includes push-button netlist optimization options to further optimize your design after synthesis and before place-and-route (designs average a 10% increase). You can apply these options regardless of the synthesis tool used. Three netlist optimization options are offered. You can use these options in different combinations to provide the best results.

WYSIWYG Primitive Resynthesis

The *WYSIWYG Primitive Resynthesis* option unmaps logic element (LE) primitives from a third-party atom netlist to gates and then remaps the gates back to Altera-specific primitives. This feature uses different architecturally-aware techniques during the remapping process that can achieve significant performance improvements.

Gate-Level Register ReTiming

The *Gate-Level Register ReTiming* option enables movement of registers across combinational logic to balance timing, allowing the Quartus II software to trade off the delay between critical paths and non-critical paths.

Logic Element Duplication

After place-and-route, placement information and routing delay information give an accurate indication as to whether or not logic duplication will increase the performance of a design. The LE duplication option allows you to automatically duplicate LEs based on this information. You can duplicate an LE that fans out to multiple locations to reduce the delay of one path without degrading the delay of another.

Timing Closure Floorplan

The Quartus II software introduced a Timing Closure floorplan to help you interactively analyze designs and make assignments. This new

continued on page 38

Quartus II & the New Timing Closure Methodology, continued from page 35

Figure 2. Timing Closure Flow

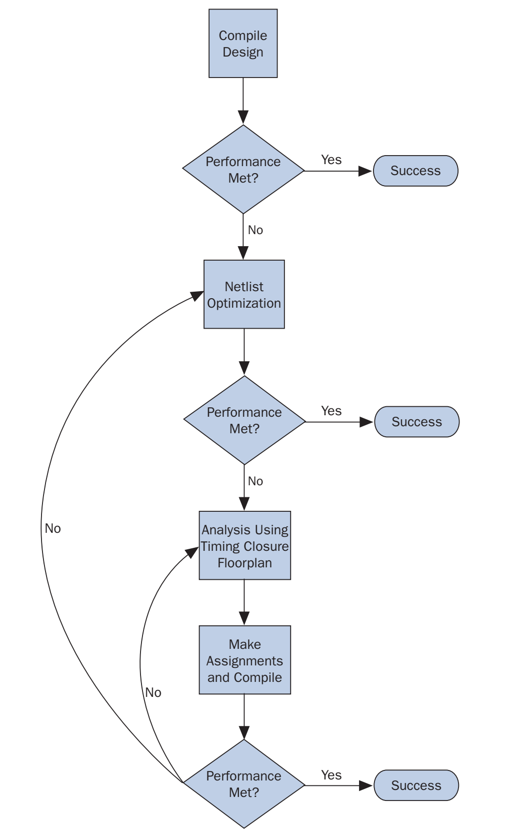
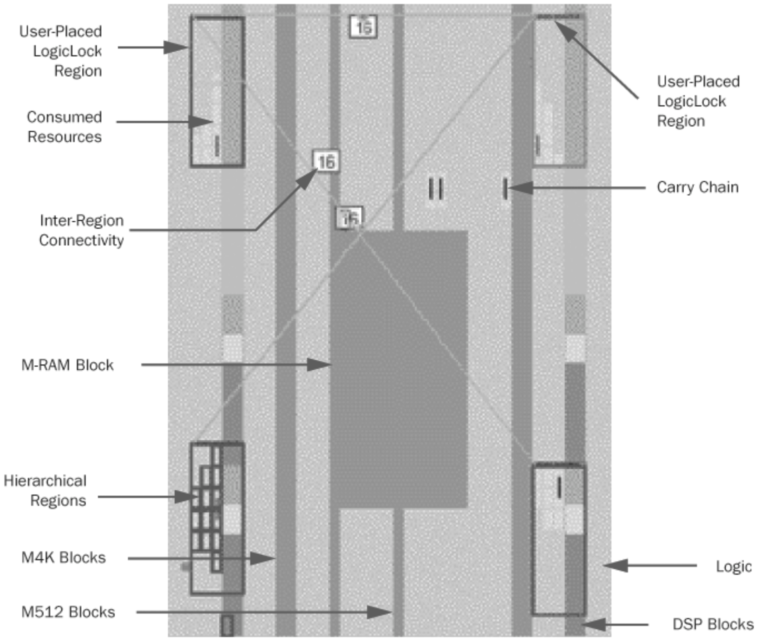


Figure 3. Timing Closure Floorplan



floorplan, used in conjunction with traditional Quartus II timing analysis features, provides a powerful method to perform design analysis.

The Timing Closure floorplan (see Figure 3) allows you to see user assignments and fitter placements simultaneously. It also introduces a number of features to aid in the analysis of designs.

Physical Timing Estimates

The *Physical Timing Estimates* option allows designers to interactively calculate the probable delay from a source to a destination. Once a resource is selected, all other resources on the floorplan are shaded a different color corresponding to the delay from the selected resource. You can also determine the delay in nanoseconds from a resource to any other specific resource, and the delay within a LogicLock™ region.

LogicLock Region Connectivity

The *LogicLock Region Connectivity* option shows how LogicLock regions interface by viewing the connectivity between regions. This feature allows you to see how nodes assigned to LogicLock regions interact with nodes assigned to other LogicLock regions.

Critical Paths

It is now possible to display paths in the floorplan based on criticality, which allows you to see where problem paths are based on the location of their nodes in the floorplan. You can view critical f_{MAX}, clock-to-out, and set-up paths. In addition to the features above, there have also been changes to how assignments can be made. To provide a more complete timing closure solution, it is now possible to make path-based assignments. The path-based assignments feature allows you to make placement assignments directly to a failing path from within the timing analysis tool, the Timing Closure floorplan, or a new path window.

Conclusion

The new Quartus II timing closure methodology provides significant benefits over traditional FPGA design flows. For more information on the timing closure methodology, refer to *AN 198: Timing Closure with the Quartus II Software.*

Discontinued Devices Update

Altera will be obsoleting select devices from product-term and FPGA families (see Table 1). Most of the devices will have longer-than-usual last-time buy (18 months) and last-time ship dates (an additional 6 months) to allow customers to gradually transition to using alternative ordering codes.

Select ordering codes from mature families such as MAX® 7000S are being obsoleted to increase the operational efficiency in the manufacturing flow. On mainstream product families such as the MAX 7000A, FLEX® 10KE, APEX™ 20K and newer product families, such as the MAX 7000B, ACEX® 1K, and APEX 20KE, ordering codes have been consolidated to offer a limited set of codes that will cover the various package and speed grade options.

Continued support for devices beyond the phase-out period may be available through Rochester Electronics, an extended after-market supplier. For more information, contact Rochester Electronics at (508) 462-9332 or your local Altera sales office.

Continued support for devices beyond the phase out period may be available through Rochester Electronics, an extended after-market supplier.

Table 1. Discontinued Device Update

Product Family	Device	Last Order Date	Last Shipment Date	Reference
MAX 7000	Selected devices with a dry pack option	05/31/02	11/30/02	PDN 0117
	EPM7192EGM160-15	05/31/02	11/30/02	PDN 0117
MAX 7000S	Selected devices in plastic quad flat pack (PQFP) and thin quad flat pack (TQFP) packages	05/31/02	11/30/02	PDN 0117
	Selected devices with fixed-pulse width programming option	10/31/03	04/30/04	PDN 0203
MAX 7000A	Selected devices in micro ball-grid array (BGA), FineLine BGA™, and TQFP packages	10/31/03	04/30/04	PDN 0203
MAX 7000B	Selected devices	10/31/03	04/30/04	PDN 0203
MAX 9000	EPM9560BC356-20	05/31/02	11/31/02	PDN 0117
	EPM9320RC208-15F	05/31/02	11/31/02	PDN 0117
FLEX 8000	Selected FLEX 8000 pin-grid array (PGA) packages	02/28/02	08/31/02	PDN 0107
	Selected devices	02/28/03	08/31/03	PDN 0107
	EPF8820ABC225-3	05/31/02	11/31/02	PDN 0117
FLEX 10K	Selected FLEX 10K PGA packages	02/28/02	08/31/02	PDN 0107
	Selected FLEX 10K PGA & BGA packages	02/28/03	08/31/03	PDN 0107
	EPF10K10LI84-4	05/31/02	11/31/02	PDN 0117
FLEX 10KA	Selected PGA packages	02/28/03	08/31/03	PDN 0107
	EPF10K30AFI256-2	05/31/02	11/31/02	PDN 0117
FLEX 10KE	Selected PGA packages	02/28/03	08/31/03	PDN 0107
	Selected devices	10/31/03	04/30/04	PDN 0204
APEX 20K	Selected PGA packages	02/28/03	08/31/03	PDN 0107
	EP20K400BI652-2	05/31/02	11/31/02	PDN 0117
	Selected devices	10/31/03	04/30/04	PDN 0204
APEX 20KE	Selected devices	10/31/03	04/30/04	PDN 0204
ACEX 1K	Selected devices	10/31/03	04/30/04	PDN 0204

How to Contact Altera

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

Information Type	U.S. & Canada	All Other Locations
Product Literature	**http://www.altera.com**	**http://www.altera.com**
Altera Literature Services *(1)*	**lit_req@altera.com**	**lit_req@altera.com**
News & Views Information	**http://www.altera.com/literature/nview.html** **n_v@altera.com**	**http://www.altera.com/literature/nview.html** **n_v@altera.com**
Non-Technical Customer Service	(800) 767-3753	(408) 544-7000
Technical Support	**https://www.altera.com/mysupport**	**https://www.altera.com/mysupport**
	(408) 544-6401	(408) 544-6401 *(2)*
FTP Site	**ftp.altera.com**	**ftp.altera.com**
General Product Information	(408) 544-7104	(408) 544-7104 *(2)*
	http://www.altera.com	**http://www.altera.com**

Notes:

(1) The ***Quartus Installation and Licensing*** and ***MAX+PLUS II Getting Started*** manuals are available from the Altera® web site. To obtain other MAX+PLUS® II software manuals, contact your local distributor.

(2) You can also contact your local Altera sales office or sales representative. See the Altera web site for the latest listing.