Cahners
MICROPROCESSOR REPORT
www.MPRonline.com
THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

# FPGA Processor Cores Get Serious

*FPGA Embedded Processors Set to Flood High-End Mainstream*

***By Cary D. Snyder {9/18/00-01}***

Enthusiastic competitors in the Field Programmable Gate Array (FPGA) business have leaped into the system-on-a-chip (SOC) market and are promising to speed delivery on simple, fast, and easy-to-drive SOC utility vehicles. The 2001 embedded-processor FPGA models will gain a strong following, as established worldwide dealers will quickly stock showroom floors with both economy and luxury-powerhouse lines.

Altera, a longtime FPGA manufacturer, is poised to successfully enter this market by announcing an embedded-processor strategy that includes both a soft FPGA embedded-processor core product called Nios (pronounced "knee-ohs") and license agreements to use ARM and MIPS embedded-processor cores in future FPGA products. Altera has also announced a forthcoming relationship with Motorola to use a PowerPC core in embedded FPGA products.

In the same time frame, Xilinx and IBM announced a major partnership that licenses the PowerPC processors and associated intellectual property (IP) for use in Xilinx's next-generation FPGAs, which will also target IBM's latest silicon process. In all, these announcements ensure that the performance race is on.

FPGA vendors have wasted no time in realizing they can combine proven processor cores and cutting-edge programmable logic into speedy utility vehicles of choice for successful SOC design. Compared with ASIC SOC implementations, FPGAs have expensive device cost, lower performance, and higher power consumption. However, ASIC SOC development costs, in time and materials, are considerably higher than those for a development process that includes FPGAs. FPGAs' ease of use and flexibility save time and avoid some of the higher ASIC development costs. These properties give the FPGA embedded-processor hybrid chip advantages ASIC SOC chips can't match. The FPGA embedded-processor hybrid chip concept often allows a chip revision to be a simple download of microprocessor software instead of a laborious, time-consuming, and costly ASIC spin.

Technology advances in FPGA devices and corresponding FPGA design tools allow more effective use of FPGAs early in the development process, as well as in later stages of ASIC and product development, when it is imperative to validate and test actual ASIC HDL code beyond ASIC simulation. Combining powerful processor cores with FPGA logic enables the chips of the new millennium—chips that change in "Internet time." The breadth and depth of embedded-processor-related announcements from Altera and Xilinx, together with an announced IBM-Xilinx partnership, signify a strong commitment on all sides, demonstrating that the time for embedded-processor cores in FPGAs has finally arrived.

## The FPGA Embedded-Processor Showroom

Luxury models include large embedded-processor soft cores from ARC, Lexra, Tensilica, and others targeting the larger programmable-logic devices from the Altera APEX or Xilinx Virtex family. Compared with ASICs, these devices range in price from high (hundreds of dollars) to ultraexpensive (exceeding $4,000) for the largest devices. At the same time, these large FPGA devices can consume more than 10W of power, or 200mW/MHz, corresponding to their large number of I/O pins, RAM bits, and flip-flops. With more than 50K flip-flops and more than 800 user I/O pins possible per device, power requirements increase in proportion to the number of pins being driven and flops being clocked. Present embedded-processor soft cores targeting the larger FPGAs usually have clock speeds in the 25MHz to sub-50MHz range, resulting in performance limitations. The initial high-end hybrid embedded-processor hard-core devices could run in the 200MHz range of ASIC-based counterparts and would initially be in the high-price-device category, especially compared with the corresponding ASIC design.

However, the plethora of options in these higher-end FPGAs—options like built-in phase-locked loops (PLLs), numerous I/O pins, and support standards—will make them ideal SOC development devices, even for ASIC designs. Typical I/O standards in these devices are LVDS, LVPECL, AGP, CTT, SSTL, GTL+, HSTL (see I/O sidebar). Multiple output voltages, such as 1.8-, 2.5-, 3.3-, and 5VDC, are desirable options for an SOC.

One may have to look to find economy FPGA devices in a price range suitable for embedded-processor use. These typically smaller and slower devices have fewer features and options, but the $10–$15 volume prices can't be ignored. Devices in this category include Altera's announced ACEX 1K and 2K FPGA families and Xilinx's Spartan and Spartan II devices. Size and performance limitations may, for now, restrict their use to the simpler soft embedded-processor cores like Altera's Nios or to smaller designs like the xr16/32, which targets Xilinx devices *(*see *www.fpgacpu.org/xsoc/xr16.html)*.

Compact embedded processor designs are useful learning tools at the university level and rapidly introduce experienced designers to FPGA technology and the logic synthesis tools. The most attractive aspect of this type of training material is its low cost. The book *Rapid Prototyping of Digital Systems*, by J. Hamblen and M. Furman (Kluwer Academic Publishers, August 1999, second printing January 2000) is $75 with a CD-ROM that contains the Altera MAX+Plus II Student Edition, including all the book's design examples (32-bit RISC design example). The UP1 board used in the text is $149. The Georgia Institute of Technology Web site *(http://users.ece.gatech.edu/~hamblen/book/book.htm)* contains other useful instructional information.

However, the possibility that an embedded hard-core hybrid product will target lower-cost, high-performance, mid-density-range FPGAs is exciting. Devices in this category are significant: they may run a bit slower, but power requirements drop to an interesting 130mW, or 4mW/MHz, range. If the initial FPGA embedded-processor hybrid devices cost $1,000 at introduction, history indicates they can reach the $10 range within two years. Programmable devices that have even lower power consumption can use hand-built micro embedded-processor cores. The hand-built embedded-processor niche is interesting from the standpoint of the number of functions you can cram into a microcontroller inside an ultralow-power programmable device. All devices in the ultralow-power category will trade low density to get ultralow power consumption. This area is often ignored, but it will be getting attention as startups like ARC, Chameleon, Lexra, MorphICs, QuickSilver, Triscend, and Tensilica announce new products having reconfigurable logic.

Embedded-processor cores that specifically target standard FPGAs span a wide performance range in mips and specialized capabilities. These cores can occupy from 500 to 11,000 or more logic elements (LEs), roughly equivalent to 14,000 to 300,000 ASIC gates. They come from ARC, Lexra, Tensilica, and VAutomation—to name only a few companies. This article does not intend to compare features of different embedded cores. Its purpose is to discuss the viability of FPGAs and the role they will play in the

## FPGA I/O Standards

Larger FPGA devices from Altera (APEX) and Xilinx (Virtex) support multiple and programmable I/O buffers with interface for 5-, 3.3-, 2.5-, and 1.8V devices. The user-selectable I/O support includes low-voltage TTL or LVTTL and low-voltage CMOS or LVCMOS standards. These devices can also support the following standards: Stub-Series Terminated Logic SSTL-2 for 2.5V and SSTL-3 for 3.3V; Advanced Graphics Port I/O or AGP, which requires a 1.32V Vref and a 3.3V Vccio; Center-tap terminated CTT, HSTL, GTL+, and LVDS and LVPECL. The tabulation lists some standards and applications (Check with the FPGA vendor for specific details):

| Low-voltage Standard | I/O Support Application |
|---|---|
| LVTTL | General-purpose—low-voltage TTL |
| LVCMOS | General-purpose—low-voltage CMOS |
| LVDS | High-speed backplane driver and datalink—low-voltage differential signaling |
| LVPECL | High-performance clocking, backplanes, optical transceiver, high-speed networking; low-voltage and power ECL |
| PCI | 3.3V 66MHz PCI and PCI-X |
| GTL+ | Backplane driver |
| HSTL | High-speed SRAM interface—high-speed TTL |
| SSTL-2/3 | Synchronous DRAM—stub series terminated logic |
| AGP | Graphics interface—advanced graphics port |
| CTT | Center tapped terminated |

embedded-processor and SOC design community, with the FPGA companies themselves now entering the fray and ready to do battle.

### How FPGAs Affect Embedded Processors

The most compelling aspect of programmable devices using embedded-processor cores goes beyond the typical widespread prototyping use and will surely focus on the flexibility they provide in end-user products and complex systems. Combining configurable high-performance embedded-processor cores and the flexibility of an FPGA is an engineering dream. At first glance, these FPGA-based product and partnership announcements could be dismissed as promoting expensive novelty chips or as a great advancement in ASIC prototyping platforms. However, the high-end capabilities that the tools and the FPGA devices provide will make them indispensable to the SOC design process, driving FPGA-based embedded-processor cores to appear in an increasing number of mainstream products.

### Altera Unveils Embedded-Processor Strategy

A series of announcements from Altera centers on its Excalibur embedded-processor family. The first announcement unveiled Altera's overall strategy. It included a product based on Altera's proprietary soft-core embedded processor called Nios. A major part of Altera's embedded-processor strategy and roadmap is the embedding of transistor-level processor cores in FPGAs; Altera has license agreements to use both MIPS and ARM processor cores. A related announcement, made at the **Embedded Processor Forum**, is that Altera is involved in discussions with Motorola to include a PowerPC core.

Missing from the series of announcements, and perhaps from Altera's primary embedded-processor strategy, is any reference to other embedded-processor soft cores from Altera IP partners like Lexra, Tensilica, and VAutomation. This is surprising, given that Altera is an investor in Tensilica. Including selected IP partners could substantially strengthen Altera's overall processor strategy by leveraging its investment in Tensilica and its association with other IP partners to ease the heavy IP support burden. This support burden is especially heavy when a company is optimizing performance across a wide range of device sizes and package options. One might conclude that Altera is less equipped to support or promote embedded-processor cores beyond the traditional, fixed architecture ARM, MIPS (PowerPC), and Nios embedded processors it has selected.

### IBM and Xilinx Team Up

IBM and Xilinx have announced a partnership to embed PowerPC microprocessor cores and system IP from IBM into Xilinx's next-generation Virtex II FPGAs and have outlined their focus and strategy in this area. These devices will take advantage of IBM's latest 0.13-micron and 0.10-micron CMOS processes. Programmable logic vendors rapidly and aggressively push new process technologies with new devices priced to absorb the lower yields expected with any new silicon fabrication process.

Xilinx or IBM should announce a product resulting from this partnership by year-end. On the IBM and (potentially) Motorola side of these partnerships, announcements could signal a relaxing of the traditionally strict boundaries between ASICs and programmable logic. This attitude change may suggest acceptance of the useful role that true programmable logic fills in the embedded-processor and SOC ASIC development process; it may also suggest a mutual understanding of, and respect for, the benefits of close cooperation.

For Xilinx to say it will license IBM's PowerPC processor cores and related IP indicates that Xilinx may be very serious about making sure that this effort is successful, where it too needs a serious commitment to the necessary "system" resources to ensure success. The challenge of large, costly FPGAs will be in finding a real-world balance between their being inherently less cost-effective than a corresponding ASIC device and their being the best choice in meeting time-to-market requirements (often resulting in a lower development cost). ASICs will always be more cost-effective for higher-volume use, and FPGAs will always be more flexible. There is a new and largely unexplored range of applications beyond the reach of ASICs that is ideally suited for programmable logic. Applications in this area include "dynamic logic" or "configurable chip computing" or "adaptive computing" devices that will come from companies like Chameleon, MorphICs, and QuickSilver.

### Initial Implementation Use

Initially, the high relative cost of large FPGAs will limit their widespread use to the traditional ASIC development process as prototyping platforms; otherwise, they must find specialized and less cost-sensitive applications. A good example of less cost-sensitive applications, or those where benefits outweigh cost, is the common use of FPGAs in network routers and switches. These limitations may initially provide an ideal situation, as supply is likely to lag demand, with a number of users unable to obtain the largest devices. For example, prototype unit pricing for the Altera EP20K1500E (1.5-million-gate or 2.5-million-system-gate FPGA) starts at $4,995, and, even at that price, they are in short supply. Both Xilinx and Altera offer high-volume mask-programmed logic-type devices that allow production versions to be priced at approximately $100 each; however, this type of device is more expensive than a corresponding ASIC device and consumes substantially more power. Power consumption in these masked devices is only about 10% less than in an FPGA version of the design and, together with cost issues, could become a problem in some applications.

The cost issue can be addressed by rapidly developing lower-cost device families, or by targeting embedded-processor designs into devices like Altera's ACEX xK and Xilinx's Spartan II. For example, the Altera EP1K100 ACEX device has 4,992 LEs with 49,152 RAM bits and sells for $11.95 in high volume. The ACEX 1K family is a 2.5V, 0.22-micron/0.18-micron, five-layer-metal device, and the ACEX 2K family is a 1.8V, 0.18-micron, six-layer-metal device with up to 150,000 gates.

Xilinx's Spartan is similar to the ACEX 1K, and the Spartan-II resembles the ACEX 2K, with ACEX FPGAs being 0.18-micron, six-layer-metal devices with a similar gate count and a target price of less than $10 for a 100,000-gate device. Both Xilinx and Altera respond well to customer demands; therefore, after being introduced, these new FPGA embedded hybrid devices shouldn't take too long to migrate to lower-cost devices. In the lower-cost situation, they should find a place beyond clever prototyping or an occasional appearance in mainstream products. On the basis of current and projected costs, one can expect an FPGA with an ARM core and 100,000 gates to be in the $15 range by 2002.

## Nios Processor Architecture

The Nios Processor Architecture from Altera, with its proprietary 16-bit instruction set and push-button (select your processor options and hit the compile button) clock speed of 50MHz, is not by itself a stellar processor architecture; neither, however, is the 16MHz Dragonball used in the Palm series of PDA devices.

What is notable about Nios is what it isn't. It's not just another embedded processor but a complete system—what Altera likes to bill as its complete SOPC (system on a programmable chip). The truly interesting aspect of Nios is its system-control capability, not its datapath manipulation. Other processors running at 200MHz, or faster, are much better suited to number crunching or, in the case of clever Nios usage, datapath manipulation in FPGA-based hardware. Nios is perfectly capable of handling higher-level control functions written in C code. Nios, with its flexible control capability, will be useful in selectively adding or modifying hardware-based IP DSP module functions like Reed-Solomon encoding or decoding. It certainly seems to fulfill the promise of being a simple, fast SOC device.

Altera, through its various partner programs, currently offers its customers the choice of many soft processor cores and other DSP-type modules at its IP MegaStore Web site: *www.altera.com/html/mega/megastore-home.html*.

Altera's entry into the soft-core embedded processor market includes Nios as the first Excalibur product. The Nios core is a significant soft-core development, owing to its astute focus on providing a complete SOC development system. Nios provides the required function and capability to solve many real-world problems at an attractive price. In this regard, Altera is setting the bar high for what its embedded-processor core customers can expect.

The combination of the number of dedicated Altera resources and the company's unusual "system-level" product approach is unique both in cost and in what's being delivered. As a processor, Nios can be trivialized by "Who needs another processor?" As part of a system-on-a-chip development environment, however, Nios has compelling attributes that deliver on the promise of an SOC that is easy to develop and ship. When we dig into the Nios development kit, we find incredible depth and breadth for a "systems"-type product from a chip company.

The clock rate of the initial 1.0 version NIOS processor out-of-box is in the 33MHz to 42MHz range, in a slower –2 speed grade part (the device that comes on the development board). Compiling the Nios design for a faster –1 part provides a 37–46MHz range and should easily reach the promised 50MHz clock speed and corresponding 44 mips by using a design-specific constraint file that isn't in the initial release. Note that a constraint file would typically be completed once a design is 100% functionally correct and locked down. The Nios clock-frequency-to-instruction-execution ratio results in 0.88 mips per megahertz; a 57MHz clock would therefore result in greater than 50 Dhrystone mips. Depending on the amount of effort used to create the constraint file, clock speeds of 66–75MHz should be achievable with current-speed devices, corresponding to a respective 58 and 66 mips.

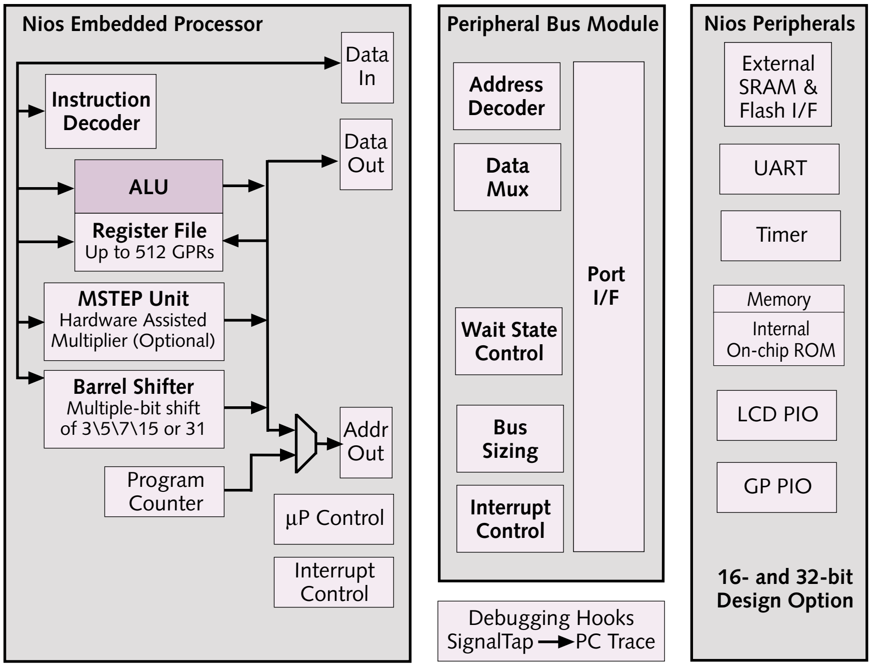

**Figure 1.** Nios embedded processor and reference design peripherals.

A cursory glance at the Nios data sheet reveals a simple 16-bit instruction set, optional 32-bit datapath, and five-stage pipelined RISC architecture running at the 50-mips rate. This information suggests that the performance of Nios, as a processor, may not compare favorably with that of other, newer processors. Closer examination, however, reveals that the designer's use of custom or semicustom FPGA hardware acceleration or DSP type-modules may enable Nios to deliver substantially higher performance than the numbers indicate. The design-tradeoff among performance, architecture, size, and function seems balanced, given that Nios is a soft core targeting APEX programmable-logic devices.

Power consumption might be an issue for some applications, as the example 16-bit Nios system design in the APEX EP20K200E consumes only 130mW, or 4mW/MHz. However, fill the device and drive all the capable I/O pins, and power could rise to more than 800mW, or 16mW/MHz. The Nios system-design measurement of 130mW represented the Nios core and reference-design peripherals running at 33.33MHz.

## Performance

Raw performance numbers or mips ratings will be deceptive, owing to the inherent ability and ease of adding other custom DSP and coprocessing modules to the Nios processor core. Typical Nios applications support a number of DSP-type modules, and we expect this number to increase as specific software functions are implemented in soft hardware modules. A variety of simple and complex software algorithms, made up of both hardware and software, can reside totally within the FPGA. Hardware-accelerated functions may often provide significant performance increases that won't correlate directly to standard mips measurements or other benchmarking functions, especially when Nios is used primarily for higher-level control functions.

Another interesting and desirable feature of Nios is its ability for total device configuration on the fly, where an APEX device can initially come up with a single Nios core and code to start a simple system-configuration process. This code could establish a remote connection and download a new "device image" into an inexpensive flash device, where the embedded Nios processor could reconfigure itself by reloading the APEX device on command. A new image is reloaded into the APEX device, allowing it to come up configured with multiple Nios modules or to load an entirely different function, perhaps a semicustom system using the embedded ARM or MIPS processor.

Nios's slower "out-of-box" clock speed of 33–42MHz results from the fact that the compile is done without any Nios-specific constraints or speed optimization. Device constraints can be a big issue, especially for complex designs, together with limitations imposed by the Quartus tools. Location-attribute assignments within Quartus are made either coarse-grained or overly cumbersome by requiring that large portions of logic receive hand-placed constraint assignments.

Quartus doesn't support a "relative location" constraint, where single "X" location is defined, and all predefined locations key off this relative point. Such a feature would be particularly useful in multi-Nios-processor support in future releases. However, the constraint-file problem should be greatly diminished in subsequent releases of Nios and not tied to the Quartus constraint-placement capability. Nios support files include a large number of Perl scripts in source-file format. The Perl scripts optimize the core by creating various other source files that the development tools use, and this method is likely to address constraint-file issues.

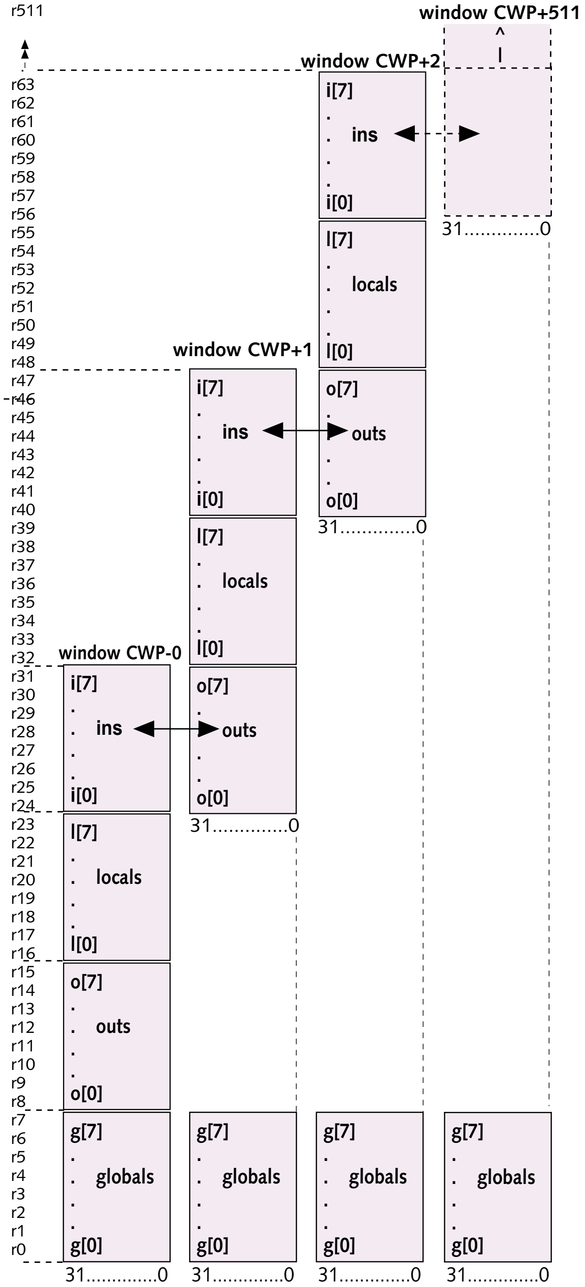

**Figure 2.** Nios register file window.

## Nios Architecture Details

The Altera default example design demonstrates the logical design tradeoffs among function, fit, and performance. First, Nios is SPARC-like in its 32-bit instruction set and 256 general-purpose registers with option to double to 512. Concerned about core size and the amount of logic required to implement a core of that size, Altera created a 16-bit version of Nios. All registers appear in a movable, overlapping 32-register window controlled by the current window pointer. This arrangement provides access to a total of 32 overlapped registers, with 8 as "in" registers and 8 as "out" registers to allow contents to be rapidly switched between the two. There are 8 additional "local" and "global" registers, as shown in Figure 2. The window increments up or down by 16 registers, and an exception is generated for an overflow or underflow of the total register space. This register window does not wrap around, so as not to impinge on existing SPARC patents. The Nios design team is located in Altera's Santa Cruz development center.

Recommended use of these registers is to save the whole register file on an underflow or overflow exception, starting over at the top or bottom, depending on whether the event was an underflow or overflow. The software included with Nios performs this recommended use by default. The windowed register feature is useful for preserving register states without incurring performance penalties from pushing or pulling the memory stack; it is especially handy for tight instruction loops like those found in low-latency interrupt routines.

The hardware-assisted multiply, or MSTEP, unit uses dedicated hardware, together with an optimized math library, for multiply operations. Using this feature results in a 1-bit-per-clock multiply that produces about a sixfold improvement over software-based multiplying routines. A 16 x 16 multiply takes 16 clock cycles and is programmed by using 16 consecutive MSTEP instructions. This feature can be disabled during design compilation, reducing core size by a small amount.

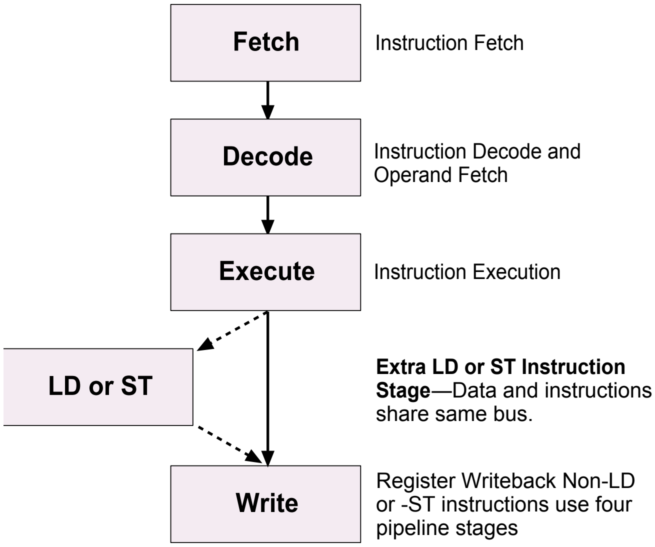
Fetch
Instruction Fetch
Decode
Instruction Decode and Operand Fetch
Execute
Instruction Execution
LD or ST
**Extra LD or ST Instruction Stage**—Data and instructions share same bus.
Write
Register Writeback Non-LD or -ST instructions use four pipeline stages

**Figure 3.** Altera Nios pipeline stages

Another compile option is the addition of a barrel-shift function. The default compile option is a 7-bit shift-right or shift-left command. Calling this command will shift the operand value right or left by 7 to 1 bit(s) in a single clock cycle. Operand shifts that exceed 7 bits require multiple shifts per clock, with increments of 7. Design-compilation options include shifts of 1, 3, 7, 15, or 31 bits per clock, whereas select value defines the number of shifts per clock period and can be expanded to 15 or 31 bits per clock.

Nios has a four-stage-pipeline with a "fifth stage" added for load (LD) and store (ST) operations. This selective pipeline stage is required because the Nios CPU core shares the same bus for data and instructions. The five-stage processor pipeline stages are shown in Figure 3.

## Nios Development Kit—SOC Development in a Box

The most admirable aspect of Nios is the ease of system design and the speed with which a user can start software development on actual hardware. Getting the bargain-priced Excalibur Development Kit and attending a half-day workshop could have a designer up and running in a single day. Of course part of this low-cost attribute is that the user gets all the needed software-development tools, including the Altera Quartus development software and the GNUPro compiler and debugger from Cygnus. What isn't apparent in the Nios kit is the extended hardware-debugging capability offered by the embedded logic analyzer (SignalTap) that is part of the Quartus development software. Of particular interest to someone debugging code is the ability to "scope" register contents, PC trace, or other internal operations when debugging code.

The Quartus software license in this package ends up being restricted after 120 days to the smaller APEX EP20-K200E device on the SOPC development board. This restriction shouldn't normally be an issue, but it is something to be aware of for longer-term development, as the yearly license fee is $1,995. The initial 1.0 demo-kit release does not include RTOS support, but we expect this situation to change rapidly and add yet another interesting wrinkle to standard-setting SOC development systems.

The only tool that seems to be missing is a third-party simulation tool. The normal tool that is typically bundled with Quartus is ModelSim from Model Tech, but the license for this product is available with the full version of Quartus only and is not included in the Nios development kit. Judging from the availability of the Excalibur development kit and the Excalibur workshops, it's obvious Altera is serious about applying the necessary resources to delivering system-development tools that contain all essential components in a reasonably priced $995 kit. A big test for all will be the level of support designers will get for this price.

The default example design is important in that it provides baseline size and performance starting points.

Hardware-configurable features of the default example include 256 registers out of a possible 512, a 7-bit barrel shifter that can be adjusted up or down, and a hardware-assisted multiply (MSTEP) unit that can be disabled. The default design includes a number of peripheral devices tied to the processor via a peripheral bus module. Peripherals include a 115.2-kb/s UART; a 32-bit interval timer; several parallel input/output (PIO) ports for controlling the demo board's buttons and switches; an extra PIO for user expansion (or for driving the included LCD display); an on-chip ROM; an interface to 256KB external SRAM; and an interface to 1MB external flash memory.

The two common Nios cores use a variable datapath, with the 16-bit version using 1,100 logic elements (LEs) of an APEX device or an unofficially supported ACEX device, and the corresponding 32-bit version using 1,700 LEs. These numbers equate to about 25,000 ASIC gates that consume 12% of an APEX EP20K200E and 2% of an APEX EP20-K1500E. The ACEX 1K and 2K families aren't yet fully supported, and one can expect a 50% clock-speed reduction with push-button compile of these devices. Adding in the standard default peripherals, as stated above, increases the number of LEs to 1,637 LEs for 16-bit devices and 2,375 LEs for 32-bit devices.

These baseline examples are not the normal stripped-down versions that deflate core size or device resource requirements. If need be, the Nios core can easily have functions stripped to basics that will provide a 16-bit core that consumes about 150 fewer LEs. This stripped-down version would still have vectored exception handling, all addressing modes, and the windowed register file. Some additional features could be removed to free a couple of hundred extra LEs if core size becomes that important.

Programmable-logic clock performance can be optimized by applying a little more knowledge and varying degrees of effort in creating a constraint file that includes pin assignments, special devices features, timing parameters, and location attributes. Knowing how to optimize all aspects of the design (maybe by rewriting some HDL code), and expending significant effort to do so, could allow a design to hit 100MHz with current APEX silicon and 150MHz (132 mips) in future silicon. Applying this much effort doesn't become a "push-button" compile and would be the reason Altera isn't committing to anything over 50 mips at this time.

Higher clock speeds and resulting performance increases can be achieved by applying extra effort in creating constraint information or optimizing HDL code to decrease connection path delays. Customers should not expect highly tuned constraint files, as they are extremely application- and design-dependent. The good news about Nios is that Altera seems to have received and fully embraced the "it's the system" message: Altera has defined a dedicated hardware-, software-, and tool-support staff for Excalibur, clearly demonstrating that the company is "trying to do Nios right the first time."

### The SOC Development Process

The heavy use of integrated Nios MegaWizard Plug-Ins (Perl-script driven) in the tool process is what makes the SOC development so easy. The 0.9 release included a Nios processor plug-in, a peripheral plug-in, and a peripheral bus module, which all auto-magically generate all the required hardware design files to include Verilog files, peripheral-translation files, and system-description files. The plug-ins can also generate system-software test-bench files. The interesting aspect of this process is that it allows dynamic bus sizing of all peripherals, whereas bus width ends up being transparent to software. The 1.0 release version is said to have a higher level of integration with the MegaWizard Plug-In and will maintain the flexible development process and the close association with the Quartus development tool.

### Future Excalibur Processor Products

The Nios embedded processor by itself is not architecturally significant. Add it together with hard or fixed microprocessor cores in the Excalibur family, however, and it begins to tell a more compelling story of things to come and of Altera's planned direction. Altera has at this time announced only license agreements involving ARM and MIPS embedded-processor cores; expect product announcements based on these cores combined into large FPGAs as embedded-processor hybrid-type chips using ARM's ARM9T and MIPS Technologies' MIPS32 4K processor cores.

How soon Altera will announce a product based on the Motorola PowerPC processor cores and whether the agreement includes access to advanced fabrication processes are open questions at this time. Motorola is not yet offering packaged PowerPC cores and related IP products. This situation is potentially problematic and might delay new products based on PowerPC unless significant resources and commitment are available to fully support the Motorola PowerPC cores as an intellectual-property-based product. Altera's wanting to use AMBA as its standard bus interface and a requirement for an AMBA-to-PowerPC interface is one area that potentially calls for significant effort: the cores aren't limited to older processes and slower-speed devices but include state-of-the-art 32-bit devices with a clear upgrade path to 64-bit bus widths.

Designs using embedded hard cores, like the ARM9T and MIPS32 4K in the large next-generation APEX devices, will create interesting challenges for Altera: it will have to develop chips that can compete in the SOC chip market pricewise and continue to erode the ASIC chip-development advantage. Custom packaging situations create conditions in which embedded-processor cores end up being paradoxically more widespread in their use but potentially less visible than traditional processor applications. Being able to selectively add highly customizable and proprietary enhancements at will allows tremendous product variation by using a simple download to an inexpensive flash memory device.

### Price & Availability

The $995 Excalibur Development Kit featuring the Nios embedded processor is available from Altera and can be ordered online from the Quick Link: www.altera.com/html/products/nios.html.

ES Version is shipping now with Version 1.0 Production Orders shipping the end of September 2000.

## Delivering Real-World Solutions

Combining random bits of hardware and microcode also increases design security. Gone are the days when one could examine a board, find some sort of processor, predict performance, and usually figure out (or get a good idea of) how the design is put together and what the primary functions of the board or product are. At the same time, applications that include an embedded hard core in a hybrid device are likely to include a number of simpler soft cores, like Nios, for specific dedicated functions, such as system configuration and management. Another possible use would be to reconfigure soft cores expressly for a particular speed enhancement: for example, "ripping a CD" with specific hardware acceleration features.

Simple is good. Look at the success of Palm. Proof of success isn't the attention or press coverage a technology gets but the number of devices sold. Success in this case directly relates to how widely the technology is adopted, and that correlates with the number of design wins and with the higher end-user acceptance that greatly increases sales.

Both Altera and Xilinx are keenly focused on these attributes of the success formula. Being rivals competing for the top spot creates other opportunities. Traditional ASIC designers know and understand programmable logic and are using it in ever-increasing quantities. Success in this case means careful attention to the essential requirements of the task at hand: the problem the designer is trying to solve. ASIC and SOC designers know they are part of a team designing a system, with system attributes being the difference between life and death, success and failure.

An accelerated design process must focus on simple system attributes and on the system's ability to solve real-world problems. The system process and time-to-market requirements give flexibility and adaptability a higher priority than price. How many times has a user said, "I'd pay more if I got what I really wanted"? The device best able to meet these needs will be one that combines a powerful, flexible processor and user-defined logic. Licensing and partnership announcements from Altera, ARC, ARM, IBM, MIPS, Quicklogic, and Xilinx give clear indications that products based on "hybrid" devices that combine powerful processor cores and leading-edge programmable logic will be very important chips in the "want to have it now" Internet age.

Use of chips with high-speed embedded processors and large numbers of programmable logic cores won't stop at the prototype stage. These same devices will leap into the mainstream by being integral to a new-product launch, allowing early market penetration with "soft-hardware" that won't be obsolete after just a few months. The ease with which designers can work with these systems on programmable chips (SOPCs) and the satisfaction of customers who receive obsolescence-proof early products ensure rapid adoption of these new devices.

It won't take long for sharp marketers to understand the overall attractiveness of this concept and to create ways to amortize the negative issue of high cost. Cell-phone resellers make money selling time, not phones. Game-console manufacturers make money on game sales, or software, not on gaming hardware. Early product release would cover the high cost of FPGA embedded-processor hybrid devices, estimated to be $1,000 in starting volumes, where the money is made not on the device but with the service. The formula is relatively simple, following the loss-leader philosophy of cell-phone or game-station vendors that amortize initial high costs with follow-on lower-cost products. It seems like a logical and simple way to be first to market with a product that's obsolescence proof.

There are interesting ways to take care of the issues of using expensive FPGAs by focusing on the benefits. The benefits are a product that can be updated or changed via an Internet connection—benefits that not only present designers with new weapons in the time-to-market war but that promise the tremendous marketing advantage of using custom features to appeal to a larger market: Have it your way.

All of us would be more willing to pay more for technology that can be fixed and upgraded with a simple download, delivering what we want, when we want it. ◇