

designfeature *By Gabe Moretti, Technical Editor*

# EDA tools for FPGAs break down the complexity gridlock

**FPGA DEVICES OFFER EXECUTION SPEED AND GATE CAPACITY THAT RIVALS MANY ASIC IMPLEMENTATIONS AND FOSTERS THE GROWTH OF EDA TOOLS IN THAT MARKET.**

*At a glance*.....28  
*Embedded software and FPGAs: a partnership ready for prime time*.....28  
*For more information*.....32

**A**DVANCES IN SEMICONDUCTOR MANUFACTURING are the primary reasons for the increased popularity of FPGAs. As long as engineers could use only a PLD or an FPGA to implement relatively

simple glue-logic circuits, use of ASIC devices that provided greater flexibility and lower volume costs offered the only solution for implementing specific functions on an IC. FPGA vendors are now manufacturing devices at both 130- and 90-nm, half-pitch dimensions. These devices not only enable designers to implement circuits that require more than a million logic gates, but also offer a rich inventory of IP (intellectual-property) cores that decrease development time and cost. Concurrently, the NRE (non-recurring-engineering) cost of ASIC devices manufactured with the same 130- and 90-nm technology has risen dramatically. A mistake that requires a new set of masks can easily cost \$250,000 to \$1 million, depending on the severity of the error.

Managers are therefore often opting to

use FPGA devices, either for the entire life of a product, if applications require only a few tens of thousands of devices, or for prototyping and volume ramp-up. Once volume production shows that a design is stable, engineers can port the design to an ASIC device. The porting is generally easy, because in most applications, engineers do not take advantage of the field programmability of the device, once the major characteristic that differentiated these devices from PLDs. FPGA devices are attractive, because the cost of modifying an implementation is practically equal to engineering-development cost. Thus, they allow engineers to use attractive debugging methods, especially in embedded systems, where the integration of hardware and software requires a hardware prototype as early in the development cycle as possible. The absence

of standard microprocessor cores in FPGA fabrics had been a serious impediment to engineers' using these devices for embedded-software applications. But, beginning with the 130-nm process node, both Altera and Xilinx have provided microprocessor cores, and ARM has also made available many of its standard microprocessor cores in libraries tailored for FPGAs. Designers also need a library of coprocessors and peripheral-IP cores to implement true SOC (system-on-chip) products using FPGA devices (see sidebar "Embedded software and FPGAs: A partnership ready for prime time").

The density and speed of today's offerings from FPGA vendors has convinced IP vendors to port their products to FPGAs. DSP cores and graphics-display cores are becoming available. Of course, cores and proprietary logic blocks must communicate with each other within the device, and engineers until re-

#### AT A GLANCE

- ▶ FPGAs provide the capabilities that system-on-chip designs require.
- ▶ Most FPGA vendors provide their own development-support software.
- ▶ Engineers that require the flexibility to target devices from multiple vendors need software tools from traditional EDA vendors.

cently implemented this task by using buses. All three popular microprocessor cores—Nios and ARM922T (through a licensing agreement with ARM) from Altera and the PowerPC from Xilinx (through a licensing agreement with IBM)—use standard buses. But the speed required to transfer data and control information among the cores has inspired Nallatech to develop a faster communication approach for FPGAs.

System communication can consume as much as 80% of application-development time, according to Craig Sander-son, a system-application engineer at Nallatech. The Dimetalk communications-development tool for FPGA-computing applications enables developers to deploy packet-based networks that can span systems using multiple FPGAs. Designers can deploy interface nodes at any point within the network as well as insert blocks for communication with external interfaces.

The most significant remaining barrier to the widespread use of FPGAs for system design is their unit cost. Even in large volumes, the cost of one FPGA device is higher than the cost of the same design implemented in ASIC or structured-ASIC technology. But increasingly, the unit cost of a part during manufacturing is losing significance in the overall product-cost equation, as development costs and costs related to lost

## EMBEDDED SOFTWARE AND FPGAs: A PARTNERSHIP READY FOR PRIME TIME

By David Stewart, CriticalBlue

FPGAs, given their ease of programming, should be a natural hardware platform for embedded-software developers. However, FPGAs have remained largely in the domain of hardware engineering.

Currently, embedded-software developers count on the ability to rapidly reprogram the systems they develop. The need to deal with product bugs and react to quickly changing market requirements has made reprogrammable microprocessors the natural hardware platform for embedded-software engineers. Highly reprogrammable, microprocessors also work within mature tool chains that enable embedded-software engineers to automatically and efficiently map source code onto the chosen microprocessor architecture.

This model works well until the microprocessor can no longer deliver the data-processing performance that the application requires. At this point, it

generally becomes necessary to develop a hardware "assistant," which uses parallel computing resources to speed the key application bottlenecks, to sit next to the microprocessor. An FPGA fabric, offering a hardware-reprogramming capability and even the possibility of an on-chip microprocessor should be the perfect platform for an embedded-software developer to target in these circumstances.

Unfortunately, the analogy between microprocessors and FPGAs breaks down here. Whereas a painless path exists from embedded software to a microprocessor, an easy route to an FPGA emerged only recently. Multiple options can get the designer from RTL (register-transfer level) onto an FPGA fabric, but the route from embedded software to RTL involves a number of resource-intensive, manual activities. The functions that you need to accelerate you must recode as a hardware

model in C, SystemC, or RTL, and you need to define and validate the communication interface between the hardware accelerator and the main microprocessor so that you avoid hardware- and software-integration problems in the late stages of the project. The complexity of this approach may be intimidating enough to prevent designers from using FPGAs in applications in which they would be natural hardware assistants to a main microprocessor.

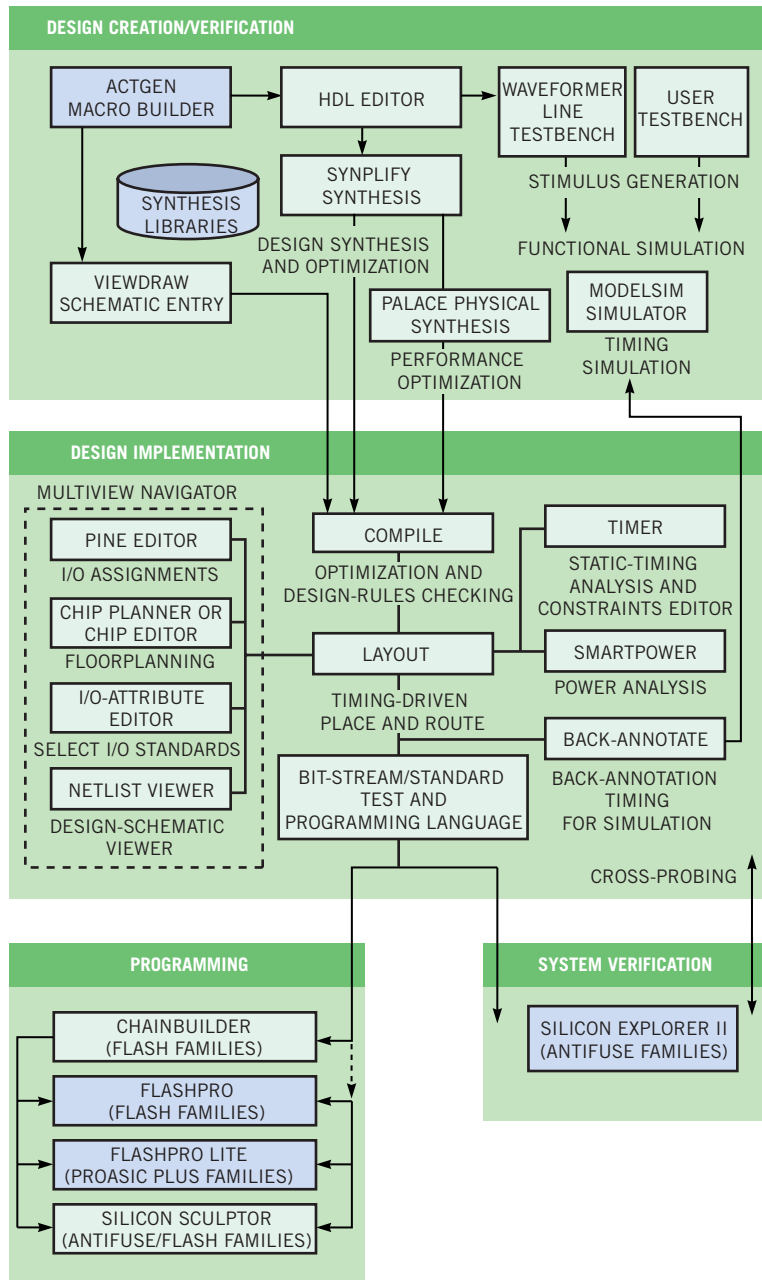
To deliver a smooth path from embedded software to RTL, a design must meet two clear requirements. First, the underlying architecture of the hardware assistant must be a true co-processor—in other words, a programmable architecture under control of the main processor. This requirement ensures that the developer can migrate the full richness of embedded-software languages onto the co-processor, if needed, and need

not adopt any new coding style in his chosen programming language. Second, it is essential that the output of the embedded-software-development environment—in other words, the executable code—drives the migration path toward RTL. This requirement ensures the preservation of a system company's investment in the developer's desktop environment and allows automatic handling of the interfacing between the main processor and the co-processor.

Now that an approach exists that meets the above requirements, embedded-software developers may finally be able to embrace FPGAs as a natural hardware platform for their products.

#### AUTHOR'S BIOGRAPHY

David Stewart is chief executive officer of CriticalBlue (San Jose, CA) and has more than 20 years' experience in the EDA and semiconductor industries.



**Figure 1** The complexities of FPGA-design methods are similar to those of ASIC design.

market opportunities are escalating by at least one order of magnitude with every new available process technology. At the same time, the difference in device-unit costs between ASICs and FPGAs is becoming smaller.

**VENDOR-SUPPLIED TOOLS**

Since June, Actel, Altera, Lattice, and Xilinx have all introduced new versions of their software. All FPGA vendors bun-

dle third-party software in their offerings. The EDA vendors provide the FPGA vendors with tailored versions of their products that generally do not offer all of the capabilities and power of the original tools, because FPGA vendors offer their tools at much lower prices.

Actel's philosophy is to focus its resources on the back-end flow in place and route, which requires a deep, clear understanding of the architecture. Actel

was a leader in 1994 in introducing static-timing analysis for FPGAs to help designers achieve timing closure before committing designs to silicon. The company works with traditional EDA vendors to supply front-end tools, such as Mentor Graphics' ModelSim logic simulator, Synplicity's logic-synthesis Synplify, and Magma's Palace for physical synthesis. It integrates the tools in the Libero IDE (integrated design environment), which is available in three configurations, depending on customers' needs, that sell for \$595 to \$2595. The design flow for FPGA development has become sophisticated (Figure 1). For designers who prefer to use their own EDA tools, Actel's Designer includes the place-and-route tool as well as the static-timing-analysis product.

Altera provides Quartus II software to its customers in both a CD and a Web-based product. The Web product provides users with a license that expires after 150 days and does not support all of the Altera devices. Altera has since its inception invested a significant amount of money in developing and supporting its own design tools. It internally develops almost all of the tools in an FPGA-development flow, including logic- and physical-synthesis tools. It believes that when an FPGA vendor develops both the architecture and the synthesis tool for a new product, it obtains better insight about the best way to optimize the device structure than if it had no knowledge of synthesis technology. Logic simulation is the only area in which Altera exclusively

uses third-party products. Quartus II provides an Altera version of Mentor Graphics' ModelSim but also supports Cadence's Incisive simulation platform. You can also use tools from both Synplicity and Synopsys in conjunction with the tools available in Quartus II. The latest version of Quartus II introduces timing- and resource-optimization features to guide users during their design cycles. Engineers also receive help during debugging from SignalTap II, an embedded logic-analyzer viewer. The Quartus II software is available for an annual subscription license starting at \$2000.

Lattice Semiconductor has released Version 4.1 of the ispLever design-tool suite. It includes all the tools needed to move a programmable-logic design from concept through implementation. It includes tools for design entry, project management, design fitting, place and route, floorplanning, device programming, and on-chip logic analysis. Engineers can choose between synthesis products from Synplicity or Mentor and also receive a version of ModelSim tailored for Lattice-based designs. Lattice also provides its own functional simulator. List prices for ispLever begin at \$995.

Xilinx is another FPGA vendor that has invested a considerable amount of money in EDA-tool development; designers can choose from five configurations of its ISE design tools. The company has recently updated its ISE development environment with version 6.3i, which supports a complete front-to-back flow for FPGA users. On the front

end, ISE includes the Pace and ISE Floorplanner floorplanning tools. Xilinx also acquired Hier Design and now offers the RTL floorplanner Plan Ahead as a separate purchasable option for ISE customers. Engineers can use either ModelSim or Synopsys Verilog simulator with ISE. Project Navigator, the main task manager for ISE, lets users configure and drive the design implementation. ISE provides a synthesis tool that Xilinx develops internally, but Synplicity, Mentor, and Synopsys also provide synthesis tools that engineers can use in conjunction with ISE. The five configurations of ISE range in price from a free Web-downloadable package to ISE Foundation for \$2495.

### THIRD-PARTY TOOLS

For many years, programmable devices were so simple that EDA vendors found that they could not charge enough for the tools to justify entering the market. Most engineers using FPGA and PLD devices were pc-board developers who lacked access to EDA tools targeting IC design. FPGA vendors developed their own schematic-based tools, because, even today, pc-board designers use schematics to enter designs into the development flow. Vendors either gave free FPGA tools to volume customers or sold them for less than \$1000. As devices grew in complexity, engineers began to want to verify designs before implementing them on breadboards, because debugging a circuit using an oscilloscope or a logic analyzer became time-consuming. Model Technology, now part of Mentor Graphics, was quick to enter the market and now has a commanding position in FPGA verification, although a number of other EDA vendors also address this market.

Today, engineers also need robust synthesis tools to develop FPGA designs, and most EDA vendors that support ASIC synthesis also provide tools for FPGAs. Other vendors, such as Altium and Aldec, whose primary market is pc-board development, have also recognized that designers must take a system approach to product development. An engineer can no longer consider an FPGA as a separate system with no impact on the rest of the board. EDA companies are providing flows that seamlessly integrate with the

## FOR MORE INFORMATION...

For more information on products such as those discussed in this article, contact any of the following manufacturers directly, and please let them know you read about their products in *EDN*.

#### Accelchip

[www.accelchip.com](http://www.accelchip.com)

#### Actel

[www.actel.com](http://www.actel.com)

#### Aldec

[www.aldec.com](http://www.aldec.com)

#### Altera

[www.altera.com](http://www.altera.com)

#### Altium

[www.altium.com](http://www.altium.com)

#### Bluespec

[www.bluespec.com](http://www.bluespec.com)

#### Cadence Design Automation

[www.cadence.com](http://www.cadence.com)

#### CriticalBlue

[www.criticalblue.com](http://www.criticalblue.com)

#### Lattice Semiconductor

[www.latticesemi.com](http://www.latticesemi.com)

#### Magma Design Automation

[www.magma-da.com](http://www.magma-da.com)

#### Mentor Graphics

[www.mentor.com](http://www.mentor.com)

#### Nallatech

[www.nallatech.com](http://www.nallatech.com)

#### Synopsys

[www.synopsys.com](http://www.synopsys.com)

#### Synplicity

[www.synplicity.com](http://www.synplicity.com)

#### The MathWorks

[www.mathworks.com](http://www.mathworks.com)

#### Xilinx

[www.xilinx.com](http://www.xilinx.com)

back-end, vendor-specific, FPGA tools. Although these tools generally cost more than tools from FPGA vendors, they offer more functions and allow designers to work independently of FPGA vendors. In addition to products that ease the integration of FPGA devices and pc boards, traditional EDA vendors have focused on verification and synthesis as the two market segments that best suit third-party support.

Design verification figures as the largest cost of developing an IC, and it is becoming a significant expense in designs for FPGAs, as the complexity and size of the design grows with the sophistication of available devices. As more engineers trained in ASIC development move to FPGA devices, they tend to use the hardware-description language they are familiar with. Verilog, once almost nonexistent in the FPGA market, is gaining market share. Bluespec and Mentor Graphics offer support for SystemVerilog, the latest version of Verilog. ModelSim from Mentor Graphics is still the most popular logic simulator in the FPGA market. It supports all hardware-description languages now available to designers. Cadence's Incisive simulation platform and Synopsys' VCS are present mostly in design flows that have incorporated FPGA design into traditional ASIC design.

Engineers are also beginning to use design languages that support higher levels of abstraction, such as C and its hardware-oriented dialects, as well as the MathWorks' Matlab. Mentor has introduced Catapult C, a product that allows designers to synthesize designs described in untimed C, and Catalytic offers a product that allows designers using Matlab to transform a floating-point-DSP algorithm into a fixed-point one. Engineers using any of the DSP cores available for FPGA devices need fixed-point algorithms, and Catalytic makes it possible for them to verify that both implementations are equivalent within the Matlab environment. In-circuit debugging of FPGAs presents several challenges. For example, clock speeds can exceed 200 MHz, and the design can have multiple circuits running at dif-

ferent clock speeds. Agilent Technologies has introduced the 16900 series of logic analyzers to address issues such as connection to the device under test, dynamic probing, and viewing and analyzing system behavior in various formats.

### SYNTHESIS

As device complexity grows, designers need more sophisticated synthesis tools. Although you can still develop a functionally correct FPGA using only logic synthesis, the most sophisticated designs require engineers to also use physical synthesis to meet timing requirements. Physical synthesis differs from logic synthesis because it takes the eventual chip layout into consideration while performing circuit optimization. Physical synthesis can handle more complex timing requirements by experimenting with various chip-layout approaches to find the topology that best meets the requirements. Both Altera and Xilinx offer their own logic-synthesis products in addition to third-party tools; Actel and Lattice prefer to offer only third-party tools.

EDA vendors Mentor and Synplicity dominate the market for FPGA logic and physical synthesis. Until recently, Synplicity focused solely on the FPGA market; only in the last couple of years has it widened its interest to embrace both structured- and traditional-ASIC devices. It offers Synplify, a logic-synthesis tool; Synplify Pro, a more powerful version of Synplify; and Amplify, which adds physical-synthesis capabilities to Synplify Pro. Mentor Graphics has had a presence in the FPGA-development-tools market for many years but was late in providing a competitive synthesis product. With its Precision synthesis tool, the company has regained some of the ground it lost to Synplicity.

The Dataquest *2003 Market Trends* report puts Synplicity shares of the FPGA-synthesis market at 44% and Mentor shares at 43%. Synopsys, the undisputed leader for logic and physical synthesis in the ASIC market is a distant third in the FPGA world. As long as the capabilities of ASIC and FPGA devices differed significantly, vendors assigned different teams of designers to developments in-

volving the two types of devices. Although Synopsys tried twice before to enter the FPGA market, it was unsuccessful and eventually abandoned the effort. But as FPGA vendors began to use both 130- and 90-nm processes, the performance of FPGAs justified their use for both ASIC prototyping and as ASIC replacements in early production runs. Therefore, it became common for FPGA designers to also work on ASIC design. Both Synplicity and Mentor have extended their FPGA tools to address some of the ASIC market. Thus, it made sense for Synopsys to once again provide an FPGA-synthesis tool. DC FPGA shares the front end with the popular and successful Design Compiler synthesis product and targets ASIC designers involved in FPGA development. Magma has also entered the FPGA physical-synthesis market with Palace, which is part of the suite of third-party tools offered by Actel. Bluespec has introduced a synthesis product that supports SystemVerilog, the latest proposed standard in the Verilog market.

FPGAs can be effective for certain DSP functions, especially algorithms that take advantage of parallel operations. Many designers of DSPs are unfamiliar with EDA tools. Instead, they begin the algorithmic development using The MathWorks' Matlab and Simulink and then translate their design into a hardware implementation that uses a DSP and some embedded software. Two years ago, Accelechip introduced a product that allows engineers to use Matlab and Simulink to develop and verify the algorithms and then implement the circuitry in an FPGA without manually reimplementing the design in VHDL or Verilog. Recently, Synplicity introduced Synplify DSP, which also allows designers to begin the algorithmic development of a DSP using Matlab and Simulink and then generate the RTL code that they can enter into the synthesis tools. Altera customers can use DSP Builder, which the company developed internally, to link their Matlab and Simulink designs to the Quartus II environment. □



TALK TO US

Post comments via TalkBack at the online version of this article at [www.edn.com](http://www.edn.com).