News & Views
Fourth Quarter 2001
Newsletter for Altera Customers

ALTERA®
The Programmable Solutions Company®

Altera Presents HardCopy Devices—The Low-Risk, Low-Cost Solution for High-Density PLDs

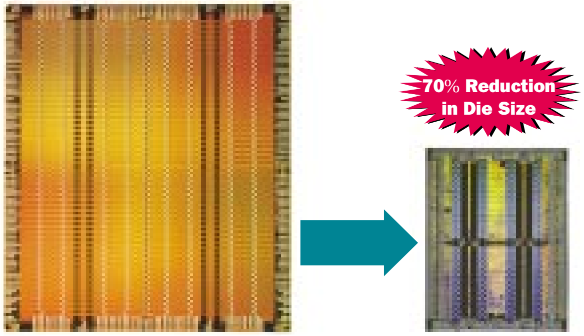
The transition from a high-density programmable logic device (PLD) to an application-specific integrated circuit (ASIC) is difficult. Uncertainty, development costs, resource issues, time-to-market pressures, and conversion risks are just a few of the challenges product-development teams face. Altera now offers a new solution: HardCopy™ devices. Through a combination of proprietary silicon design and an automated migration process, Altera helps designers move seamlessly from a programmable solution to a low-cost, custom implementation of their designs. As a result, Altera® HardCopy devices extend the flexibility and time-to-market advantages of high-density PLDs to the higher volume, more cost-sensitive applications traditionally covered by ASICs. Using HardCopy devices, designers can leverage Altera's solutions from prototype to production, while reducing costs and accelerating time-to-market.

HardCopy devices are a risk-free alternative to ASICs, and are created with Altera's proprietary design migration methodology. They are directly based on Altera PLD architectures and use an area-efficient sea-of-logic-elements core. Essentially, HardCopy devices are an exact reproduction of the PLD of choice with the programmability removed and the customer-specific configuration and routing implemented using metal interconnects. The result is a much smaller and more cost-effective device (see Figure 1).

The HardCopy products support Altera's high-density APEX™ 20KE, APEX 20KC, APEX II, and Excalibur™ devices. HardCopy devices are manufactured with the same process technology as their APEX 20K, APEX II, and Excalibur counterparts (see Table 1 on page 4).

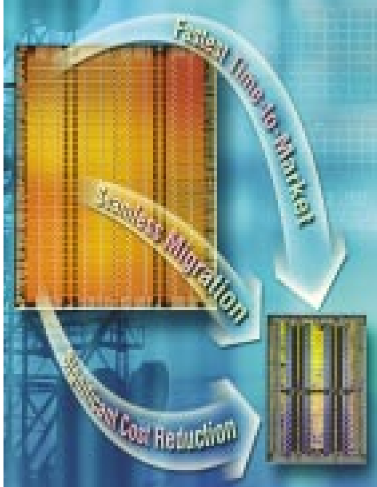
continued on page 4

Figure 1. HardCopy Die Size Comparison



APEX 20KE
EP20K1500E

HardCopy
HC20K1500



Inside This Issue:

Table of Contents

Feature

Devices & Tools

Customer Applications

Contributed Article

Technical Articles

Altera News

In Every Issue

Printed on recycled paper.

John Panattoni,
Publisher
Greg Steinke,
Technical Editor
101 Innovation Drive
San Jose, CA 95134
Tel: (408) 544-7000
Fax: (408) 544-7809
n_v@altera.com

nsai
I.S. EN ISO 9001

ALTERA®

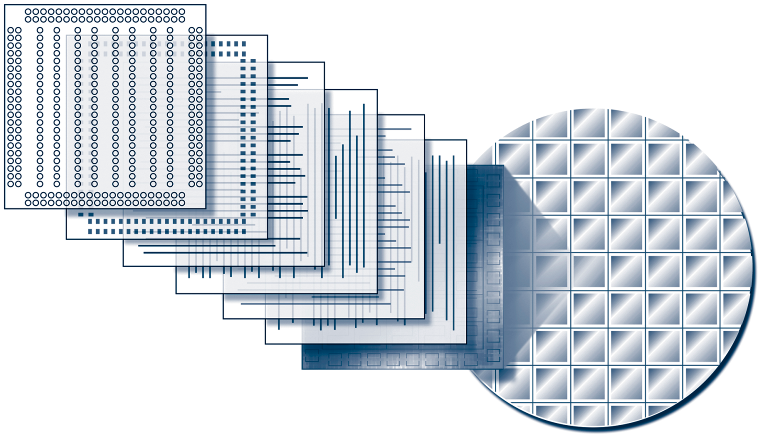
Altera Presents HardCopy Devices—The Low-Risk, Low-Cost Solution for High-Density PLDs, *continued from page 1*

Table 1. HardCopy Device Packaging

Family	Devices	Packages
APEX 20K	EP20K400E, EP20K600E, EP20K1000E, EP20K1500E, EP20K400C, EP20K600C, EP20K1000C	652-pin ball-grid array (BGA), 672-pin FineLine BGA^{TM}, 1,020-pin FineLine BGA
Excalibur	EPXA4, EPXA10	672-pin FineLine BGA, 1,020-pin FineLine BGA
APEX II	EP2A25, EP2A40, EP2A70	724-pin BGA, 672-pin FineLine BGA, 1,020-pin FineLine BGA, 1,508-pin FineLine BGA

HardCopy base wafers, up to the poly layer, are pre-fabricated and await customization in inventory. The customer-specific configuration pattern is extracted from the SRAM Object File (**.sof**). This information is used to configure each individual block using metal interconnects instead of random access memory (RAM) bits. In addition, placement and routing data is fed into an ASIC-type place-and-route tool that completes the interconnection of all the internal blocks of the device. This process uses fewer metal layers than the original PLD. Figure 2 shows the various metal layers used in the HardCopy process. Only 4 metal layers (layers 1 through 4) are used for configuration and place-and-route. Metal layers 5 and 6 are generic and are used for power distribution and bump array layout and distribution.

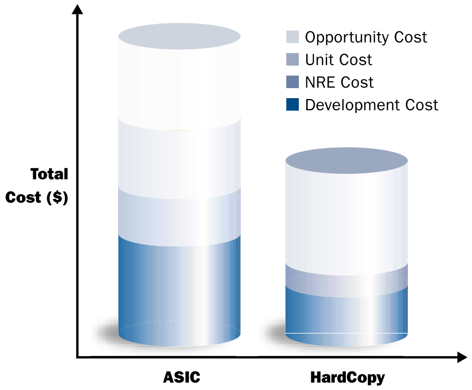
Figure 2. Six-Metal-Layer Process



Reduce Development Costs

Designing and implementing complex ASICs is becoming increasingly, and sometimes prohibitively expensive. Creating a system-on-a-chip (SOC) design requires integrating complex functions such as embedded memory blocks and soft and/or hard embedded processors along with other complex intellectual property (IP) functions and logic. In addition, the rapid evolution of technology, features, and standards can result in shortened product life and escalating revision costs. The development cost of high-density, full-featured ASIC designs with state-of-the-art technology requires an extensive budget and months of engineering resources. HardCopy devices reduce costs and shorten time-to-market because they are a natural extension of flexible PLDs (see Figure 3).

Figure 3. HardCopy Cost Ownership



As new process technologies progress toward deep sub-micron levels, developing an ASIC will become even more expensive. A complete mask set for a 0.13-µm, 6-layer metal standard cell exceeds $750,000. Once the ASIC vendor's other expenses, such as fabrication and design resources, are added to this sum, the non-recurring engineering (NRE) costs for such a design will be in excess of $1,000,000. Few companies will be able to afford such a high cost of entry. In contrast, a HardCopy design, even for the next-generation HardCopy APEX II devices utilizing 0.13-µm technology, will remain well under $300,000.

Another key component of the development cost of an ASIC is the amount of human resources dedicated to this task. Despite the recent developments in EDA tools, the productivity in terms of gates per man-hours has stagnated in recent years. It is not unusual for a million-gate ASIC to take six to twelve months to develop by a team of ten or more engineers. The attractiveness of using a PLD versus an ASIC is that PLDs are significantly easier to verify and induce a very low risk in case the design needs modification. When the time comes to convert a PLD design to an ASIC, the different architecture and timing characteristics make re-synthesis and simulation in the library difficult. A designer must perform a thorough timing verification and spend considerable time on place-and-route and timing closure.

HardCopy devices provide many cost advantages:

- Significantly lower unit cost due to a die size shrink of up to 70% (see Figure 1)
- Lower NRE costs than equivalent ASIC technology
- Lower total cost of ownership (see Figure 3)
- No requirement for customer design engineering resources
- No additional licensing fees or re-characterization when using Altera MegaCore® functions in HardCopy devices
- Guaranteed device functionality and performance

Using HardCopy devices saves money, time, and resources with low risk. Together, these advantages result in a significant reduction to the cost of ownership that no ASIC conversion can match.

Seamless Migration Process

Altera's proprietary design migration methodology generates a HardCopy device that matches the exact functionality of its PLD counterpart and offers equal or better performance. For migrating a high-density PLD design to a HardCopy device, Altera requires output files directly from the Quartus® II software. Customers do not need to resynthesize their design to target a specific ASIC library, and they do not need to invest in costly ASIC development tools.

HARDCOPY™

Altera requires the following output files to complete the HardCopy migration process:

- SRAM Object File (**.sof**)
- Compiler Setting Report File (**.csf.rpt**)
- Pin-Out File (**.pin**)
- Standard Delay Format Output File (**.sdo**)
- Verilog Output File (**.vo**)

The SOF generated by the Quartus II software is used as the input to the HardCopy migration process. The other files are required for verification purposes.

Altera's proprietary design migration methodology generates a HardCopy device that matches the exact functionality of its PLD counterpart and offers equal or better performance.

The migration process uses Altera-developed software and industry-leading, third-party design tools. Designers do not need to provide test vectors or run extensive functional and timing simulations to verify the migration. Because Altera performs all of the migration tasks, HardCopy requires very little customer effort.

An important part of the migration process is ensuring that the new device's timing will match the original timing requirements. Unless the PLD design is fully synchronous, it is impossible to provide an exactly matching HardCopy device in terms of timing characteristics. HardCopy development utilizes pre-existing blocks in the original PLD wherever possible. From that point of view, the

continued on page 6

Altera Presents HardCopy Devices—The Low-Risk, Low-Cost Solution for High-Density PLDs, continued from page 5

timing of the I/O blocks is very close to the PLD's. Likewise, the internal look-up table (LUT) logic and embedded system blocks (ESBs) are also physically identical to the programmable device blocks. However, they are up to 30% faster, due to shorter and more efficient interconnects and removal of programmability overhead.

The largest gain in speed is obtained through the removal of the programmable interconnect. However, this gain is design-dependent and is not predictable on a systematic basis. Once the place-and-route is completed, Altera engineers extract actual interconnect delay information from the layout and back-annotate that delay to the HardCopy netlist, just like an ASIC. At this point, a careful timing analysis is performed and all the results are compared to the PLD performance and checked against customer timing constraints. In most cases, there are few discrepancies or violations. Typically, some hold-time violations may occur because of the faster interconnect. Such violations are immediately fixed by the automatic insertion of delay buffers that are programmed out of the embedded gate array cells that are available throughout the HardCopy die. Figure 4 shows a detailed description of the implementation timeline.

Offering design flexibility and high-performance system-integration functionality, HardCopy devices support the high-density APEX 20KE, APEX 20KC, APEX II, and Excalibur devices.

Another key factor in a successful migration is formal verification at every step of the HardCopy process. Any intervention for either testing or timing engineering change orders (ECOs) is immediately followed by a formal verification step, ensuring that the functionality of the device was not altered.

Testing is also a very important aspect of the HardCopy migration flow. A test coverage of 95% or more is ensured with embedded testability features and automatically generated test vectors. Designers do not need to generate any test vectors; each HardCopy device is thoroughly tested before shipment. Internal logic and I/O buffers are tested through full scan. Initial testability analysis ensures all scan chains will be connected properly and function as expected. Any untestable nodes or combinatorial feedback logic are re-routed through the insertion of additional test logic that will operate only in test mode. ESBs are tested separately, based on their individual configuration, with functional vectors that are shifted in and out through scan registers. Circuitry such as LVDS I/O pins and phase-locked loops (PLLs) have their own dedicated embedded test logic and are tested separately. In addition, the Joint Test Action Group (JTAG) test capabilities are also available as they are in the PLD. However, the JTAG scan order is different in the HardCopy device. Users should download the appropriate BSDL file from the Altera web site at **http://www.altera.com**.

State-of-the-Art Features

Offering design flexibility and high-performance system-integration functionality, HardCopy devices support the high-density APEX 20KE, APEX 20KC, APEX II, and Excalibur devices. HardCopy devices support the same powerful system-level device features, including:

- Packaging: HardCopy devices are available in the same packages with the same pin-outs as the corresponding PLD, maintaining pin-compatibility (see Table 1 on page 4).
- Performance: Altera guarantees that the performance of HardCopy devices will be equal or better than the highest speed grade version of the corresponding PLD. Dedicated and optimized clock trees, timing driven place-and-route, and tight timing controls ensure that no timing violations are introduced during the migration.
- Features: HardCopy devices maintain the exact features of the corresponding PLD circuitry, including the LUT logic structure, True-LVDS™ circuitry, ESBs, phase-locked loops (PLLs), all supported I/O standards, MultiVolt™ operation, and leading-edge process technology. These features eliminate any need to re-characterize the HardCopy device when its programmable counterpart is already qualified.

- Built-in testability: All HardCopy devices include enhanced scan capability automatically inserted during migration. Altera automatically generates production test vectors with a high-fault coverage and does not require any customer-generated-test vectors.
- Intellectual property: Migration of Altera MegaCore and Altera Megafunction Partners Program (AMPP^{SM}) functions from the PLD to the HardCopy device is seamless; no customer involvement is necessary. In addition, customers pay no extra licensing fees for converting Altera IP functions.
- Power consumption: HardCopy devices use significantly less power than their PLD counterparts. The exact reduction in power is design-dependent.

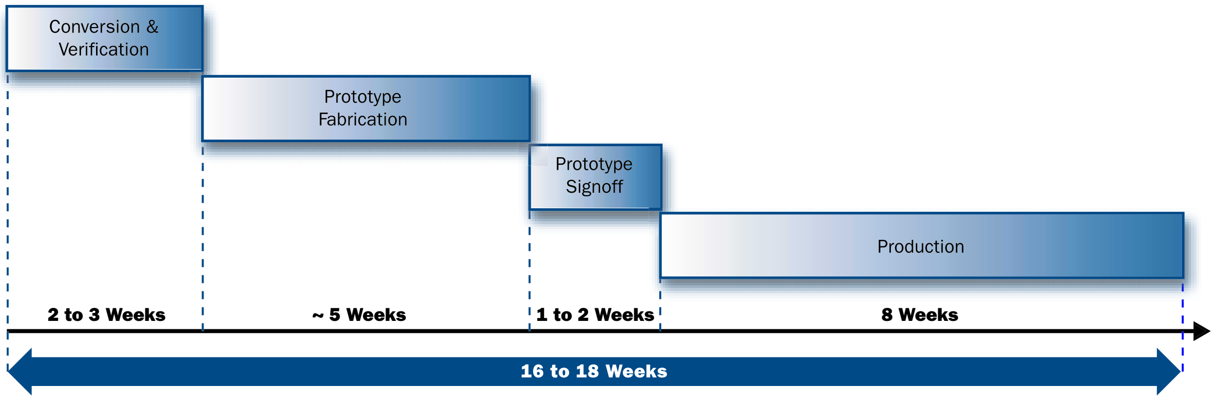
Faster Time-to-Market

Using competitively priced HardCopy devices, manufacturers can enjoy all the benefits of an ASIC and take advantage of shorter development times, expedited production schedules, and faster time-to-market. With an easy migration process, minimal customer involvement, and base wafers already manufactured and ready for customization, Altera HardCopy devices can deliver guaranteed, fully operational production units and prototypes in the shortest time possible, as shown in Figure 4. As system-on-a-programmable-chip (SOPC) design features and IP requirements become more and more complex, HardCopy devices provide the fastest way to migrate high-density PLD designs to a low-risk, low-cost device.

Additional Documentation

For more information, see the HardCopy-related documents at **http://www.altera.com**.

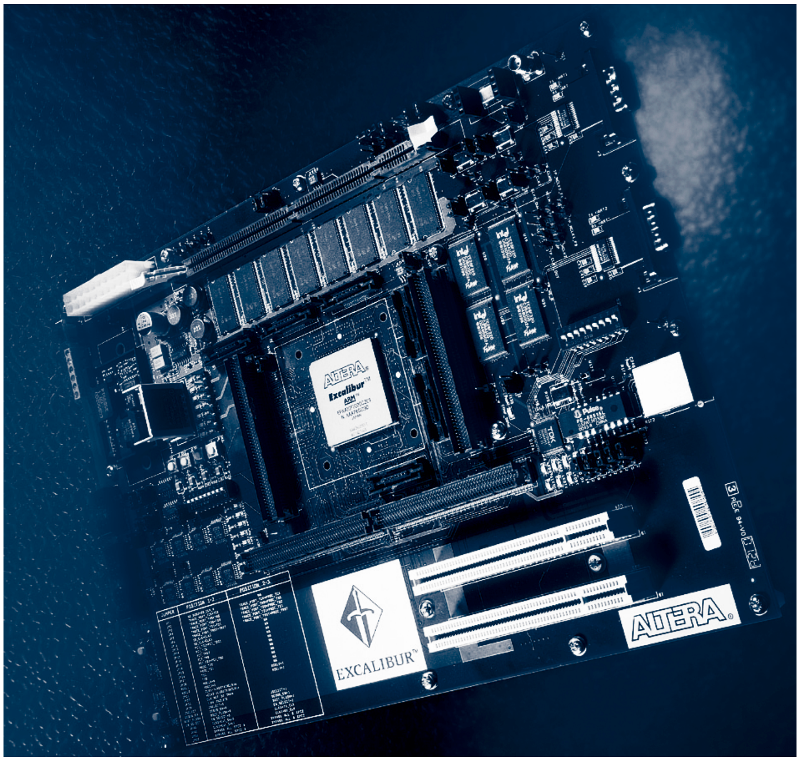
Figure 4. Implementation Timeline



Altera Ships ARM-Based Excalibur Devices & Development Systems

Altera is now shipping complete ARM®-based Excalibur™ development systems. As a part of these development systems, Altera offers the Excalibur Development Kit featuring the EPXA10 device. Included in this kit are the EPXA10 development board (see Figure 1), an Excalibur Solutions Pack (a utilities and resources CD containing numerous hardware and software reference designs), and tutorials. Also included are all power supplies and cables required to get the board up and running.

Figure 1. EPXA10 Development Board



EPXA10 Development Board

This development board enables a designer to easily generate and debug hardware and embedded software designs. The board includes an EPXA10 device that integrates an ARM922T-based processor and memory subsystem (embedded processor stripe) into an APEX™-like architecture.

With 16 Mbytes of 16-bit-wide flash memory, the board provides ample room for storing large user applications, as well as multiple programmable logic configuration images. For an end application that does not use flash memory, eight on-board EPC2 configuration devices can be programmed with the embedded application and the programmable logic configuration data. The EPXA10 development board is comprised of the following:

- EPXA10 device in a 1,020-pin FineLine BGA™ package
- 10/100 Ethernet physical layer (PHY) interface
- Four expansion headers for mounting daughter cards
- 20-pin multi-ICE Joint Test Action Group (JTAG) debug connector
- MICTOR-style embedded trace logic cell connector for embedded system debug

The EPXA10 development board is supported by operating system vendors such as Wind River Systems (including the latest VxWorks operating system), OSE Systems, and Linux.

The board's standard DIMM socket is supplied with a PC100 32-Mbyte, 32-bit single data rate (SDR) SDRAM DIMM.

An Ethernet PHY is available to interface to a 10/100 Ethernet media access controller (MAC) implemented in the programmable logic device (PLD). This intellectual property (IP) core is included and supplied with a protocol stack, allowing the board to be used as an Ethernet device in both development and production environments.

For ease of interfacing to peripheral cards and devices, two full-length 3.3-V, 32-bit peripheral component interconnect (PCI) connectors are provided.

Five oscillators with zero clock-skew distribution circuitry enable multiple independent clock domains within the device.

Also present are general-purpose switches and LEDs for monitoring and user-defined interaction.

Excalibur Solutions Pack

The Excalibur Solutions Pack provides development resources for both the hardware and embedded software, allowing users to quickly get started with their design.

Included are several hardware and embedded software reference designs covering various topics, a getting started guide and "Hello World" example, information on developing multi-layered AMBA™ advanced high-performance bus (AHB) structures, Altera® IP, and instructions on interfacing the embedded processor and programmable logic through the dual-port RAM.

The Excalibur Solutions Pack contains a suite of utilities for creating programming files and downloading them to the board. Software engineers can install and use these utilities without restriction.

Intellectual Property

Using IP functions for system development and product differentiation is fundamental to developing system-level solutions with Excalibur embedded processors. The Excalibur Solutions Pack includes a 1-year license for a synthesizable version of the embedded stripe's universal asynchronous receiver/transmitter (UART) along with an OpenCore® Plus evaluation copy of Altera's 16550 UART and the 10/100 Ethernet MAC MegaCore® functions. The 10/100 Ethernet MAC function includes low-level software drivers with source code. Future plans include additional OpenCore Plus IP functions with software drivers for user evaluation.

Utilities, Resources & Reference Designs

In addition to the utilities and resources CD, the Excalibur Solutions Pack contains demonstrations and reference system designs available from third-party vendors specifically targeting the Excalibur embedded solutions. Offerings include reference designs from Altera Megafunction Partners Program (AMPP℠) partners, as well as additional embedded software development tools and operating systems. Specifically, Red Hat, Inc., provides the GNUPro Tools for EPXA10 devices and Linux operating system support. A demonstration package of Wind River Systems' VxWorks is also included.

Quartus II Support

The Quartus® II subscription includes ARM-based Excalibur device support. Multiple simulation models are included with the support. A functional bus model allows simulation of individual peripherals for support of AHB transactions. A cycle-accurate model of the embedded stripe provides complete hardware and software simulation support. These models operate in the most popular simulation packages.

ARM Developers Suite Lite is provided with the Quartus II software version 1.1. This tool set provides complete support for compiling, linking, and debugging embedded software development using C/C++, as well as Assembly language. You can perform embedded software development within the Quartus II environment.

Using IP functions for system development and product differentiation is fundamental to developing system-level solutions with Excalibur embedded processors.

Conclusion

With the availability of the Excalibur Development Kit featuring the EPXA10 device, a convenient Excalibur desktop SOPC development environment is now available. The tools and resources required for both hardware and embedded software design are included, allowing the developer to rapidly prototype system-level solutions using a high-performance processor combined with high-density, high-performance programmable logic.

Altera Targets High-Performance Systems with Nios Version 2.0

With the Nios™ embedded processor version 2.0, Altera enhances the popular soft core processor to address the needs of high-performance embedded systems designers. By optimizing data flow and data processing operations, the Nios embedded processor provides developers with a level of performance and configurability never before possible in embedded microprocessors.

Nios developers can use the flexibility of programmable logic to accelerate software algorithms by adding custom instruction logic to the processor arithmetic logic unit (ALU). These custom logic units extend the Nios instruction set and let you reduce the number of operations required to perform "inner loop" tasks to a single instruction.

With custom instructions, a complex sequence of operations can be reduced to a single instruction that is implemented in hardware.

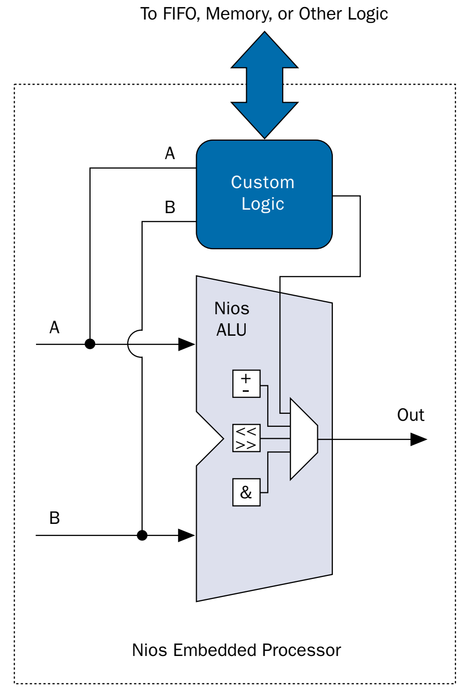
Data-flow optimization is provided for high-throughput applications with the addition of the simultaneous multi-master Avalon™ bus. This new bus architecture permits high-bandwidth peripherals and the Nios embedded CPU access to memory resources simultaneously, allowing the Nios embedded CPU to perform traffic-management tasks instead of being involved in every data transaction.

Data Processing

Developers can accelerate time-critical software algorithms by associating custom logic with a Nios opcode, thus adding new processing capabilities to the Nios instruction set. With custom instructions, a complex sequence of operations can be reduced to a single instruction that is implemented in hardware. This feature is especially useful for developers who need to optimize their software inner loops in digital signal processing (DSP), packet header processing, and other computation-intensive applications.

Custom instructions can be implemented as single-cycle (combinatorial) or multi-cycle (sequential) operations. Additionally, custom instruction logic can access memory and/or logic outside of the Nios system. Figure 1 shows a block diagram of the custom instruction logic.

Figure 1. Custom Instruction Logic



The SOPC Builder software, included with the Nios development kit, provides a graphical user interface that developers can use to add custom instruction logic. Up to five custom instructions can be added to the Nios embedded processor. During the system-generation process, the SOPC Builder automatically creates C-language and Assembly-language macros that can be called from application software.

Custom Instruction Example

One operation that might be implemented as a custom instruction is a byte-order conversion operation. The Nios embedded CPU is little-endian (i.e., byte 0 is the least significant byte), but network-byte ordering is big-endian. Consequently, it is often necessary to convert from big- to little-endian. This conversion can be accomplished in software using a series of logical shifts and masking operations. However, such a software routine requires dozens of instructions to carry out. Figure 2 shows the assembly instructions necessary to convert a 32-bit word from big- to little-endian. This routine (included in the Nios Ethernet Development Kit protocol library) takes more than 20 CPU cycles for a 32-bit value.

Figure 2. Assembly Code for Byte-Order Conversion

```
Address   Instruction
42834:    save %sp,0x17
42836:    mov %g2,%i0
42838:    lsri %i0,0x18
4283a:    mov %g1,%g2
4283c:    lsri %g1,0x8
4283e:    pfx %hi(0xff00)
42840:    and %g1,0x0
42842:    or %i0,%g1
42844:    lsli %g2,0x8
42846:    mov %g1,%g2
42848:    pfx %hi(0x0)
4284a:    movi %g3,0x0
4284c:    pfx %hi(0xe0)
4284e:    movhi %g3,0x1f
42850:    and %g1,%g3
42852:    or %i0,%g1
42854:    lsli %g2,0x10
42856:    or %i0,%g2
42858:    ret
4285a:    restore
```

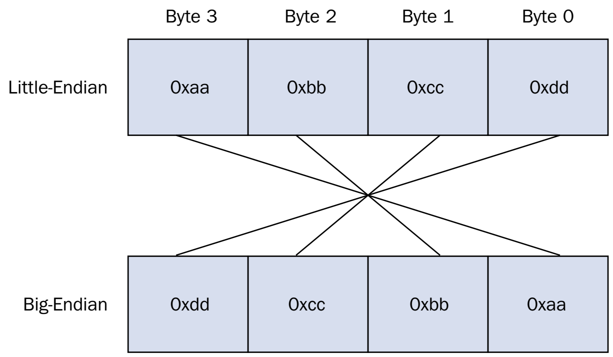
This same operation can be executed in a single clock cycle when implemented as a custom instruction. Figure 3 shows an illustration of the corresponding custom instruction logic to implement the byte-order conversion. This conversion is accomplished using a simple byte-swap technique efficiently implemented as “wires” in programmable logic.

A single custom instruction can be parameterized as well, allowing it to perform one of several operations. By issuing a prefix instruction prior to the custom instruction, an 11-bit value is passed to the custom instruction logic. In this example, the prefix value can determine if the conversion will be big-to-little or little-to-big endian. Custom instructions allow the designer to make hardware/software tradeoffs, tuning their system-on-a-programmable-chip (SOPC) design for optimum data processing performance.

Data Flow

Traditionally, processors have increased data flow for high-bandwidth peripherals by using direct memory access (DMA) techniques. In these applications, the system bus is controlled (or mastered) by the CPU and one or more peripherals. Access to the system bus is provided by a bus arbiter ensuring only one master device is controlling the bus at any time. While bus arbitration permits sharing of a common bus, multiple masters cannot access the bus simultaneously. All but one master must wait their turn before using this shared resource, which imposes an upper limit to the amount of data that can flow through the system.

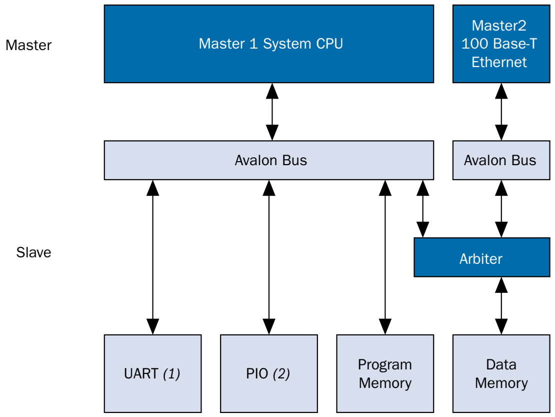
Figure 3. Hardware Byte-Swap



The Avalon bus in the Nios embedded processor version 2.0 implements slave-side arbitration. Slave-side arbitration avoids the bandwidth bottleneck imposed by traditional bus-arbitration schemes. Figure 4 shows slave arbitration performed on the data memory block. The 100 Base-T Ethernet port can act as

continued on page 12

Figure 4. Slave-Side Arbitration



Notes to Figure 4:

(1) UART: universal asynchronous receiver/transmitter.
(2) PIO: parallel I/O.

Altera Targets High-Performance Systems with Nios Version 2.0, continued from page 11

the master of the data memory while the Nios embedded CPU continues to execute instructions out of program memory or service another peripheral.

The Nios embedded processor version 2.0 also introduces a DMA peripheral that provides bus-mastering capabilities to any slave peripheral. The DMA peripheral is used to free the CPU from direct involvement in data transfers, thus increasing overall system performance.

The Nios embedded processor version 2.0 will begin shipping in December 2001. All current Nios embedded processor users will receive the version 2.0 upgrade at no additional cost.

DMA Example

High-bandwidth peripherals found in network, telecommunications, and data-storage applications benefit from the optimized data flow provided by a DMA peripheral and the simultaneous multi-master Avalon bus. For example, consider a system that routes Internet protocol packets between 100 megabits per second (Mbps) Ethernet channels. Using DMA, developers can minimize the number of CPU cycles necessary to transfer each packet.

DMA is used to stream the entire packet into memory with only a few CPU instructions used to set up the DMA peripheral control registers. At 50 MHz, a Nios system can execute approximately 30 million instructions per second (assuming a conservative 0.6 instructions per clock cycle). For example, if an average packet size is 1,024 bytes and the processing required for each packet is 100 instructions, then this system can handle 24 channels. The throughput of this system is greater than 2.4 gigabits per second (Gbps).

A similar system without DMA or simultaneous multi-master bus requires several hundred instructions to process each packet. Using the 1,024-byte packet example, an additional 256 load and store operations would be required to read the data (as 32-bit words) and write it back out. Assuming a new total of 612 instructions per packet, the system throughput would be reduced to under 700 Mbps (i.e., fewer than seven channels). In other words, it would take an equivalent CPU running at 300 MHz to process 24 channels of 100-Mbps traffic that the Nios embedded processor with DMA and simultaneous multi-master Avalon bus could handle at 50 MHz.

Summary & Availability

With the release of the Nios embedded processor version 2.0, Altera is raising the bar yet again on the level of performance and configurability possible in a soft core processor. The Nios embedded processor version 2.0 will begin shipping in December 2001. All current Nios embedded processor users will receive the version 2.0 upgrade at no additional cost. This development kit is available for purchase at the Altera web site at **http://www.altera.com** or by contacting your local Altera sales representative.

To learn more about the Nios embedded processor, visit the Altera web site at **http://www.altera.com**.

MERCURY

Mercury Devices Continue to Roll Out

All devices and speed grades of the Mercury™ product family are now shipping, and industrial temperature devices will be released shortly. As a result, products featuring up to 18 channels of clock data recovery (CDR) support are now readily available. EP1M350 devices feature 18 channels of CDR support, 350,000 typical gates, 14,400 logic elements (LEs), 112 Kbits of RAM, and a maximum of 486 user I/O pins in a 780-pin FineLine BGA™ package. High-speed serial links make these devices ideal for serial backplane applications. See Table 1 for availability schedules.

Table 1. Mercury Device Availability

Device	Package	Temperature Grade	Speed Grade	Availability
EP1M120	484-pin FineLine BGA	Commercial	-5, -6, -7	Now
		Industrial	-6	Q1 2002
EP1M350	484-pin FineLine BGA	Commercial	-5, -6, -7	Now
		Industrial	-6	Q1 2002

Mercury Multiplier Performance

The Mercury architecture incorporates a distributed multiplier capability, enabling the simple and direct implementation of high-speed multipliers (see Table 2). This support is provided through dedicated silicon built into each LE, and through highly optimized aspects of the core routing. Optimized for 16 × 16 implementation, this multiplier can also be configured for smaller or larger implementations, and is provided via push-button implementation in the Quartus® II software version 1.1. Combined with a performance-targeted logic array, this multiplier mode provides the power necessary to process the high-bandwidth capabilities of a Mercury device.

Table 2. Mercury Device Multiplier Performance

Design	EP1M350	EP20K400C
16 x 16 pipelined multiplier	303 MHz	130 MHz
16 x 16 non-pipelined multiplier	154 MHz	64 MHz

EXCALIBUR

ARM-Based Excalibur Development System

As a part of the Excalibur™ development system, Altera is supplying users with the EPXA10 development kit. Included in this kit are the EPXA10 development board, an Excalibur Solutions Pack, a utilities and resources CD containing numerous hardware and software reference designs, and tutorials. For more information, see “Altera Ships Arm-Based Excalibur Devices & Development Systems” on page 8.

Altera Announces Nios Version 2.0

Altera released the Nios™ embedded processor version 2.0, the industry-leading soft core embedded processor optimized for the high-performance, high-bandwidth needs of network, telecommunications, and mass storage applications. The Nios embedded processor version 2.0 offers features that expand the performance and flexibility of system-on-a-programmable-chip (SOPC) solutions, including:

- Simultaneous multi-master Avalon™ bus
- Custom instruction capabilities
- Advanced debug solutions

The Avalon bus is Altera’s parameterized bus used by the Nios embedded processor. It can be thought of as a set of pre-defined signal types where a user can connect one or more intellectual property (IP) blocks.

continued on page 14

Devices & Tools, continued from page 13

The simultaneous multi-master Avalon bus and custom processor instructions deliver a level of performance and configuration never before possible in a soft core processor implemented in programmable logic. The Nios on-chip debug peripheral provides world-class debugging capabilities to accelerate embedded software development through a robust set of real-time debug and software trace facilities. These enhancements provide designers with significant advantages compared to alternative embedded processor or soft core solutions. For more information, see "Altera Targets High-Performance Systems with Nios Version 2.0" on page 10.

APEX II

APEX II Devices Now Available

EP2A15 and EP2A25 devices are now available. APEX™ II devices range in density from 16,640 to 67,200 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. Based on a leading-edge 0.13-µm and 0.15-µm all-layer-copper interconnect processes, 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, HyperTransport, HSTL, and SSTL. With True-LVDS™ circuitry, APEX II devices can achieve data transfer rates of up to 1 gigabits per second (Gbps) per channel. With these I/O features, APEX II devices can be used in the following applications:

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

See Tables 3 and 4 for availability schedules and software support for APEX II devices.

Table 3. APEX II Device Availability

Device	Availability
EP2A15	Now
EP2A25	Now
EP2A40	Q1 2002
EP2A70	Q4 2001

Table 4. APEX II Devices & Quartus II Software Support Availability

Device	Package	Software Support Availability
EP2A15	672-pin FineLine BGA *(1)* 724-pin BGA	Now
EP2A25	672-pin FineLine BGA 724-pin BGA 1,020-pin FineLine 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

Note to Table 4:
(1) BGA: Ball-grid array.

APEX II IP Offerings

Altera offers a wide variety of APEX II-optimized IP functions to further speed design cycles. Some APEX II-optimized IP functions now available for purchase include the POS-PHY Level 4, Flexbus Level 4, DDR SDRAM controller, and SDR SRAM controller. Additionally, the ZBT SRAM and QDR SRAM controller reference designs are available for free. To obtain these functions, contact your local Altera sales representative or go to the IP MegaStore™ web site at **http://www.altera.com**.

APEX II HardCopy Solution

Altera offers a migration solution from APEX II programmable logic devices (PLDs) to HardCopy™ devices for system designers who need a low-risk cost-reduction path for high-volume production. Time-sensitive applications can be prototyped and ramped up into production using APEX II devices. When the design is ready for high-volume production, system designers can reduce overall costs by migrating the design to HardCopy devices. HardCopy devices preserve the functionality and timing of the design and allow system designers to improve time-to-market at the lowest cost for high-volume production. For more information on HardCopy devices, see "Altera Presents HardCopy Devices—The Low-Risk, Low-Cost Solution for High Density PLDs" on page 1.

APEX II Industrial Offerings

Altera has proactively selected industrial-grade devices for the APEX II device family. Therefore, you can now identify industrial-grade APEX II devices in the early stages of the family rollout, further compressing design cycles for the fastest possible time-to-market. A -8 speed grade will be available in the industrial-grade production versions of the device offerings. Table 5 shows the availability for industrial-grade offerings.

Table 5. APEX II Industrial Device Offering

Device	Package	Availability
EP2A15	672-pin FineLine BGA	Q1 2002
EP2A25	672-pin FineLine BGA 724-pin BGA 1,020-pin FineLine BGA	Q1 2002 Q1 2002 Q1 2002
EP2A40	1,020-pin FineLine BGA	Q2 2002

APEX II Price Reduction

Due to the overwhelming response to the APEX II family, Altera is able to reduce the cost of these devices. As a result, APEX II device pricing has been reduced by over 40%. Additionally, long-term volume pricing has been reduced significantly. Contact your local sales representative to get the latest APEX II device pricing.

APEX

APEX 20KC Devices Now Shipping

EP20K200C, EP20K400C, EP20K600C, and EP20K1000C devices are now shipping. These new high-performance APEX devices address the high-bandwidth needs of SOPC solutions. These products combine the state-of-the-art features found in APEX 20KE devices with an industry-leading 0.15-µm all-layer-copper interconnect technology to provide performance improvements of 25% over 0.18-µm-based devices. Table 6 shows the availability schedule for APEX 20KC devices. All APEX 20KC devices are supported in the Quartus II software version 1.1, as shown in Table 7.

Table 6. APEX 20KC Device Availability

Device	Availability
EP20K200C	Now
EP20K400C	Now
EP20K600C	Now
EP20K1000C	Now
EP20K1500C	Q2 2002

Table 7. APEX 20KC Devices & Quartus II Software Support Availability

Device	Package	Software Support Availability
EP20K200C	208-pin PQFP *(1)* 240-pin PQFP 356-pin BGA 484-pin FineLine BGA	Now Now Now Now
EP20K400C	652-pin BGA 672-pin FineLine BGA	Now Now
EP20K600C	652-pin BGA 672-pin FineLine BGA 1,020-pin FineLine BGA	Now Now Now
EP20K1000C	652-pin BGA 672-pin FineLine BGA 1,020-pin FineLine BGA	Now Now Now
EP20K1500C	652-pin BGA 1,020-pin FineLine BGA	Now Now

Note to Table 7:
(1) PQFP: plastic quad flat pack.

continued on page 16

Devices & Tools, continued from page 15

Industrial-Grade APEX Offerings

Industrial-grade APEX devices are available in various package offerings. See Tables 8 through 10.

Table 8. APEX 20KC Device Industrial Offering

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

Table 9. APEX 20KE Device Industrial Offering

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

Note to Table 9:

(1) The “X” denotes phase-locked loop (PLL) and LVDS support.

Table 10. 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 10:

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

ACEX

ACEX 1K Software Support

The Quartus II software version 1.1 and the MAX+PLUS® II software version 10.1 offer full software support for ACEX™ 1K devices. Industrial offerings are available in 13 of the 14 device and package combinations. Table 11 shows which packages are now available in industrial grades. In addition, a wide range of ACEX device-optimized IP functions can now be found at the Altera® IP MegaStore web site.

Table 11. ACEX 1K Industrial-Temperature Device Availability

Device	Package	Speed Grade *(1)*
EP1K10	100-pin TQFP *(2)* 144-pin TQFP 208-pin PQFP 256-pin FineLine BGA	-2 -2 *(3)* -2
EP1K30	144-pin TQFP 208-pin PQFP 256-pin FineLine BGA	-2 -2 -2
EP1K50	144-pin TQFP 208-pin PQFP 256-pin FineLine BGA 484-pin FineLine BGA	-2 -2 -2 -2
EP1K100	208-pin PQFP 256-pin FineLine BGA 484-pin FineLine BGA	-2 -2 -2

Notes to Table 11:

(1) This speed grade supports PLLs.
(2) TQFP: Thin quad flat pack.
(3) Consult Altera for more information.

MAX

MAX 7000B Industrial Ordering Codes

Industrial-grade devices are widely used for a range of telecommunications and networking applications that are exposed to extreme temperatures. These devices undergo extensive product characterization and reliability stresses to withstand extreme temperatures. Specific applications that use industrial-grade products include wireless base stations, telecom switches, industrial controllers, and automotive applications.

MAX® 7000B devices offer a wide variety of packages and speed-grade options in the industrial-temperature range. To further enhance its industrial-grade offerings, Altera has introduced five new MAX 7000B ordering codes in the industrial-temperature range. With the addition of these ordering codes, a total of 19 industrial ordering codes are now available for MAX 7000B devices. Table 12 lists package and speed-grade availability for the new industrial ordering codes. The ordering codes have been priced 25 to 45% lower than the previously offered (-5) and (-7) industrial ordering codes in the same package, to target price-sensitive applications.

MAX Applications Web Site

The MAX Applications web site has been published at **http://www.altera.com**. This web site describes applications where MAX devices are used, particularly outside the typical glue/control logic realm. This web site provides detailed technical information for using MAX devices in these specific applications and includes references to white papers on these applications.

The MAX Applications web site currently features four applications:

- Integrating multiple I/O transceivers and buffers
- Integrating multiple LED driver chips
- Interfacing with multiple I/O voltages on a system board with a single MAX device
- Integrating digital switch matrices

Lattice Emergency Obsolescence

Starting January 1, 2002, Lattice Semiconductor will be discontinuing several of their PAL, MACH1, and MACH2 devices. Most designers will have to re-engineer their existing designs to new PLD families. Altera can help designers with this transition by offering the MAX 7000AE, MAX 7000S, and MAX 3000A devices.

Altera is the largest supplier of product-term-based PLDs in the market today and is committed to maintain a leadership position in this market segment by providing a reliable, long-term supply of a variety of product-term device families.

CONFIGURATION

Enhanced Configuration Devices

Altera provides a complete single-device solution for a wide range of density requirements with its enhanced configuration devices. Vertical migration capability allows a designer to migrate easily from the EPC4 to the EPC8 to the EPC16 device in the same package, without having to change the board layout. EPC4, EPC8, and EPC16 devices are now available.

continued on page 18

Table 12. New MAX 7000B Device Industrial Ordering Codes

Device	Package				
	44-Pin PLCC	44-Pin TQFP	100-Pin TQFP	256-Pin BGA	256-Pin FineLine BGA
EPM7064B	Ind. (-7)	Ind. (-7)	Ind. (-7)		
EPM7512B				Ind. (–10)	Ind. (-10)

Devices & Tools, continued from page 17

Enhanced configuration devices offer in-system programmability (ISP) through a built-in IEEE standard for the boundary-scan-based in-system configuration standard of programmable devices (IEEE 1532). The inclusion of ISP and reprogrammability 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 parallel configuration capability to accelerate configuration times, a new page mode that allows users to store multiple configurations, block protection for partial reprogramming support, and full-clocking flexibility through the programmable clock and external clock features.

Unused portions of the flash memory can be used as general-purpose memory via a built-in processor access feature. This advanced feature set enhances the overall PLD design experience while improving manufacturability for a high-volume environment.

TOOLS

Quartus II Version 1.1 Service Pack 2 Now Available

The Quartus II software version 1.1 Service Pack 2 (SP2) will be available on the Altera web site in November 2001. Customers will also be able to request a CD-ROM with SP2 from the Altera web site. SP2 includes all of the updates and enhancements that are in Service Pack 1 (SP1). Both SP1 and SP2 include support for new Altera devices, enables LogicLock™ features, and resolves several hundred software issues. Table 13 shows the software support for new Altera devices.

LogicLock Design Flow Included at No Charge

Installing the Quartus II software version 1.1 SP1 or SP2 now enables LogicLock design flow features without requiring any additional licensing. The LogicLock design flow features let you shorten design cycles, preserve performance, and have more control over logic placement in Altera devices. For more information on the LogicLock design methodology, visit the Design Software section of the Altera web site at **http://www.altera.com**.

Table 13. The Quartus II Software Version 1.1 SP1 & SP2 Support for New Devices

Support	Family	Device	Package
Advanced compilation and simulation support	APEX II	EP2A40	724-pin BGA, 1,020-pin FineLine BGA
		EP2A70	724-pin BGA, 1,020-pin FineLine BGA, 1,736-pin FineLine BGA
	ARM-Based Excalibur	EPXA4	672-pin FineLine BGA
	Mercury	EP1M120	484-pin FineLine BGA
Full support, includes Programmer Object File (**.pof**) generation	APEX II	EP2A15	672-pin FineLine BGA
		EP2A25	672-pin FineLine BGA, 724-pin BGA, 1,020-pin FineLine BGA
	Mercury	EP1M350	780-pin FineLine BGA
	Configuration Devices	EPC4	100-pin PQFP
		EPC8	100-pin PQFP
		EPC16	100-pin PQFP

The following documents are also available:

- *LogicLock Methodology White Paper*
- *AN 161: Using the LogicLock Methodology in the Quartus II Design Software*
- *AN 164: LeonardoSpectrum & the Quartus II LogicLock Design Flow*
- *AN 165: Synplify & the Quartus II LogicLock Design Flow*
- *AN 171: FPGA Express & the Quartus II LogicLock Design Flow*

For more information on the LogicLock design flow, see "Motorola on Using the LogicLock Incremental Design Flow in the Quartus II Software" on page 38.

Using SignalTap Logic Analysis

SignalTap® logic analysis can be used to capture any internal node or I/O signals from APEX II, APEX 20K, or ARM®-based Excalibur devices operating in-system and at system speeds. For details on how to set up and use the SignalTap logic analysis capabilities in the Quartus II software version 1.1, see *Application Note 175 (SignalTap Analysis in the Quartus II Software Version 1.1)*.

New Altera Programming Unit Now Available with USB Support

The Altera Programming Unit (APU), used with the appropriate programming adapters, includes all of the hardware and software necessary to externally program Altera MAX, Classic™, and configuration device families (see Figure 1).

A direct connection from the APU to the universal serial bus (USB) port of a PC simplifies installation and enables APU use with modern high-performance PCs. The APU has the following features:

- Support for externally programming Altera MAX and configuration devices
- USB interface to the PC (ISA card not required)
- Advanced pin-driver circuits supporting low voltage device families
- Built-in self-test function to guarantee performance
- Support for high pin-count devices
- Integrated with MAX+PLUS II version 10.1 and higher, Quartus II APU support will begin in Q1, 2002 on PC platforms
- Compatible with the same adapters used with the PL-ASAP2 (PL-MPU + LP6 card) programmer

The APU, ordering code PL-APU, is available today. Additional adapters are required for operation; a list of available adapters can be found in the Development Kits/Cables section of the Altera web site. The APU supports the Windows 2000 and Windows 98 operating systems.

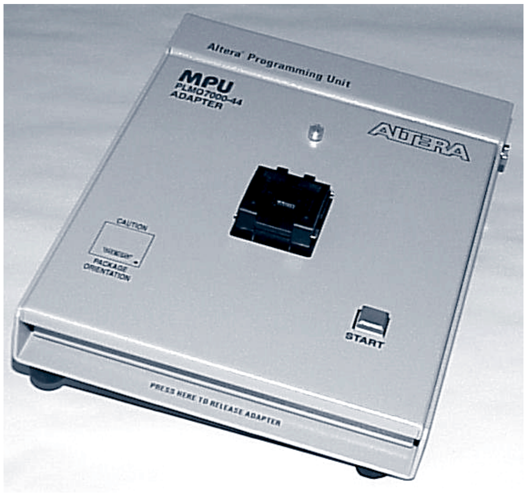
Download the Latest OEM Synthesis & Simulation Tools

All customers with active subscriptions can download the latest versions of the OEM synthesis and simulation tools Altera includes with software subscriptions from the Altera web site. These new versions include support for the new APEX II device family. Table 14 shows the versions available.

Table 14. OEM Synthesis & Simulation Tools

Tool	Version
Exemplar Logic™ LeonardoSpectrum™-Altera Edition	2001.1d
Model Technology ModelSim® -Altera Edition	5.5b

Figure 1. Altera Programming Unit (Shown with the PLMQ7000-44 Adapter)



Discontinued Devices Update

To increase operational efficiency in its manufacturing flow, Altera will be discontinuing the devices shown in Table 1. Table 1 also provides full details on last-order and last-shipment dates.

Altera distributes advisories (ADVs) and product discontinuance notices (PDNs) that provide information on discontinued devices. The ADVs and PDNs are available on the Altera web site at **http://www.altera.com.** Continued support for devices beyond the phase-out period may be available through Rochester Electronics, an extended after-market supplier. Contact Rochester Electronics at (508) 462-9332 for more information. For additional information, contact your local Altera sales office.

The Classic™ device process migration from Cypress to TSMC has been cancelled. Altera will continue to supply Classic devices to its customers from its existing Cypress supply source. For additional information, contact your local Altera sales representative.

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	11/31/02	PDN 0117
	EPM7192EGM160-15	05/31/02	11/31/02	PDN 0117
MAX 7000S	Selected devices in plastic quad flat pack (PQFP) and thin quad flat pack (TQFP) packages	05/31/02	11/31/02	PDN 0117
MAX 9000	Selected device packages	02/28/02	08/31/02	PDN 0106
	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 & ball-grid array (BGA) packages	02/28/03	08/31/03	PDN 0107
	EPF10K10LI84-4	05/31/02	11/31/02	PDN 0117
FLEX 10KA	Selected FLEX 10KA PGA packages	02/28/03	08/31/03	PDN 0107
	EPF10K30AFI256-2	05/31/02	11/31/02	PDN 0117
FLEX 10KE	Selected FLEX 10KE PGA packages	02/28/03	08/31/03	PDN 0107
APEX™ 20K	Selected APEX 20K PGA packages	02/28/03	08/31/03	PDN 0107
	EP20K400BI652-2	05/31/02	11/31/02	PDN 0117

Questions & Answers

Q *How do I make pin assignments in VHDL using the LeonardoSpectrum™ tool?*

A It is possible to make pin assignments directly in your VHDL code when using the LeonardoSpectrum tool. The pin assignments will be embedded in the resulting EDIF and will be recognized and implemented by Altera's software. An Exemplar™ library must be included to define the `PIN_NUMBER` and `ARRAY_PIN_NUMBER` used to make the assignments (see Figure 1).

Figure 1. Example Code

```
library EXEMPLAR;
 use EXEMPLAR.EXEMPLAR_1164.ALL;

ENTITY pin_ex IS
  PORT(
       myin : IN bit_vector (2
downto 1);
       myout      : OUT bit
      );

          ATTRIBUTE PIN_NUMBER
OF myout : SIGNAL IS “144”;
          ATTRIBUTE
ARRAY_PIN_NUMBER OF myin :
SIGNAL IS (“141”, “142”);
END pin_ex;

ARCHITECTURE arch OF pin_ex IS
BEGIN
  PROCESS (myin(1), myin(2))
  BEGIN
    myout <= (myin(1) AND
myin(2));
  END PROCESS;
            END arch;
```

You cannot make VHDL pin assignments to signals that are part of a bus in versions 2001.1d and lower.

Q *What are the rise and fall times for Mercury™ devices with and without slow-slew rate?*

A Rise and fall times were observed for the Mercury EP1M350 device under no-load conditions (see Table 1). These measurements were made at room temperature with nominal V_{CC} and a typical process. Therefore, these reported values can be used for guideline and should not be interpreted as a strict specification. Rise time is the change from 10% to 90% of V_{CCIO} and fall time is 90% to 10% of V_{CCIO}. SSR refers to the slow-slew rate option and FSR refers to fast-slew rate (slow-slew rate turned off).

Table 1. EP1M350 Device Rise & Fall Times

V_{CCIO}	SSR (ns)	FSR (ns)
3.3-V rise time	1.59	1.03
3.3-V fall time	1.68	0.96
2.5-V rise time	2.11	1.22
2.5-V fall time	2.01	1.05
1.8-V rise time	1.45	0.82
1.8-V fall time	1.67	0.84

Q *What behavior should I expect when I perform a warm reset on my ARM®-based Excalibur™ device?*

A A warm reset clears the contents of the programmable logic device (PLD) and re-boots the stripe.

If `BOOT_FLASH` is `0`, then a warm reset asserts `nSTATUS` to reconfigure the PLD and the device is booted from an external source. The processor is held in reset while the PLD and the stripe are reconfigured.

If `BOOT_FLASH` is `1`, a warm reset will cause the bottom 32 Kbytes of `EBI0` to be mapped at address `0`. The processor will access the boot code from either 8- or 16-bit flash memory.

In conclusion, during a warm reset, the PLD is cleared and reconfigured and the stripe is re-booted.

continued on page 22

Questions & Answers, continued from page 21

Q *Can the ARM processor in the ARM-based Excalibur device boot from the embedded SRAM within the stripe?*

A Yes. The ARM processor in ARM-based Excalibur devices can boot from either the external flash memory or from the embedded SRAM in the ARM stripe. When booting the processor using the embedded SRAM, the configuration unit (download cable or configuration device) loads the embedded SRAM and the PLD while the processor is held in reset.

Q *Is the Atlantic™ interface bidirectional?*

A The Atlantic interface is unidirectional; separate interfaces are required for transmit and receive. Some cores implement both transmit and receive on the same function (e.g., Packet Processor - PP155) and some implement them with separate instantiations (e.g., POS-PHY Level 2). Refer to the appropriate user guide for any particular function.

Q *Do the compiled libraries provided in the <modelsim install path>\altera directory need to be compatible with the version of the Quartus® II software that is used to create the simulation netlist?*

A Yes. It is critical that the compiled libraries provided in the <*modelsim install path*>**\altera** directory are compatible with the version of the Quartus II software that is used to create the simulation netlist.

To ensure that you have the right version of the libraries installed, check the version in the <*modelsim install path*>**\altera\version.txt** file against the Quartus II version in the source file tags.

The Quartus II version can be found by using the following DOS command:

```
cd <modelsim install path>\altera
   <modelsim install path>\win32aloem\
   vdir -l -lib ↵
```

where <*library name*> = (e.g., mercury).

Q *How do I enable or disable the Quartus II TalkBack feature?*

A This feature is disabled by default in the Quartus II software.

By using the following procedure, you may enable or disable TalkBack at any time.

1. Go to the Quartus II software installation directory (version 1.1 or higher).
2. In the <*Quartus II install*>**\bin** folder, run **qtb_install.exe.**
3. Read the **TalkBack Feature Notice**, and choose **Accept** if you accept the terms.
4. Choose **Enable** or **Disable** and then click **OK.**

The full TalkBack disclosure can be accessed by running **qtb_install.exe**.

Q *Can I use the LogicLock™ design methodology in conjunction with Mercury devices?*

A This functionality is not supported in the Quartus II software version 1.1.

LogicLock methodology support for the Mercury family will be available in a future version of the Quartus II software.

Q *When using the LogicLock design flow, do I need to generate a Quartus II Verilog Quartus Mapped (**.vqm**) file if I already have a third-party EDIF input file (**.edf**) or VQM?*

A No. If all the logic is represented in the ATOM netlist (EDF or VQM file), you do not need to generate an ATOM netlist from the Quartus II software. An ATOM netlist fixes the nodes and node names of the design, allowing you to back-annotate the node locations of the design.

If you are black-boxing library of parameterized modules (LPM), you must generate a Quartus II VQM for the LPM or generate a Quartus II VQM for the whole design.

Altera Delivers Competitive Edge to BlueArc

BlueArc corporation has introduced the Si7500 Storage System, the world's fastest network storage solution. Leveraging the power of BlueArc's revolutionary SiliconServer™ architecture, the Si7500 Storage System delivers unparalleled throughput, scalability, and performance to network storage applications. In combination with Altera's system-on-a-programmable-chip (SOPC) solutions, BlueArc has redesigned traditional storage systems for multi-gigabit data throughput.

SiliconServer Architecture

The SiliconServer architecture is a revolutionary approach to storage technology that overcomes the limitations of traditional server architectures by implementing speed-critical data-paths using programmable logic-accelerated hardware rather than software. BlueArc is currently shipping the first product based on the SiliconServer architecture, the Si7500 Storage Server, marking an important milestone in the evolution of products that achieve higher performance by accelerating mission-critical functions with programmable logic.

SOPC Solutions

Altera's SOPC solutions allow BlueArc to develop a product that far surpasses the capabilities of traditional servers—with the performance and reliability of hardware and the flexibility of software.

About the Si7500 Storage System

BlueArc's Si7500 Storage System is the world's fastest network-attached storage system, and delivers unparalleled throughput and performance to enterprise storage applications. The Si7500 delivers a breakthrough design utilizing Altera programmable logic that brings new levels of data throughput, scalability, and reliability. Leveraging the power of the revolutionary new SiliconServer architecture BlueArc delivers:

- 2,000 megabits per second (Mbps) throughput
- Up to 250 terabytes of storage capacity
- Thousands of simultaneously connected users
- 99% solid-state availability

BlueArc's Si7500 Storage System is built around the revolutionary SiliconServer architecture, and has two dedicated, unidirectional high-speed data paths. This architecture utilizes Altera programmable hardware, rather than software, to transfer data on and off the network at Gigabit rates—faster than any other storage solution available.

"Programmable logic is an enabling technology for the SiliconServer architecture," said Geoff Barrall, chief technology officer at BlueArc. "By providing BlueArc with a solution that has the performance and reliability of hardware and the flexibility of software, Altera's SOPC solutions allowed us to develop a product that far surpasses the capabilities of traditional servers. Additionally, they afforded us the traditional benefits of programmable logic usage, including development flexibility, rapid design cycles and fast time-to-market."

Barrall continued, "The in-field reprogrammability of Altera's programmable logic devices (PLDs) allows BlueArc to upgrade the Si7500 Storage System in the field as easily as delivering a software patch."

Conclusion

By leveraging the power of programmable logic, BlueArc has designed and delivered a revolutionary product to the storage market. The Si7500 has achieved new levels of performance and scalability, delivering the benefits of true gigabit networking to the end user.

About BluArc

Founded in 1998, BlueArc Corporation delivers the world's highest-performance, enterprise-class network storage systems. BlueArc's headquarters is located in San Jose, California. For more information on BlueArc Corporation, visit their web site at **http://www.bluearc.com**.

Company:
BlueArc Corporation

Industry:
Storage Systems

End Product:
Si7500 Storage Server

Altera Products:
APEX 20KE Devices

Logic Product Development's Digital Trail Camera Wins Cabela's Annual Buyers' Choice Award

Company:
Logic Product Development

Industry:
Digital Cameras

End Product:
Recreational Digital Camera

Altera Products:
ACEX 1K Devices

The bronze award winner of Cabela's 2001 Annual Buyer's Choice Awards, Nature Vision's Game-Vu digital trail camera, was developed by the Minneapolis firm Logic Product Development. Logic Product Development is the largest fully integrated product development consulting firm in the Midwest and a member of the Altera Consultants Alliance Program (ACAP®) partnership. Cabela's is one of the nation's largest outdoor retailers, with a worldwide catalog circulation of 60 million, seven retail superstores, and an extensive web site that is popular among sportsmen (**www.cabelas.com**).

About ACAP

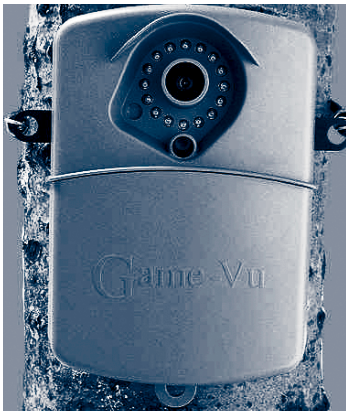
As programmable logic device (PLD) densities continue to increase, Altera recognizes that designers require tools and expertise that will increase productivity and allow them to keep pace with the increasing capacity of PLDs. Increasing PLD densities drive more complex designs, which in turn increases the need for support. Over the past few years, this requirement has caused a growing trend to outsource design work.

At times, the necessary design work is either not completely defined or is in transition and cannot be successfully solved by using off-the-shelf intellectual property (IP) functions. In these situations, a consultant with a specific specialization can be used to respond to changes and modification requests in real-time. ACAP partners provide expert design assistance to users of Altera PLDs and help them quickly get their products to market. Additionally, ACAP consultants specialize in niche areas, which makes them particularly effective at solving specific problems and facilitating time-to-market needs.

Game-Vu Digital Camera

Unlike traditional, unattended 35-mm scouting and trail cameras, Game-Vu (**www.gamevu.com**) is an easy-to-use system that utilizes the latest technology to operate without noise, visible flash, or film. Game-Vu does not require focusing or complicated settings, and users can view time- and date-stamped images in the field or elsewhere on a variety of video devices. Game-Vu incorporates a solid-state design with no moving parts, ideal for monitoring game trails and for security. It employs 16 infrared LEDs that provide an invisible source of illumination. Upon detecting a target, Game-Vu activates the right number of infrared LEDs and captures subjects in the dark up to 25 feet away (see Figure 1).

Figure 1. Game-Vu Digital Trail Camera



Integrated Approach

Logic's integrated approach enables different disciplines to work concurrently on developing the product, which reduced development time and helped Nature Vision bring the Game-Vu to market quickly. Logic's industrial design team created several mockups of the user interface and developed the one most intuitive for the end user, while the mechanical engineering team modeled the printed circuit board (PCB) in Pro-Engineer to ensure that the mechanical enclosure and the electronics would piece together correctly. The software engineering team coaxed the entire software design onto a low-cost eight-bit microcontroller, while electrical engineers designed the board. Logic's PLD team used VHDL to design the control and communications logic into an Altera® ACEX™ EP1K10 device.

"If we hadn't possessed in-house PLD capabilities, there is no way we could have successfully created this product. The initial design incorporated most of the functionality within the processor instead of the PLD. This issue proved too intensive for the microcontroller, so we moved to a PLD-based approach. The EP1K10 device handled bus switching and translation of four different bus topologies flawlessly and much more elegantly than the original solution," said Hans Rempel, the Logic program manager.

The Logic team had many challenges to overcome. Traditional digital cameras are notorious for quickly using up battery power. The Game-Vu powers on, focuses, and takes a picture within 100 milliseconds under any lighting condition, an extraordinary achievement given the time and energy a traditional digital camera takes to accomplish the same tasks. The Game-Vu uses 6 AA batteries that provide over 150 hours of life, and operates from 0° to 120° F. The temperature range increases considerably with a 12-V power source; for continuous operation, the unit can be connected to a 12-V DC adaptor.

"The original concept design was technically sound. However, due to I/O requirements and the number of external components needing control and supervision, production units would have required a costly, high-performance microprocessor. We were able to leverage our expertise in programmable logic design to replace the time-critical and I/O-intensive operations with the EP1K10 device. Using the EP1K10 device allowed us to produce a working camera that captures and stores pictures at higher speeds, and at a much lower cost than the microprocessor-dominant solution," said Scott Wilken, who leads the application-specific integrated circuit (ASIC) and PLD design group at Logic.

"Using the EP1K10 device, allowed us to produce a working camera that captures and stores pictures at higher speeds, and at a much lower cost than the microprocessor-dominant solution."

About Logic Product Development

Logic Product Development creates superior value and competitive advantages for clients seeking design solutions. By integrating consumer needs and client capabilities, Logic generates award-winning, market-smart solutions with the latest technology. For more information on Logic, visit their web site at **http://www.logicpd.com**.

Contributed *Article*

BLIS for Altera: Using Block-Level Incremental Synthesis in Synopsys FPGA Compiler II

Advancements in programmable logic device (PLD) capacity and speed have made the technology increasingly attractive for million-gate designs. As design complexity grows, so does the need for advanced synthesis and place-and-route tools. One highly sought feature is the ability to recompile only the modified portion of a design after modifications have been made. This feature is needed not only to reduce compilation time, but also to preserve current timing behavior of certain sections of a design when other parts of the design are changed.

To address these requirements, Synopsys' FPGA Compiler II introduces block-level incremental synthesis (BLIS) to allow design engineers to modify a subset of a design and then resynthesize just the modified subset.

To address these requirements, Synopsys' FPGA Compiler II introduces block-level incremental synthesis (BLIS) to allow design engineers to modify a subset of a design and then resynthesize just the modified subset. A design can be divided into blocks, which is the smallest subset to which BLIS can be applied. FPGA Compiler II generates an optimized netlist for each block. A block's netlist does not change unless the design associated with that block has been modified. The netlist of each block is then presented individually to the place-and-route tool, which can recompile only the modified netlists. Not only does this capability increase the likelihood of preserving post-place-and-route timing behavior for the unmodified blocks, overall compilation time for both synthesis and place-and-route is significantly reduced.

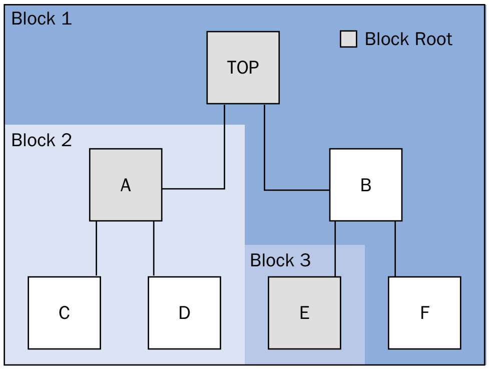
Identifying Blocks

A block can be a Verilog HDL module, a VHDL entity, an EDIF netlist, or a collection of all as long as they form a tree within the design hierarchy. The top-level module/entity/netlist of this tree is by definition the block root. The components of a block include the block root and all modules/entities/netlists in its tree of the design hierarchy that do not include another block root.

For example, the top-level design TOP in Figure 1 has two subdesigns, A and B. A instantiates C and D. B instantiates E and F. By definition, the top-level design TOP is a block root. If the user decides to also designate A and E as block roots, then the entire design will have three blocks:

- Block 1 : {TOP, B, F}
- Block 2 : {A, C, D}
- Block 3 : {E}

Figure 1. Block & Block Roots Example



Modifications to any module/entity/netlist contained in a block cause the entire block to be resynthesized. For example, if F is modified, then every member of Block 1 (TOP, B, F) will be resynthesized even though TOP and B have not been changed.

Blocks can be identified in both the graphical user interface (GUI) and the shell of the FPGA Compiler II. In the GUI, right-click an elaborated implementation in the **Chips** window and select **Edit Constraints**. Select the **Modules** tab to display the **Modules Constraint Table**. You can then specify any subdesigns as block roots in the **Block Partition** column. To remove a block root designation, click the particular cell and select **Remove**. The top-level design is always identified as a block root by definition and it cannot be removed. Figure 2 shows the **Modules Constraint Table**.

In the shell, use the command `set_module_block` followed by the option true and the path to the module/entity/netlist to be specified as the block root. For example:

```
fc2_shell> set_module_block true
/TOP/A
```

In the interactive mode, users can remove any block root designations by using the false option. For example:

```
fc2_shell> set_module_block
false /TOP/A
```

In batch mode, the same result can be achieved simply by removing the `set_module_block` commands associated with the block roots to be removed in the `fc2_shell` scripts.

Refer to the man page of `set_module_block` for additional usage and syntax information. To access the man page, type:

```
fc2_shell> man set_module_block↵
```

Block roots can only be designated on user-created modules/entities/netlists. Attempting to modify the setting on a primitive or the top-level design will result in the following error message:

```
Error: Cannot set block option
on module 'AND'
```

Furthermore, the concept of block and block root only applies when the target architecture supports BLIS. Attempting to apply this feature on an architecture that BLIS does not support will result in the following error message:

```
Error: block assignments are not
supported for the target
technology of this chip
```

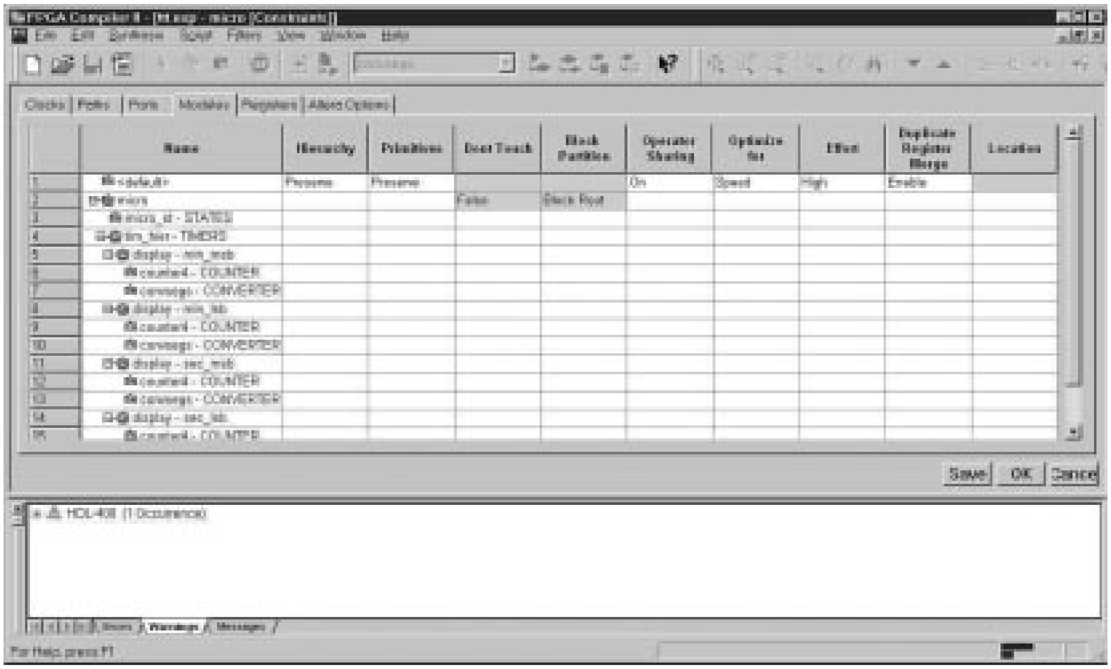
BLIS is currently available for Altera® APEX™, APEX II, ACEX™, Mercury™, and Excalibur™ architectures.

Design Planning

The advancement in synthesis and place-and-route technologies plays an important role in quality of results (QOR), but thorough design

continued on page 28

Figure 2. Modules Constraint Table



	Name	Hierarchy	Primitives	Dont Touch	Block Partition	Operator Sharing	Optimize for	Effort	Duplicate Register Merge	Location
1	<default>	Preserve	Preserve			On	Speed	High	Enable	
2	micro			False	Block Root					
3	micro_sl - STATES									
4	tim_tmr - TIMERS									
5	display - min_msb									
6	counter4 - COUNTER									
7	convsegs - CONVERTER									
8	display - min_lsb									
9	counter4 - COUNTER									
10	convsegs - CONVERTER									
11	display - sec_msb									
12	counter4 - COUNTER									
13	convsegs - CONVERTER									
14	display - sec_lsb									
15	counter4 - COUNTER									

HDL-408 (1 Occurrence)

BLIS for Altera: Using Block-Level Incremental Synthesis in Synopsys FPGA Compiler II, continued from page 27

planning can never be replaced. To take full advantage of BLIS, it is important to understand the following:

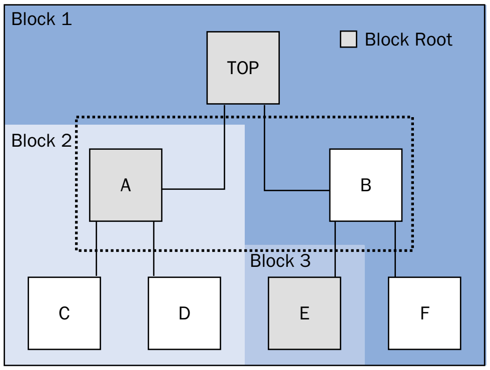
- FPGA Compiler II uses a time stamp to determine if an implementation is out-of-date. If the time stamp of any of the analyzed design source files is newer than the elaborated implementation, then the elaborated implementation needs to be updated.
- A block is the smallest subset to which BLIS can be applied and any changes in any member of a block cause the entire block to be resynthesized.
- FPGA Compiler II does not optimize across blocks.

BLIS is most effective when modifications are done within a module/entity. Changing the partition or the pins of the modules/entities of the design causes the entire design to be resynthesized.

Therefore, it is recommended that each module/entity/netlist in a design be described in its own design source file. Doing so ensures that modifying a module/entity/netlist will not affect the time stamp of other modules/entities/netlists that could potentially be members of other blocks.

In Figure 3, A and B are described in the same design source file. Modifying A not only causes the entire Block 2 to be resynthesized, it also affects the time stamp of B because it is in the same design source file as A. The newer time stamp of B causes the entire Block 1 to be resynthesized.

Figure 3. Design Planning Example



When all modules/entities/netlists are described in a single design source file, the BLIS cannot be used.

Because the FPGA Compiler II does not incrementally optimize across block boundaries, it is important not to break combinatorial logic into different blocks. For best QOR, use conventional register transfer level (RTL) coding style when partitioning designs. This style groups related combinatorial logic within a module/entity and registers all outputs of each module/entity.

BLIS is most effective when modifications are done within a module/entity. Changing the partition or the pins of the modules/entities of the design causes the entire design to be resynthesized. Therefore, thorough planning in the early stage of the design process is vital to success.

Implementation Update

This section describes how to update an implementation after a design has been modified. It is assumed that blocks have been defined as described previously, an optimized implementation has been created, and place-and-route netlists have been generated. If you are not familiar with the synthesis process of the FPGA Compiler II, refer to the ***FPGA Compiler II Getting Started Manual*** for step-by-step instructions.

In FPGA Compiler II, a red question mark next to a design source file in the **Design Sources** window indicates that the file has been modified. Right-clicking the icon for the modified file and selecting **Update File** reanalyzes the file for syntax errors.

Similarly, a red question mark next to an elaborated implementation in the **Chips** window indicates that the implementation is out-of-date with respect to its design source files. Right-clicking the icon for the elaborated implementation and selecting **Update Chip** re-elaborates the implementation.

At least two blocks must be defined to enable BLIS. Once the design has been re-elaborated, right-clicking the icon for the optimized implementation and selecting **Update Chip** reoptimizes the design. If blocks are properly defined as described previously, BLIS is automatically enabled. In this situation, only the blocks that are associated with modified files are re-optimized.

Instead of updating the design source files, the elaborated implementations, and the optimized implementations individually, users can update the entire project in one step by right-clicking the project icon in the **Design Sources** window and selecting **Update Project**. The FPGA Compiler II generates one netlist in EDIF format for each block defined. The names of the netlists are the same as the module/entity/netlist name of the respective block roots. Selecting **Update Project** ensures that only the netlists associated with modified blocks are regenerated. Explicitly choosing **Export Netlist** forces all netlists to be regenerated. This step is therefore only recommended when the design is synthesized for the first time.

Figure 4 shows a sequence of equivalent fc2_shell commands to be used in the interactive mode. Refer to the man pages for complete usage and syntax information.

Limitations

As mentioned previously, time-stamping is used to determine if an implementation is out-of-date. The FPGA Compiler II processes time stamps in whole seconds. If both analysis and elaboration finish within one second, the analyzed design source files and the elaborated implementation will have the same time stamp. When the time stamp of the elaborated implementation is older than or equal to the analyzed design source files, the existing elaborated implementation is discarded. Select **Update Chip** or **Update Project** to re-elaborate the implementation. The new elaborated implementation will then have a newer time stamp than the existing optimized implementation. This newer time stamp causes the existing optimized implementation to be discarded and reoptimized when none of the design source files have been modified.

Figure 4. fc2_shell Commands

```
create_project TOP

add_file -library WORK -format VHDL c:/designs/top/TOP.vhd
add_file -library WORK -format VHDL c:/designs/top/A.vhd
add_file -library WORK -format VHDL c:/designs/top/B.vhd
add_file -library WORK -format VHDL c:/designs/top/C.vhd
add_file -library WORK -format VHDL c:/designs/top/D.vhd
add_file -library WORK -format VHDL c:/designs/top/E.vhd
add_file -library WORK -format VHDL c:/designs/top/F.vhd

analyze_file
```

Most designs take longer than one second to be analyzed and elaborated, so the above-mentioned limitation should not present any problems for users.

BLIS is driven only by the difference in time stamps, the existence of at least two defined blocks, and any change of block roots. Other operations, such as modifying timing constraints, do not cause block-level incremental resynthesis of the implementation.

Conclusion

Synopsys' FPGA Compiler II introduces BLIS, allowing design engineers to modify a subset of a design (called a block), and resynthesize just the modified block. FPGA Compiler II and FPGA *Express* generate an optimized netlist for each block; the netlist of a block does not change unless the design associated with that block has been modified. The place-and-route tools that support block-level incremental place-and-route are able to recognize and recompile only the netlists that have been changed. Not only does this capability increase the likelihood of preserving post-place-and-route timing behavior for the unmodified blocks, but overall compilation time for both synthesis and place-and-route can be significantly reduced.

For the latest in FPGA synthesis technology, visit **http://www.synopsys.com/products/fpga**.

Mentor Graphics' Inventra IPX—Accelerating SOPC Design

Using third-party intellectual property (IP) functions in programmable logic design has graduated from theory to reality. Designers are no longer asking why they should consider using a proven block developed by someone else. Instead, the focus is on establishing appropriate IP selection criteria that enables users to efficiently leverage the performance and capacity of the programmable logic devices (PLDs) while meeting their design schedules.

Introducing Inventra IPX

Inventra™ IPX™, newly released by the Inventra IP division of Mentor Graphics®, was developed with the express purpose of easing the delay and confusion often associated with acquiring, evaluating, and integrating third-party IP for system-on-a-programmable-chip (SOPC) solutions. Delivered as a suite of 10 proven Inventra IP functions, Inventra IPX contains key building blocks for communications, computing, and consumer applications. Inventra IPX includes the following cores:

Inventra IPX leverages the Altera® encrypted IP flow and its tight integration with the Mentor Graphics LeonardoSpectrum™ synthesis tool.

- M8051 8-bit microcontroller
- MI2C I^{2}C bus interface
- M16550A UART with first-in first-out (FIFO) buffers
- M16X50 enhanced UART with FIFO and IrDA
- MPCMCIA1 PC card interface
- M82365SL PC host interface
- M1284H IEEE 1284 host parallel port
- M8237A 4-channel DMA controller
- M8255 parallel peripheral interface
- M8490 5380-compatible SCSI interface

Extensive IP Evaluation

Inventra IPX leverages the Altera encrypted IP flow and its tight integration with the Mentor Graphics LeonardoSpectrum™ synthesis tool. This encrypted IP flow enables extensive evaluation of the functions both in isolation and as an integrated portion of the full design. A selected Inventra IPX function is rapidly optimized using LeonardoSpectrum resulting in an encrypted EDIF netlist. The netlist is then imported to the Quartus® II software for place-and-route and integration into the design. The IPX cores can be used throughout this entire flow under an evaluation license. A full license is required before programming files and unencrypted gate-level netlists can be generated.

Strong Performance

The functions were chosen for inclusion in Inventra IPX based on their proven nature with multiple licensees as well as the strong performance achieved when targeting Altera® PLDs. Sample performance for several of the functions targeted to APEX 20KE devices is shown in Table 1.

Table 1. Sample Function Performance in APEX 20KE Devices

Function	f_{MAX}	Logic Cells
M8051	40	2,052
M16550A	111	434
M8255	142	144

Low-Cost Subscription Licensing

The Inventra IPX core suite is offered under a one-year renewable subscription model at $5,500 per seat per year. Designers can select and implement any of the IPX cores into their designs throughout the year. There are no royalties or component use fees. A click-wrap license agreement, specifically developed for Inventra IPX, is built into the product installation.

Accelerated SOPC Development

Faster access to a suite of cores for unlimited annual use results in less time spent searching and negotiating contracts for third-party IP cores. Coupled with an encrypted IP evaluation flow, Inventra IPX accelerates SOPC development by allowing designers to thoroughly exercise the IP.

For more information, visit **http://www.mentor.com/inventra/ipx**.

Tality Introduces New APEX-Based Development & Prototyping Platform

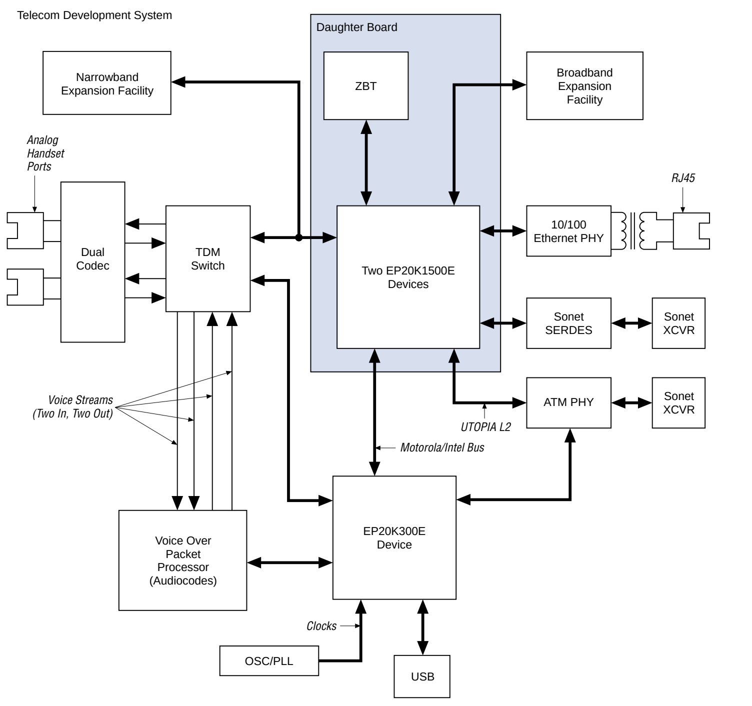
Tality Corporation, the world's largest independent provider of engineering services and IP for the design of complex electronic systems, and one of Altera's Certified Design Center partners, recently announced the availability of a new development and prototyping platform based on Altera® devices.

Tality's Telecom Development System (TDS) is a cost-effective, fully portable and ready-to-use solution that enables designers to test algorithms and prove concepts at an early stage in the design cycle using real hardware. The TDS comes with a wide variety of datacom/telecom interfaces that allow developers to verify interoperability and can also be customized and used for either production verification or as an evaluation platform (see Figure 1).

Gregory Bowes, Director of Business Management and Operations, said that "the TDS is a very exciting product that is an ideal solution for development projects such as packet classification, voice over Internet protocol (VoIP), voice over asynchronous transfer mode (VoATM), ATM to Ethernet bridging, and ATM cell handling. We believe the TDS will provide a substantial advantage to designers in both shortening time-to-market and reducing development costs." The TDS consists of four main features:

- A daughter board that hosts two Altera APEX™ EP20K1500 programmable logic devices (PLDs) that allow developers to load and test HDL algorithms and prove concepts in real hardware
- A motherboard containing a number of common datacom/telecom interfaces that enable designers to rigorously test designs by simulating real-world applications
- Expansion slots that facilitate the addition of other broad- or narrow-band interfaces
- An easy-to-use, PC-based Windows application with a graphical user interface (GUI) that allows designers to develop, download and demonstrate new designs on the daughter board from a PC or laptop host

Figure 1. APEX-Based Development & Prototyping Platform



Three Million Gates of Programmable Logic

The TDS is centered around two Altera APEX EP20K1500 devices that provide a total of three million gates of programmable logic. In addition, the daughter card has eight Mbytes of zero-bus turnaround (ZBT) RAM. This combination provides sufficient capacity to demonstrate even the most complex broadband applications such as custom digital signal processing (DSP) functions, segmentation and reassemblies (SARs), and voice transcoders. The PLDs can be loaded from a host PC through a universal serial bus (USB) port, or directly through a Joint Test Action Group (JTAG) port on the daughter card.

One PLD is connected to the media independent interface (MII) (10/100 Ethernet), a parallel interface to the serializer/deserializer (SERDES), UTOPIA Level 2 to ATM POS/PHY, and the Broadband Expansion Facility. The

continued on page 32

Tality Introduces New APEX-Based Development & Prototyping Platform, continued from page 31

second PLD is connected to eight Mbytes of ZBT RAM, a TDM switch, and the Narrowband Expansion Facility.

The interconnect bus between the two APEX devices consists of 64-bit data, 22-bit address, and 16-bit control buses.

DC/TC Interfaces Can Be Used to Verify Interoperability

The motherboard is controlled by an APEX EP20K300E device and contains a number of common datacom/telecom peripherals and features including:

- A 10/100 Ethernet physical layer (PHY) interface for demonstrating features such as VoIP through a LAN connection
- Two OC-3 transceivers, one dedicated to a SERDES and the other to an ATM/POS-PHY
- Two analog handset ports interfaced to dual coder/encoders for checking voice algorithms
- UTOPIA Level 2 interface to the ATM/POS-PHY
- Audiocodes Voice over Packet Processor for voice compression
- USB port and 8051 microprocessor chip for interfacing with the GUI and PC
- TDM switch
- JTAG port connected to the PLDs

The motherboard also has a 12-V power port to plug in a wall power pack for full portability.

Add Other Broadband or TDM Interfaces

Tality can customize the TDS to include other broadband or TDM interfaces. Interface modules such as OC-12, Firewire, Gigabit Ethernet, Bluetooth, and xDSL are supported through the Broadband Expansion Facility, while the Narrowband Expansion Facility supports interfaces to industry-standard TDM equipment (e.g., T1/E1, T3/E3, J1).

The Broadband Facility has 16 pairs of LVDS, 50 single-ended I/O signals, and a 120-pin connector, while the Narrowband Facility has 50 single-ended I/O signals and a 120-pin connector. Both expansion facilities can be powered by 2.5, 3.3, or 5.0 V from the motherboard.

Portable & Easy-to-Use Solution

Adding to the convenience of the TDS is a USB port and GUI that enable a PC to be connected and used to develop, download (without any other external hardware), and demonstrate new designs on the daughter board programmable logic. The easy-to-use GUI runs on Windows 98 and Windows 2000. The software provides direct access to all registers on the main board as well as user-designed registers and memory on the daughter board. Motherboard registers can also be accessed using sliders and other control methods.

Tality's TDS software is generic enough for any design, can be customized further by Tality if required, and comes with a basic script interpreter to aid in setting up devices on the boards.

About Tality

Tality provides a broad range of communications product design services and IP, including complete system and sub-system, and both hardware and software. The company focuses primarily on network infrastructure, wireless infrastructure, broadband access devices, consumer communications, and "infotainment products." Tality's services and IP span product conceptualization through implementation for production manufacturing, enabling customers to get better products to market faster.

The MicroBlaster Source Code: An Embedded Configuration Solution

The capability to configure Altera® programmable logic devices (PLDs) on an embedded platform is in high demand among programmable logic users. An embedded platform is less expensive and does not involve sophisticated setup. Among all the supported configuration modes, the passive serial configuration is the easiest and the most direct method of configuring a PLD. Altera developed the MicroBlaster™ source code to address these needs.

The MicroBlaster Source Code

The MicroBlaster source code is a software driver that allows you to configure Altera PLDs in passive serial mode and is targeted to embedded configuration. The MicroBlaster source code was developed and tested on the Windows NT platform; the source code is written in modules making it easy to customize or modify the I/O control routines to fit into your system.

The MicroBlaster source code supports raw binary file (**.rbf**) format, generated by the Quartus® II software.

The selected Altera device RBF sizes are shown in Table 1. The size of the MicroBlaster binary file is approximately 40 Kbytes for Windows NT.

Table 1. SOF & RBF File Size Comparison

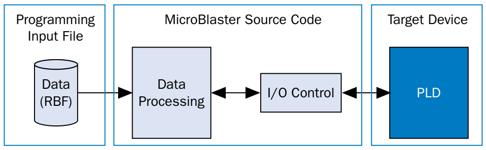
Device	SOF (Kbytes) *(1)*	RBF (Kbytes)
EP20K30E	42	44
EP20K200E	239	241
EP20K400E	473	478
EP20K1500E	1,462	1,471

Note to Table 1:
(1) SOF: SRAM Object File.

The Interface

The MicroBlaster source code block diagram is shown in Figure 1. The MicroBlaster source code can be divided into two modules, the data-processing module and the I/O control module. The data-processing module reads programming data from the RBF, rearranges it, and dumps it to the I/O control module. The I/O control module feeds the target PLD with the data received from the data processing module.

Figure 1. MicroBlaster Source Code Block Diagram



Periodically, the I/O control module senses certain configuration pins to determine if any errors take place during the configuration process. When an error occurs, the MicroBlaster source code can reinitiate the configuration process.

The MicroBlaster source code is a software driver that allows you to configure Altera PLDs in passive serial mode and is targeted to embedded configuration.

I/O Control

Two separate routines handle the read and the write operations in the I/O control module. The read operation reads the value of the required pin; the write operation writes data to the required pin. These I/O routines are platform-dependent and are the only portions of code that need to be customized.

Porting

When porting the MicroBlaster source code to an embedded system, you only have to add your customized I/O control routines to the read/write functions of the MicroBlaster source

continued on page 34

The MicroBlaster Source Code: An Embedded Configuration Solution, continued from page 33

Table 2. Pin Assignment of the Passive Serial Configuration Signals in the Parallel Port

Port/Bit	7	6	5	4	3	2	1	0
0		DATA0					nCONFIG	DCLK
1	CONF_DONE			nSTATUS				
2								

code. To reduce the modification needed (when writing), your I/O control routine should map the port value that is defined in the parallel port architecture to the I/O port definition of your system, as shown in Table 2.

Likewise, when reading from the I/O port, your I/O control routine should map the I/O port definition of your system to the port value that is defined in the parallel port architecture, as shown in Table 2. An example of reading and writing mapping processes is shown in Figure 2.

Figure 2 shows an embedded system mapping five configuration signals to the data registers `D0`, `D1`, `D3`, `D6`, and `D7` of an embedded microprocessor. When reading from the I/O ports, the I/O control routine reads the values of the data registers and maps them to the particular bits in the parallel port registers (`P0` to `P2`). These bits are later accessed and processed by the data-processing module.

When writing, the values of the signals are stored in the parallel port registers (`P0` to `P2`) by the data-processing module. The I/O control routine then reads the data from the parallel port registers and dumps it to the corresponding data registers (`D0`, `D1`, `D3`, `D6`, and `D7`).

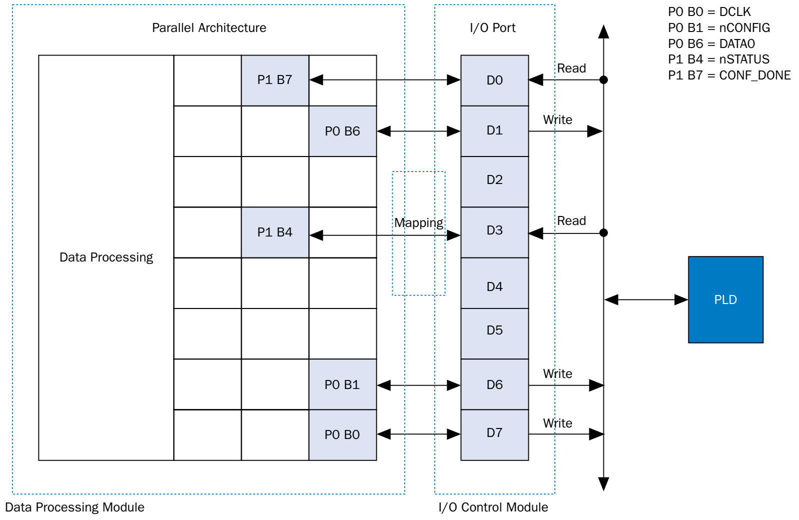
Customizing the I/O control routines through mapping can save a lot of work in modifying the source code. All you have to do is add your I/O control routines in the I/O control module.

Conclusion

The MicroBlaster source code is written modularly so you can easily port it to other platforms. It offers an inexpensive and unsophisticated embedded system to accomplish passive serial configuration for Altera PLDs.

A reference design is available on the Altera web site at **http://www.altera.com.**

Figure 2. MicroBlaster Source Code Block Diagram



Using Mercury Devices in High-Speed Serial Backplanes

This article provides an overview of the importance of the serial backplane in today's cutting-edge technologies and illustrates how Mercury™ devices provide a system-on-a-programmable-chip (SOPC) solution for these applications.

Backplane Performance an Essential Part of Leading-Edge Technology

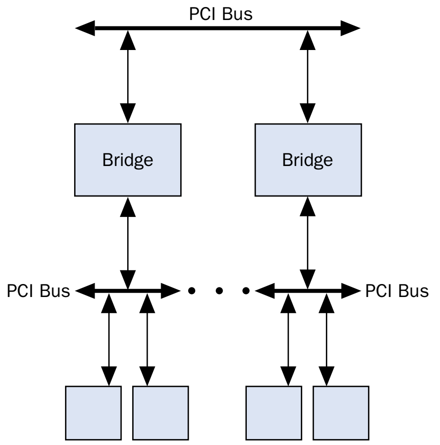
The growth of the Internet has driven a bandwidth explosion in today's network infrastructures. With the local area network (LAN) semiconductor industry alone projected to grow to over $7.5 billion by 2004 (Dataquest, November 2000), this infrastructure market will be a viable one for many years.

A massive amount of data is transmitted within these infrastructures between the various boxes that connect them. Once inside these boxes, the data is then distributed and processed among the many different cards, which are all connected through a backplane. As a result, the backplane can easily become the bottleneck for these boxes, and ultimately for the entire system. A key challenge in today's infrastructure market is for designers to keep pace with the ever-increasing demand for bandwidth and to alleviate this congestion in the backplane.

Backplane Topologies

Backplanes have traditionally used parallel interfaces such as peripheral component interconnect (PCI) to transmit data between different boards (see Figure 1). However, as data rates and port counts rise and congestion overwhelms the backplane, high-speed serial interconnects are fast becoming the standards of choice. For example, a backplane design that requires a bandwidth of 10 gigabits per second (Gbps) would require a hefty 75 PCI-X channels at 133 MHz if implemented using a traditional bus architecture. That same design, implemented with a serial backplane approach, would use just 10 serial channels at 1.0 Gbps. Imagine that this requirement is for one line card, and that there are 10 cards in a rack, and two racks on a shelf. Clearly, traditional parallel backplane architectures cannot be efficient at these bandwidth levels.

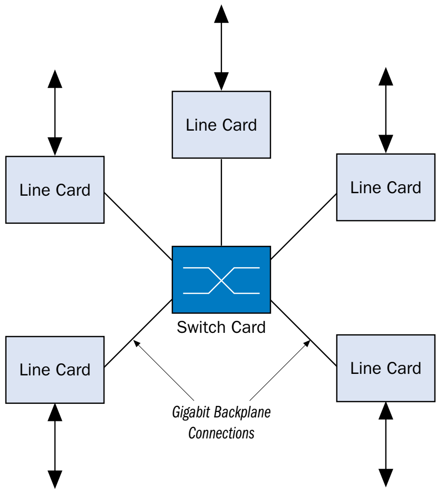
Figure 1. Traditional PCI-Based Backplane Architecture



Serial backplane architectures, as shown in Figure 2, offer advantages in pin count and noise reduction, while dramatically improving data rates. Clock data recovery (CDR), a technique that is widely used in serial backplane signal transmissions, removes skew concerns and further increases data transfer rates.

continued on page 36

Figure 2. Star Topology Switch Architecture



Using Mercury Devices in High-Speed Serial Backplanes, continued from page 35

Altera’s Mercury family allows customers to meet the challenges in this leading-edge market, relieving bottlenecks and empowering the systems to run at full capability.

Serial Backplane Applications

In the network infrastructure market, Mercury devices are particularly well-suited where there is an application space for serial backplanes. In the wide-area network (WAN), these applications consist of backbone terabit switches and routers, traditional digital cross-connects, and dense wave division multiplexing (DWDM) cross-connects. In a LAN, these applications include high-bandwidth, high-port-count ATM, Ethernet, and Fibre Channel switches.

While Mercury devices are ideal for the serial backplanes, they are also fully compliant with Gigabit Ethernet, Fibre Channel, and a host of other standards, making them viable solutions for line-side applications (see Figure 3).

Mercury Features Ideal for the Serial Backplane

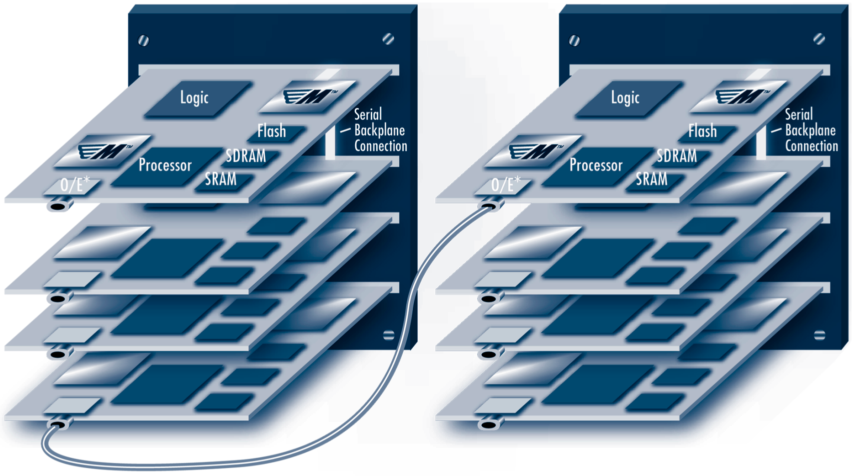
This section describes the many features that make Mercury devices an ideal solution for serial backplane applications.

CDR

Mercury devices offer the strongest CDR capability available in the industry. CDR circuitry embeds the clock in the data stream at the transmitter and recovers it at the receiver, thereby removing skew concerns between the clock and data lines, reducing trace counts, and ultimately enabling faster data rates. Mercury devices provide up to 18 full-duplex CDR channels with performance up to 1.25 Gbps, for a total bandwidth of 45 Gbps, giving users full flexibility in implementing a wide array of serial connections and in speeding up the backplane bottleneck.

Mercury CDR has programmable data rates and supports any data rate from 125 Mbps to 1.25 Gbps. For example, this range allows a designer to implement an 8-channel, 622-Mbps serial backplane interface today, and transition to an 8-channel 1.25-Gbps interface tomorrow, all by downloading a new programming file. Mercury CDR can also simultaneously support two different data rates, enabling an engineer to design a 622-Mbps chip-to-chip interface and a 1.25-Gbps backplane interface, all on a single Mercury device. Mercury devices also implement multiple differential I/O standards such as LVDS, LVPECL, and PCML.

Figure 3. Serial Backplanes



PLLs

Mercury devices include two high-speed phase-locked loops (PLLs) dedicated to generating the necessary reference frequencies to supplement CDR. These PLLs support input frequencies of up to 650 MHz and provide the ability to multiply the incoming clock by a factor of 1 to 20.

SERDES

Mercury devices include dedicated built-in serializer/deserializers (SERDES). This SERDES functionality completes Mercury CDR transceivers, giving users the ability to serialize or deserialize the data by a factor of 3 to 20. This SERDES factor is uniquely controlled and is not directly associated with the PLL multiplication factor.

This ability to have separate clock multiplication and data serialization factors provides designers full flexibility in their high-speed serial designs. For example, a system may contain a 1-Gbps serial data stream across the backplane, and on the receiving board, require deserialization by a factor of 8. Since the clock and SERDES factors do not have to be the same, the designer has the option to use a 100-MHz clock and a multiplication by 10, or a 50-MHz clock and a multiplication by 20, to generate the reference clock. Additionally, the separate clock and SERDES factors in Mercury devices make it possible to support standards such as RapidIO, which requires the incoming clock to be half the data rate, or SFI-4, which requires a 622.08-MHz clock with 622.08-Mbps data.

Synchronizer FIFO Buffers

The Mercury CDR transceiver also includes dedicated first-in first-out (FIFO) buffers for each channel. Serial backplane applications commonly result in multi-crystal operation, resulting in CDR channels driven by different sources, and come in at slightly varying rates. The built-in FIFO buffers serve to synchronize each channel so that the designer can use one global clock to handle all of the data in the logic function.

Performance-Optimized Function

Mercury devices feature an architecture that is built for bandwidth. This innovative architecture runs at an average of 57% faster than APEX^{TM} devices fabricated on an equivalent process, giving customers the required power to process the high bandwidth of data coming through the device. Even in its 0.18-µm aluminum form, this architecture is faster than competing architectures that are processed on 0.15-µm copper technology.

The integration of a SERDES and this high-performance array logic provides a complete SOPC solution. Designers can combine a transceiver and a separate logic device into one device, removing the need to route signals from chip-to-chip, and thus eliminating board traces and signal integrity issues.

This ability to have separate clock multiplication and data serialization factors provides designers full flexibility in their high-speed serial designs.

The programmable logic in Mercury devices allows customers to implement a wide range of different functions that may be crucial for their backplane design. To simplify system design, Altera provides the Mercury Gigabit Transceiver intellectual property (IP) function so designers can implement 8B/10B encoding/decoding and byte-alignment functions with the push of a button. Designers also have the option to implement their own proprietary design, a flexibility that only programmable logic can provide.

Conclusion

In today's rapidly evolving network infrastructure market, there is an endless need for more bandwidth. Design engineers are constantly faced with new challenges of providing high-bandwidth solutions with a fast time-to-market. Mercury devices meet these challenges. They boast CDR transceivers that are capable of speeds up to 1.25 Gbps and a total bandwidth of up to 45 Gbps. With up to 18 channels, these devices provide flexibility and scalability for customers in their designs. Combined with a rate-adaptive CDR, SERDES with built-in FIFO buffers, and a high-performance PLD function, it is evident that Mercury devices provide the SOPC solution for high-speed serial backplane designs.

Motorola Using the LogicLock Incremental Design Flow in the Quartus II Software

Available exclusively in the Quartus® II software, the LogicLock™ incremental design flow allows users to design, optimize, and lock-down a design one section at a time. Successfully using the LogicLock design flow requires defining a hierarchy, creating well-defined physical boundaries, and using registers to optimize and lock-down performance. This article discusses how the Motorola GTSS team used the LogicLock design flow to achieve their performance goals.

> *"The LogicLock design flow is a valuable addition to the Quartus II feature set. In a recent project involving Altera PLDs, the LogicLock design flow was critical in helping me achieve my performance goals."*
>
> George Powley,
> Senior Staff Engineer,
> Motorola GTSS

CDMA Filter Design

The Motorola GTSS team developed a code division multiple access (CDMA) filter design targeted to a programmable logic device (PLD). One module of the CDMA filter design called **channel_filter_chain,** utilizes about 10% of the device and requires an 80-MHz clock speed. The CDMA filter design contains four instantiations of the **channel_filter_chain** module and some additional control logic. Therefore, the PLD design was an excellent candidate for the LogicLock incremental design flow.

Partitioning the Design for LogicLock Flow

To prepare the design for the LogicLock incremental design flow, its entities must be designed with the LogicLock flow in mind. The following recommendations prepare your design for partitioning:

1. Use registered boundaries for all modules.
2. Do not use black-boxing modules for optimization across boundaries during synthesis.
3. The design should not cross clock domains at the module boundary.
4. Module output registers should be constrained to a LogicLock region.
5. If you are using the Floorplan Editor on an entire device, anticipate routing delays stemming from the lower-level module's output register to the respective destination modules.

These recommendations guide the Quartus II software to create modules with well-defined physical boundaries, similar to designing modules for an application-specific integrated circuits (ASICs).

When you are using the original design partition, the first recommendation suggests using registered boundaries. For example, in the original design, the output registers of **channel_filter_chain** were clocked at 80 MHz. In the next level of hierarchy, the output registers of **channel_filter_chain** were registered at 20 MHz. Crossing clock domains requires tight timing between the registers. Therefore, the 80-MHz registers should be placed as close as possible to the 20-MHz registers. Thus, the 20-MHz registers were moved into the same LogicLock region to improve the timing achieved at the top level.

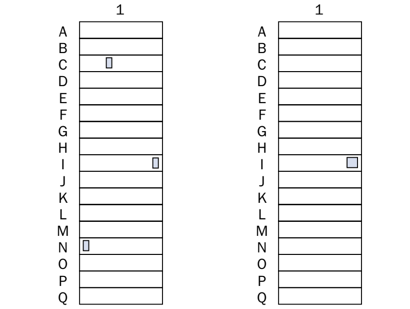
Often, signals at the module boundaries are buses. If the registers of the bus are physically spread across the module, it is difficult to achieve timing at the top level because the interconnect timing will be different for each bit in the bus. Therefore, constraining the registers to one region ensures that each bit of the bus has similar interconnect timing at the top level.

Figure 1 shows placement results with and without constraining the placement of output registers. Both designs use a 1 × 17 MegaLAB™ LogicLock region, but the design on the right constrains the output registers to a LogicLock region. Constraining output registers to the same physical location makes it easier for the top-level design to meet timing requirements.

Incremental Design-by-Example

Incremental design requires you to create an ATOM netlist, back-annotate placement of a module, and import the module into a top-level design. This process is relatively easy once the design is partitioned correctly. The following sections describe the design flow used for the example design.

Figure 1. Recommended Output Register Constraint Technique



Without Output Register Constraint (Not Recommended)

With Output Register Constraint (Recommended)

Synthesize the Module in a Synthesis Tool

Create a project directory for the module, synthesize it using LeonardoSpectrum™ or Synplicity, and generate an EDIF netlist.

Optimize the Module in Quartus II Software

Create a Quartus II project for the module and perform the following steps:

1. Turn off **Optimize I/O cell register placement for timing.** This action ensures that registers at the module boundary are not moved into the I/O cells. Choose **Compiler Settings** (Processing menu), then click the **Synthesis & Fitting** tab.
2. Enable the creation of an ATOM netlist. Choose **Compiler Settings** (Processing menu), then click the **Synthesis & Fitting** tab. Under **Incremental Synthesis**, turn on **Save a node-level netlist into a persistent source file**.
3. Create LogicLock regions for the design. For this design, one auto-sized floating region contains the entire module. A fixed-size floating region (a child of the larger region) contains the module output registers.
4. Compile the design, iterating steps 3 and 4 until the module meets timing requirements. This step may involve creative definition of LogicLock regions to meet timing requirements.

Export the Module & LogicLock Constraints

This step generates the files required to import the module into a top-level design. The ATOM netlist should be present as a Verilog HDL Quartus Mapped File (**.vqm**) in the **atom_netlists** directory. Follow these steps to generate an Entity Settings File (**.esf**) with the back-annotated placement information:

1. Back-annotate placement information for the module. Choose **LogicLock Regions** (Tools menu), and click **Back-Annotate Contents.**
2. Export the ESF to the **atom_netlists** directory. Choose **LogicLock Regions** (Tools menu), click **Export,** and choose the **atom_netlists** directory.

Import all Modules & LogicLock Constraints into the Top-Level File

Use the VQM and ESF to import the module into a top-level design (first follow the steps below):

1. Create a register transfer level (RTL) black-box for the **channel_filter_chain** module (top-level netlist).
2. Synthesize the top-level in LeonardoSpectrum using the black-box for the **channel_filter_chain** module.
3. Generate an EDIF netlist.
4. Copy the **channel_filter_chain** VQM and ESF into the top-level Quartus II project directory (i.e., **par**).
5. Click **Start Analysis & Elaboration.**
6. Import the LogicLock regions. Choose **LogicLock Regions** (Tools menu) and Click **Import.**
7. Leave the imported LogicLock regions floating, or locked to a location.
8. Compile and verify the top-level design.

By optimizing one module at a time, the designer achieves decreased compile times for the top-level module, more consistent results for the top-level module over many iterations, reduced optimization efforts, and better control over each module (see Table 1).

Table 1. CDMA Filter PLD Compilation Results

Description	Without LogicLock Flow	With LogicLock Flow
Top-level compile times	2.5 hours	1.25 hours
Top-level f_{MAX}	65 MHz	90 MHz
One filter compile times	20 minutes	8 minutes
One filter f_{MAX} simple LogicLock flow	60 MHz	80 MHz
One filter f_{MAX} scripted LogicLock flow	60 MHz	91 MHz

MAX 7000B Device Support for Multiple Voltages Drives Innovative New Applications

System boards are moving from single to multiple voltage supplies on board. This movement is largely to accommodate the diverse voltage supply requirements of processors, application-specific integrated circuits (ASICs), and complex look-up table (LUT)-based programmable logic devices (PLDs). Voltages are progressing from 5.0 and/or 3.3 V to a mixture of 1.8, 2.5, 3.3, or 5.0 V (and in some cases, 1.0 V and lower).

Product-term-based PLDs are used in applications such as memory address decoding, boot logic, controlling state machines, as well for implementing glue logic. The use of product-term devices spans market segments, causing these devices to be commonly used on the majority of system boards.

Product-term based PLDs are increasingly used in several boards for various applications—memory address decoding, boot logic, controlling state machines, as well for implementing glue logic.

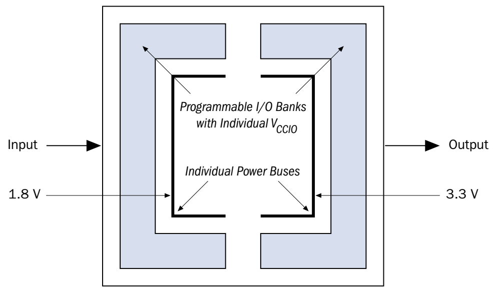
Since product-term devices interface with a variety of processors, ASICs, memory chips, and complex LUT-based PLDs that are present on board, they must have the capability to interface with multiple voltage levels.

Support for Multiple Voltage Levels

A product-term device interfacing with a 1.8-V processor on one side and providing the decoded address for a 3.3-V memory needs I/O pins capable of interfacing with multiple voltage levels.

Each I/O pin can only support one voltage level, depending on the value of the V_{CCIO} power signal. With MAX® 7000B devices, Altera has split the I/O pins into two banks with each bank powered by an independent V_{CCIO}. This split provides a high level of flexibility in configuring device I/O pins to support more than one voltage level (see Figure 1).

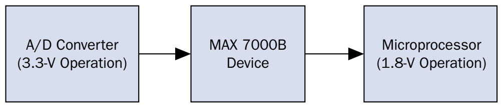
Figure 1. Split I/O Bank Architecture



With the V_{CCIO} set at the appropriate level, the I/O pins within each bank can be configured to support a voltage level independent of the I/O pins in the other bank.

This flexibility allows a single MAX 7000B device to talk with devices running at different operating voltages (see Figure 2).

Figure 2. Interfacing with Multiple-Voltage Levels


A/D Converter (3.3-V Operation)
MAX 7000B Device
Microprocessor (1.8-V Operation)

Conclusion

It is clear that we are quickly moving into a world where multiple voltage levels on a board is common. The product-term devices used in many of these boards for implementing the common glue logic functionality need to interface with the multitude of voltage levels.

With an innovative I/O bank architecture, that supports multiple voltage levels, MAX 7000B devices promise to enable various new designs.

Altera Launches mySupport—Web-Based Technical Support

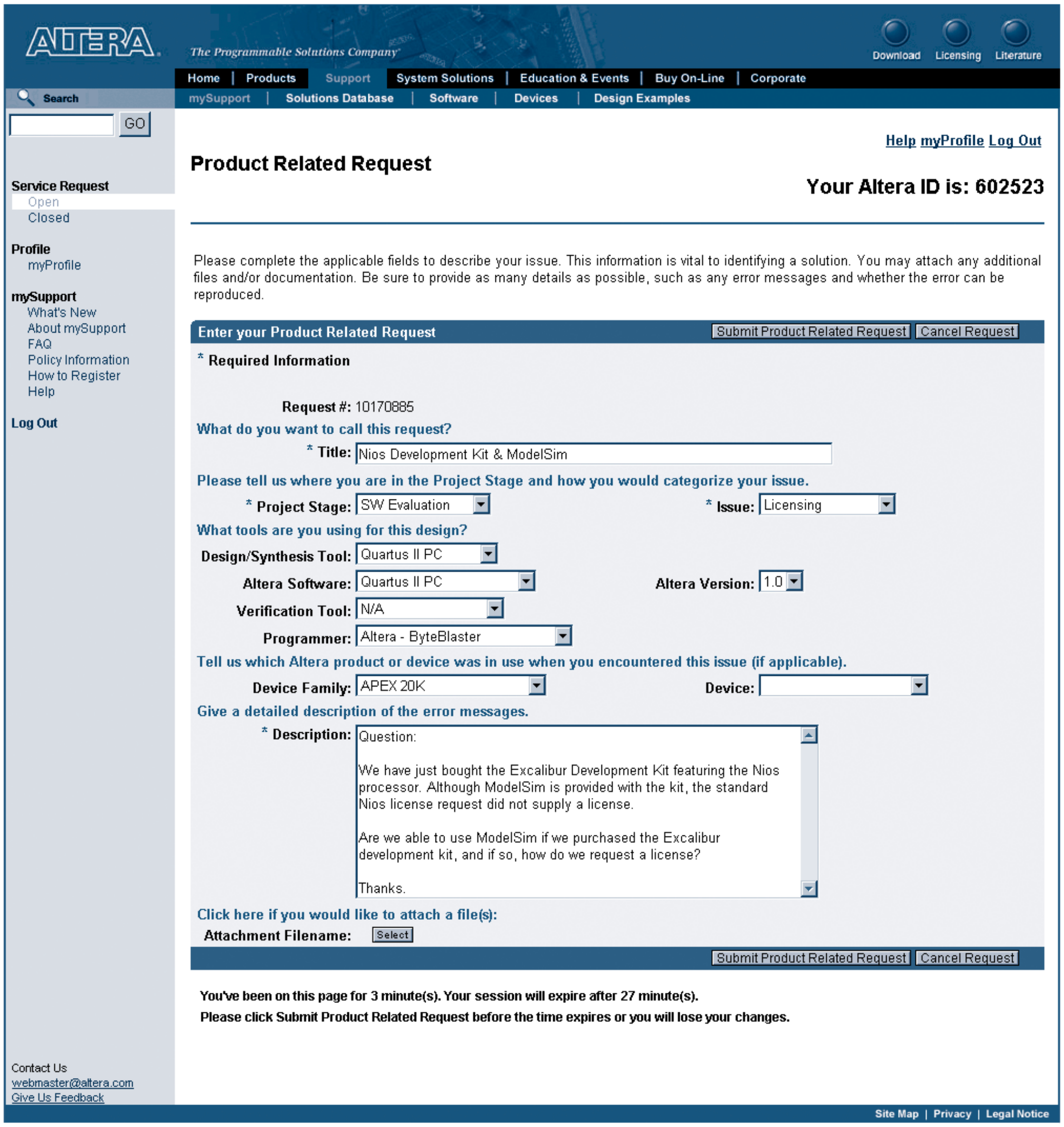
Altera has implemented a web-based issue submission and tracking application called mySupport. This web-based functionality allows new and existing Altera® designers to receive on-line technical support that is integrated seamlessly with the customer applications call center.

The features of mySupport include the ability to enter service requests related to products (see Figure 1), enhancements, and documentation. When you enter your service request, it will automatically be routed to an available application engineer.

You will be able to view and receive updates on your service requests, regardless of how you initially contacted Altera (web or phone). The mySupport web site allows you to add updates to your service requests at any time and also provides the ability to attach files to your service request. When service requests are updated, you will be notified in real time.

Register for mySupport at **http://www.altera.com/mysupport.**

Figure 1. The mySupport Web Site


ALTERA
The Programmable Solutions Company
Download
Licensing
Literature
Home | Products | Support | System Solutions | Education & Events | Buy On-Line | Corporate
Search
GO
mySupport | Solutions Database | Software | Devices | Design Examples
Service Request
Open
Closed
Profile
myProfile
mySupport
What's New
About mySupport
FAQ
Policy Information
How to Register
Help
Log Out
Help myProfile Log Out
Submit Product Related Request
Cancel Request
Select
Contact Us
webmaster@altera.com
Give Us Feedback
Site Map | Privacy | Legal Notice

Product Related Request

Your Altera ID is: 602523

Please complete the applicable fields to describe your issue. This information is vital to identifying a solution. You may attach any additional files and/or documentation. Be sure to provide as many details as possible, such as any error messages and whether the error can be reproduced.

Enter your Product Related Request

* Required Information

Request #: 10170885

What do you want to call this request?

* Title: Nios Development Kit & ModelSim

Please tell us where you are in the Project Stage and how you would categorize your issue.

* Project Stage: SW Evaluation

* Issue: Licensing

What tools are you using for this design?

Design/Synthesis Tool: Quartus II PC

Altera Software: Quartus II PC

Altera Version: 1.0

Verification Tool: N/A

Programmer: Altera - ByteBlaster

Tell us which Altera product or device was in use when you encountered this issue (if applicable).

Device Family: APEX 20K

Device:

Give a detailed description of the error messages.

* Description: Question:

We have just bought the Excalibur Development Kit featuring the Nios processor. Although ModelSim is provided with the kit, the standard Nios license request did not supply a license.

Are we able to use ModelSim if we purchased the Excalibur development kit, and if so, how do we request a license?

Thanks.

Click here if you would like to attach a file(s):

Attachment Filename:

You've been on this page for 3 minute(s). Your session will expire after 27 minute(s).
Please click Submit Product Related Request before the time expires or you will lose your changes.

Contact *Information*

How to Contact Altera

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

Information Type	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**	(408) 544-7000 (7:30 a.m. to 5:30 p.m. Pacific Time) *(2)*
	(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 a listing.