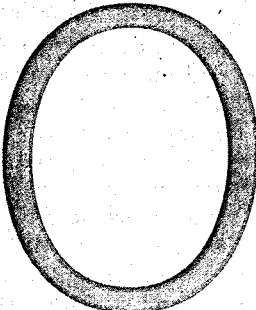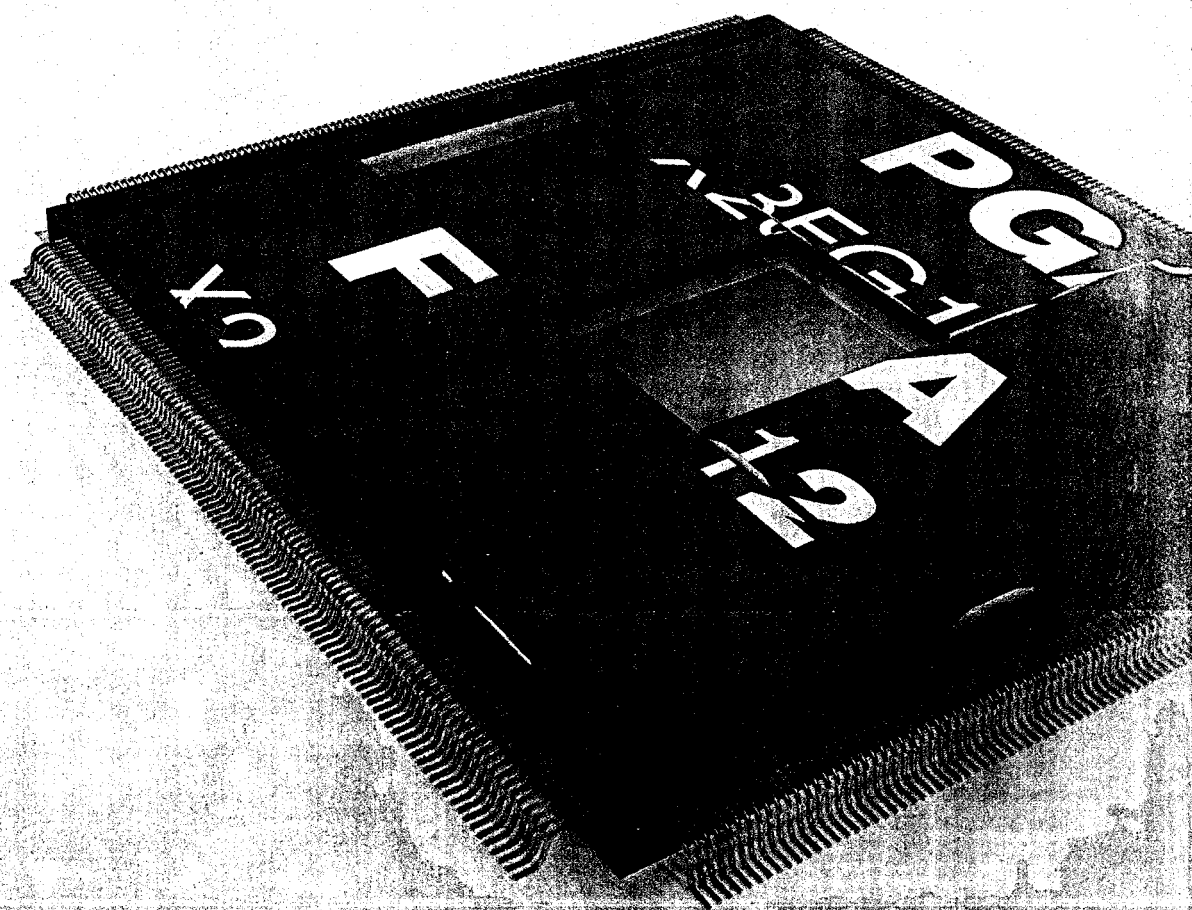BY MICHAEL SANTARINI • SENIOR EDITOR

Is FPGA a simpler puzzle for ASIC designers?

WITH RISING MASK COSTS, COMPLEXITY, AND TOOL EXPENSES TO DEVELOP ASICs AND SOCs, MANY DESIGN GROUPS TODAY ARE OPTING TO IMPLEMENT THEIR PRODUCTION DESIGNS IN FPGAs. BUT, BEFORE DESIGNERS MAKE THE LEAP, THERE ARE SEVERAL FACTORS—GOOD AND BAD—THEY SHOULD CONSIDER.

Over the last 10 years, FPGA vendors have made great strides in overcoming the shortcomings of FPGAs and taking share from the ASIC market. In the late 1990s, FPGA vendors increased the capacity of their devices to rival midsized ASICs. Then, circa 2001, FPGA vendors improved the performance of their devices to compete with midsized ASICs. Though FPGAs still consume much more power than ASICs of comparable densities and performance, last year, FPGA vendors made great strides to stabilize the amount of power that FPGAs consume (**Reference 1**).

On top of these device-attribute gains, FPGAs have also come down in price. Vendors such as Actel, Altera, Lattice, Quicklogic, and Xilinx offer a wide swath of devices ranging from pennies-per-part CPLDs (complex programmable-logic devices); to secure, nonvolatile FPGAs; to high-performance, high-LUT (look-up-table)-count, SRAM-based juggernaut FPGAs that cost thousands of dollars each.

In the early days of the FPGA industry, designers would use the most expensive and highest grade FPGAs mainly to functionally prototype code that they planned to implement in an ASIC or to do proofs of concept for system designs. They would create the logic for their ASICs, run verification, partition their ASIC designs, and then program those partitions onto several FPGAs residing on a prototyping board (**Reference 2**). Designers today still use this method, but, because FPGAs have improved on all fronts, many designers are using even the highest grade and most expensive FPGAs for production parts.

It's fairly easy to get a marketing executive at an FPGA vendor to tout the glory of FPGAs and how they are taking over ASIC real estate—even for production use. Designers are also stepping forward to say that FPGAs are truly viable production vehicles and that designers should no longer dismiss them simply as ASIC-prototyping tools.

Sanjay Singh, the technical lead in ASIC/FPGA design at Hewlett-Packard's nonstop-computer division, has over the course of his career designed 10 FPGAs and 25 to 30 ASICs. He started designing ASICs with Tandem Computer, which Compaq acquired in 1996; HP in turn in 2002 acquired Compaq. "I started at 0.5 microns doing ASICs for Toshiba nonstop computers," says Singh. Currently, his group is designing ASICs at 110- and 90-nm nodes, and, when the situation calls for it, his group uses the highest grade SRAM-based FPGAs—Altera Stratix- and Xilinx Virtex-class devices—for server applications.

"Our systems are based on Intel Itanium-server chips, and the FPGAs we design have to talk to memories, I/Os, and processors," says Singh. "Our value added is in hardware, so we have to do communications, porting functions, data-integrity functions, and duplicated functions. The FPGAs we design typically go on a

processor board, and the end systems will run upward of a million dollars."

On the other end of the design spectrum, Ranjit Rozario is a senior design engineer at communications start-up Sonoa Systems. Rozario is one of a few hardware designers in a 100-person company that comprises mostly software engineers. A long-time ASIC designer, Rozario recently took his first foray into FPGA design, ultimately choosing a Virtex-5 LX 220.

Both Singh and Rozario say they are more frequently going with FPGAs for various reasons but note there are advantages and disadvantages of designing with FPGAs. So, when making the move, ASIC designers must take into account several factors, including design size, performance and power budgets, PCB (printed-circuit-board) requirements, design and verification requirements, and FPGA-tool limitations. SRAM FPGAs also introduce new challenges, such as soft errors, which are more commonplace in SRAM fabric than in standard-cell material.

WHY MOVE TO FPGA?

There are several reasons that designers choose FPGAs over ASICs: FPGAs are reprogrammable and field-upgradable, and the design cycles are shorter than ASIC-design cycles (**Figure 1**); FPGAs have better prices for high-cost, low-volume applications; and they are relatively stable, so you can avoid re-

AT A GLANCE

◱ Designers should heed the rule of 80%: Buy an FPGA with 20% more LUTs (look-up tables) if you expect to hit high-performance targets.

◱ The biggest FPGAs can run at 550 MHz.

◱ The biggest FPGAs have 330,000 logic cells, or roughly 12 million equivalent-ASIC gates.

◱ FPGA-vendor synthesis tools are less efficient than commercial EDA FPGA-synthesis tools.

◱ When selecting an FPGA, observe the layout and hard-wired macro configuration; it can present layout challenges that affect timing.

spins, mask costs, and buying DFM (design-for-manufacturing) tools.

But Singh says that his group has two main reasons to use FPGAs. The first is that they allow his group to take a lot of functions off the PCB and integrate them onto an FPGA to speed performance and save PCB space. The second and most convincing reason to use an FPGA, according to Singh, is simply that ASICs' unit volumes sometimes don't justify the mask, design, and tool costs, and, most of all, the risk (**Figure 2**). "FPGAs have evolved, and, in many cases, they can meet your performance

and density requirements," says Singh. "If you are designing a midrange, mid-class ASIC, you should ask yourself, 'Do I want to spend $2 million to $3 million to do a 90- or 65-nm ASIC, or do I want to use a 90- or 65-nm FPGA technology with a comparable benefit?'"

Singh notes that his group tends to favor FPGAs over ASICs for designs that will have a new architecture and do not use a lot of blocks from previous designs. Because FPGAs are reprogrammable, the group can try new architectures and simply reprogram the FPGA when changes are necessary.

Similar to Singh's, Rozario's group also uses an FPGA rather than an ASIC, primarily for cost reasons. "When you're a start-up and money's tight, one of the first things to do is to look at an FPGA, because the development costs are a lot less and there aren't mask costs," says Rozario.
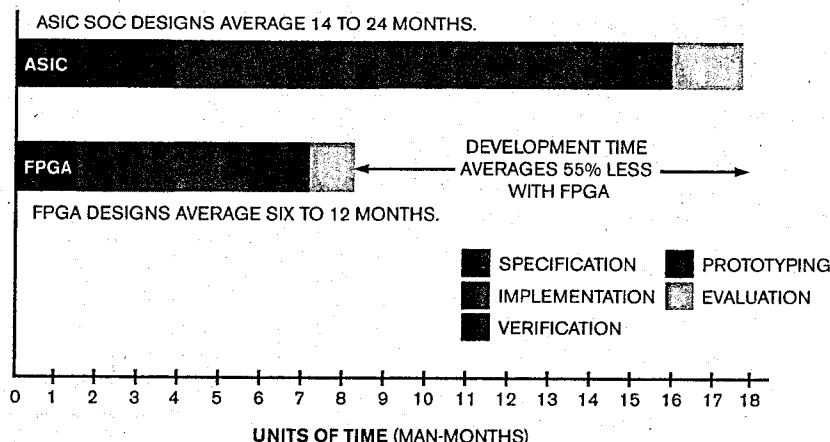
Rozario says that his company wanted to integrate and speed up some of the software functions in its product by implementing many of those functions in hardware on a single FPGA for its next-generation offering. "When we started, we really didn't know what functions we wanted to move to a chip and what functions needed to speed up," he says. He likes FPGAs because his group could add and subtract functions on the FPGA that they needed to speed up later in the process. "One of the great things about an FPGA is [that] you have the option of changes down the line."

Although the choice to go to an FPGA was fairly easy, however, Rozario and Singh say that figuring out how to design with an FPGA requires a bit of learning. Both engineers say that, after you decide to use an FPGA, the next big step is selecting the right one.

DETERMINE FPGA NEEDS

Designers need to look at what FPGA families are available from the various vendors and find the right mix of performance, power, and density. But buyer beware: One of the first things designers need to know when moving from an ASIC to an FPGA is that, to target a performance grade, you should buy a device with 20% more density than you need.

In most of the 10 FPGAs Singh has worked on so far, he says, he opted for

Figure 1 One of the overlooked facets of FPGAs is their relatively fast development.

ASIC SOC DESIGNS AVERAGE 14 TO 24 MONTHS.

FPGA DESIGNS AVERAGE SIX TO 12 MONTHS.

DEVELOPMENT TIME AVERAGES 55% LESS WITH FPGA

SPECIFICATION PROTOTYPING
IMPLEMENTATION EVALUATION
VERIFICATION

UNITS OF TIME (MAN-MONTHS)

SOURCE: *EXPERIMENTATION MATTERS: UNLOCKING THE POTENTIAL OF NEW TECHNOLOGIES FOR INNOVATION*, STEFAN H THOMKE, HARVARD BUSINESS SCHOOL PRESS, 2003. BASED ON A HARVARD STUDY OF 400 DESIGNS.

FPGAs because they fit his company's volume and performance needs. "In all those cases, we had to push the designs to the edge with respect to frequency, I/O time, and usage," he explains. A rule of thumb his group uses is that, if you are using 60 to 75% of the design's overall real estate, small changes from one compilation to another will give you a good chance of meeting your performance requirements and doing well in the lab. However, he notes, if you cross the 85%-usage mark, you may be unable to obtain the performance you need.

Singh says that his group has designed FPGAs with as much as 95% usage and reached performance goals, but doing so took a lot of work. "You have to be very familiar with the FPGA and how it works," says Singh. "In the ASIC world, you can write TCL (Tool Command Language) scripts and query the database to get the stuff you want, but, in the case of FPGAs, the tools are not as mature. They are typically GUI [graphical-user-interface]-based."

Rozario says that, when he first shopped around for FPGAs, the fact that vendors had improved the devices to meet most performance and density grades impressed him. However, FPGAs still don't reach the same top speeds or density grades as ASICs. The top-of-the-line SRAM-based FPGAs, when you scale them back in functions and optimize them to the fullest, top out at 550 MHz; ASICs can double that performance at top speeds. The biggest commercially available 65-nm FPGA, the Xilinx Virtex-5, has 330,000 logic cells, or roughly 12 million equivalent-ASIC gates, according to Xilinx.

Rozario was initially worried about FPGA-performance limitations. But the speed and density of the FPGAs surprised him. "If you are used to designing an ASIC, your expectations for performance will definitely have to come down in an FPGA," he cautions. Rozario notes that, in the first design, his group used 80% of the Virtex-5 LX 220 real estate and hit the design project's performance goals.

FPGA vendors today typically offer application-specific variants of their devices. A given FPGA-product family typically features a traditional sea-of-gates FPGA and variants that target
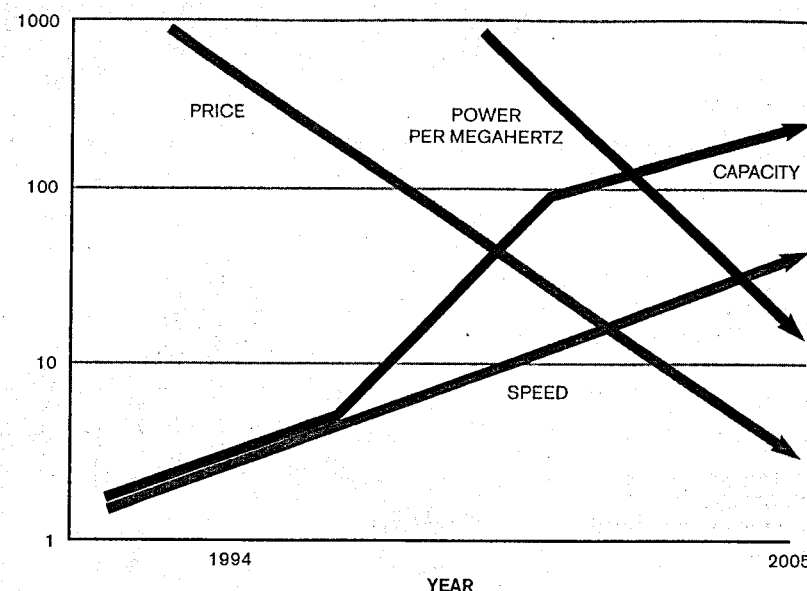
SOURCE: XILINX HISTORICAL DATA.

Figure 2 FPGAs have improved on all fronts: capacity, power, performance, and cost.

specific markets. Some devices target network applications and contain hard-wired SERDES (serializer/deserializer) cores; other devices target communications applications and contain hard-wired DSP blocks. All contain fairly large amounts of memory. Xilinx, for example, offers the Virtex-5 LX for high-performance logic, the Virtex-5 LXT for high-performance logic with serial connectivity, the Virtex-5 SXT for high-performance DSP with serial connectivity, and the Virtex-5 FXT for embedded processing with serial connectivity.

Singh and Rozario say that choosing the right device for your application is imperative, because FPGAs with unneeded hard-wired cores can consume real estate and may cause layout roadblocks later in the design cycle that could hinder you from achieving your performance goals.

Neither designer is targeting an application in which power is a primary concern, but Singh says that design groups designing for low power have to take the power-consumption issues of FPGAs into consideration, even though FPGA vendors have been making strides to control overall power consumption and leakage in the 90- and 65-nm nodes.

The only step Singh's group has taken to combat the power problem, he says, is shutting down hard-wired, 5-Gbps SERDES with the clock-gating technique. "The new FPGAs deal with low power fairly well, and there are a lot of techniques you can employ to lower power, but we haven't had to use them, yet," he says. Of course, that situation could change as server applications begin to rely on power savings as a selling point.

ASICs of the same performance and node generally consume much less power than FPGAs, though FPGA vendors are making progress on that front. Xilinx and Altera claim to have stabilized leakage power so that their 65-nm devices leak no more power than their 90-nm devices do.

FITTING SYSTEM NEEDS

In addition to ensuring that FPGAs meet performance, density, and power goals, designers also must consider the FPGA's impact on the IC package and the PCB. FPGA chips typically consume more area on a PCB, and their dense I/O often requires designers to add more layers to a PCB to handle the routing of this I/O. They also need more diligent

package and PCB-signal-integrity analysis and enough room to accommodate the extra power-supply circuitry to properly power the FPGA. All of these requirements can add costs to the design cycle and ultimately the end product.

Rozario notes that, for this design, the group had no other devices running at 1V, so, to accommodate the FPGA, it had to put an extra power brick on the PCB to power the device. This step was no problem, he notes, because the PCB has little more than the FPGA on it.

Beyond picking a part that meets performance, power, and density goals for your system, you should also look at what tools FPGA vendors and independent EDA vendors offer (Figure 3). Years ago, Altera encountered a significant setback when its Quartus development suite had usability issues. The company has since fixed those problems, but users must heed tool availability and quality.

TOOLS AND DESIGNS DIFFER

Both Singh and Rozario say that, although FPGA-vendor tools are fairly good for pushbutton use and typically free if you buy enough silicon, they are less sophisticated than ASIC tools. That is, you can't manipulate them as much as ASIC tools to perform custom tasks. In particular, Singh says, FPGA vendors offer decent compiler or synthesis technology, but the FPGA-vendor synthesis tools do not implement designs as well as or as efficiently as commercial FPGA-synthesis tools, such as those from Synplicity, Mentor Graphics, and Magma Design Automation. Some FPGA vendors act as OEMs for commercial EDA tools and offer them to customers in a "lite" configuration, targeting only that vendor's FPGAs, at little or no cost. In most cases, those tools are better than the FPGA vendor's tools but lack the features of the commercial, full-priced version of the EDA vendors' tools. Also, in the FPGA world, for the most part, users have no choice but to use their
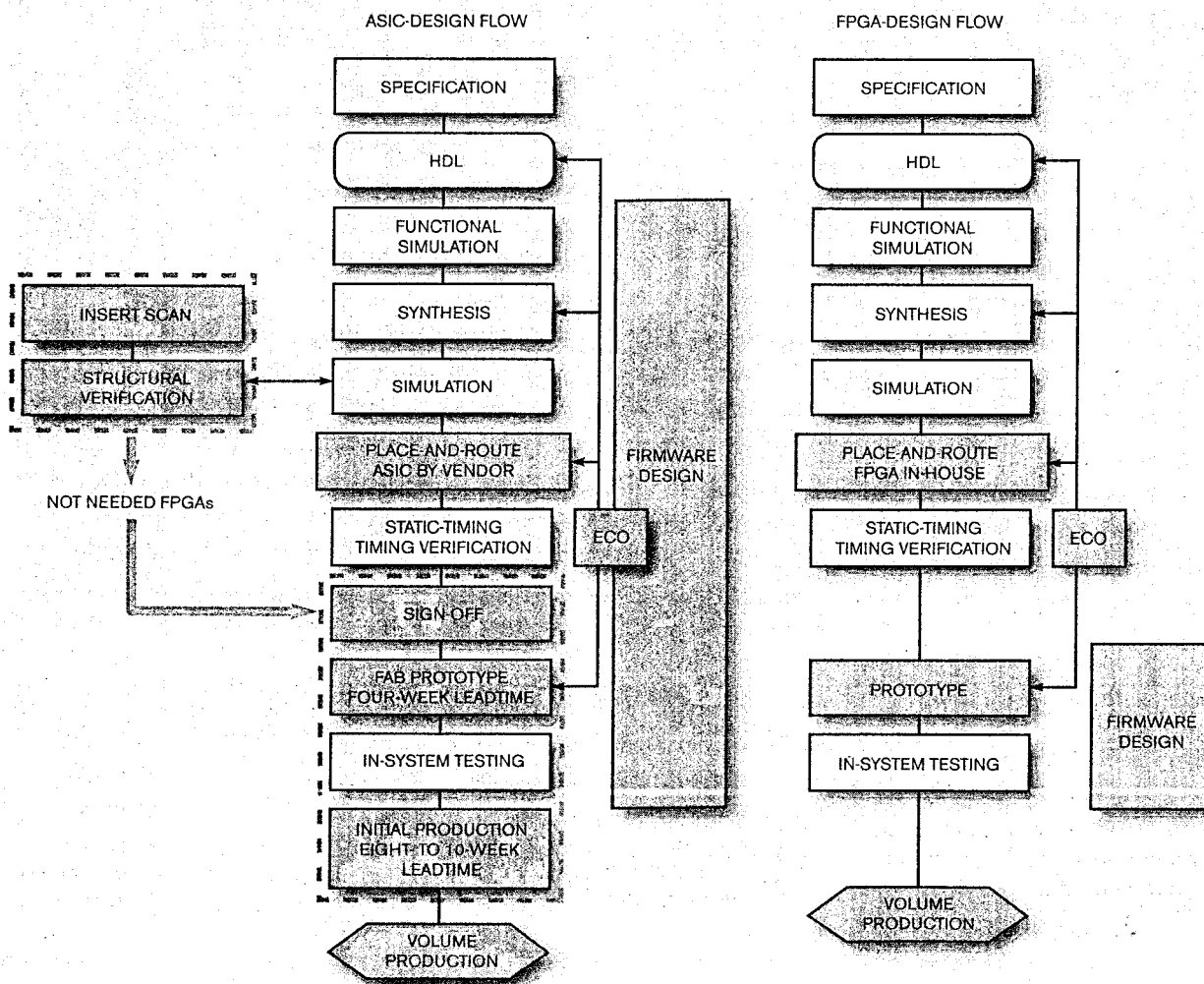

Figure 3 FPGA-development flows are far less complex and thus less expensive than ASIC flows.

FPGA vendor's internally developed physical-design tools. FPGA vendors develop their own physical-design tools to help users take advantage of their FPGA architectures. But, Singh and Rozario say, like ASIC synthesis, the layout tools are rarely as sophisticated as equivalent-ASIC tools.

According to Singh and Rozario, layout of an FPGA can be tricky because many FPGAs have fixed macros—SERDES, RAM, PLLs (phase-locked loops), and DSP cores—and some have fixed microprocessor blocks.

Altera, for example, offers the Stratix and Stratix GX lines. Singh says that the two FPGAs are similar, but the GX includes hard-wired SERDES blocks. Once you locate the RAM, PLLs, and dedicated I/O blocks, he explains, you need to design from the bottom up: Obtain the pinout, package, and macros on the die, and then do your planning and implementation. Challenges with I/O technologies, SSTL (stub-series-terminated logic) for DDR, and HSTL (high-speed-transceiver logic) for PCI have taught Singh the importance of due diligence, knowing what each device offers and what you need to do to design with it.

Singh also notes that, although ASICs have multiple clock resources, FPGAs are more limited and typically have global clocking. "If you have an ASIC design and you are thinking of moving to FPGA but it contains multiple clock domains—particularly, large clock domains—then you have to work with the FPGA vendor [to determine] whether the device can handle your design," says Singh. Besides global clocking, he says, FPGAs have local clocks. But, Singh warns, those clocks are limited to certain quadrants, so be careful about your logic and its clocking. Singh's designs employ multiple clock domains, typically 15 to 20, which require a great deal of work to implement in FPGAs. "With ASICs, you can tune your I/Os," he explains. "FPGA I/O is complex, and tuning it can tax your overhead because it's difficult to tune slew rates, drive strength, and impedance." According to Singh, signal integrity and clocking can both be big issues if you fail to properly tune I/O from the beginning of the design.

Rozario's group experienced similar issues with layout. Because his design
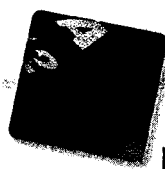
"WE DO FUNCTIONAL SIMULATION, GATE-LEVEL SIMULATION, AND DYNAMIC SIMULATION, TAKING THE GATES ALONG WITH PARASITICS BACK INTO FUNCTIONAL SIMULATION."

includes relatively few clock domains, however, the clock structures were simpler and sufficient for his group's design. "FPGAs come with built-in clocks throughout the fabric, so clock balancing was a lot simpler than with an ASIC," Rozario says. "You don't have to worry about the H-tree signal integrity because the FPGA architecture has taken care of all the clocking, and you can have a large number of clocks per region."

Rozario warns that a poorly planned layout can result in timing problems. "Some of the larger FPGAs have functions that you don't necessarily need, and you have to work around them. The device we are using has hard-wired DSP functions. We aren't using a DSP in this design, but we may use it in the next one." Once his group established the optimum layout, the Xilinx ICE (in-circuit-

emulator) tools did a "fairly good job" of adapting the design to the architecture. Initially, his group would floorplan every FPGA submodule and bring it up to the top level and then repeat that method for the entire chip. But ICE-application engineers would caution that the group's method was against their recommendation and urge the group to use automated tools. Rozario was skeptical but found that the results for both methods were similar.

Although FPGA tools are less customizable than ASIC-design tools, Singh encourages ASIC designers moving to FPGA design to take advantage of the free FPGA-vendor tools, IP (intellectual-property) portfolios, and customer support. He cites the vendors' good program management, architects, and design specialists that understand FPGAs, as well as field support. He finds the tools valuable for groundwork, such as floorplanning and signal integrity.

VERIFICATION ISSUES

When FPGAs first hit the market, one of their big selling points was that designers could program them and then directly test them on a board running in a mockup of the system, skipping simulation-based logic verification. Many FPGA veterans still use this approach. But Singh and Rozario say that, even midrange FPGAs are now generally too large and complex to simply program, run on a board, and debug during system test and bring-up.

Many veteran FPGA designers have told Rozario that they design the chip, skip logic verification, and test the chip in the lab. Rozario doesn't support that method, because, although the outsides of his chips have standard bus interfaces, the insides are complex, and debugging without visibility inside the chip is difficult. "It's a lot easier having waveforms, so we stuck with the ASIC-verification approach," he says. "We'll still debug in the lab, but we try to get most of the bugs out before we get to the lab."

Similarly, Singh says, because his group uses only the largest FPGAs, it employs the same verification methodology for FPGAs as it does for ASICs and runs each design through simulation with a testbench, debugging, and timing analysis.

"We do functional simulation, gate-

level :
tion, t
sitics
Then,
ing ar
modes
job th
and p
the fir
have t
then i
and ea
metho
sults. F
ulation
blocks,
ros tha

SOFT

In a
tween .
based l
new cl
occur v
tron co
error o
(Refere
are not
rors, bt
other ty
nies suc
base the
SRAM

Singt
to more
Thus, h
you nee
And, ev
FPGA :
add som
tion coc
based d
tion to
rors—fli
introduc

FOR I

Actel
www.actel

Altera
www.alter

Hewlett-P
www.hp.cc

Intel
www.intel.

Lattice Se
www.lattic

Magma De Automatio
www.magn

level simulation, and dynamic simulation, taking the gates along with parasitics back into functional simulation. Then, we do a very thorough static-timing analysis over multiple corners and modes," says Singh. "Once we do a good job there, we then mount it on a board and power it up." He concedes that the first time you use this method, you have to go through a few iterations, but then it becomes a standardized process and easier on subsequent projects. The method typically brings Singh good results. FPGA vendors provide great simulation models for their hard and soft blocks, SERDES, PLLs, and other macros that help his group's verification.

SOFT ERRORS

In addition to subtle differences between ASIC and FPGA design, SRAM-based FPGAs also present users with a new challenge: soft errors. Soft errors occur when a random atmospheric neutron collides with an IC, causing a bit error or, in some cases, a false signal (Reference 3). Standard-cell devices are not typically susceptible to soft errors, but SRAM structures, logic, and other types of memory are. And companies such as Xilinx, Altera, and Lattice base their highest performing FPGAs on SRAM.

Singh says that his group has started to more frequently encounter soft errors. Thus, he warns, when picking a device, you need to ask for a reliability study. And, even if the vendor claims that the FPGA is reliable, he says, you need to add some amount of ECC (error-correction coding) to your design. An SRAM-based device needs an ECC configuration to make sure there are no false errors—flipped bits that a neutron strike introduces.

If you have control circuitry in your design that checks the integrity of your configuration RAM, and, every once in a while, the configuration of your RAM changes, you've most likely encountered a soft error. Soft errors are uncommon in ASIC designs, but they can affect all the SRAM-based segments of an FPGA, including SRAM-based logic structures and on-chip SRAM blocks. "With transistors getting smaller and [device] voltage dropping, you can't help it; you're always prone to getting hit by a neutron," says Singh.

Fortunately, vendors such as Actel and Lattice have answers to the soft-error problem as they provide flash-based, nonvolatile FPGAs. These devices are neither as fast nor as dense as SRAM-based devices, but they are soft-error-resistant. The nonvolatile devices are also gaining popularity because they offer better security for companies selling into geographic markets in which piracy controls are still questionable.

In the face of rising ASIC-mask, -development, and -tool costs, FPGA vendors are presenting designers with a viable option for quickly getting designs out the door. But, before designers take the leap, they need to research FPGAs and weigh each vendor's architecture, design tools, and technical support. Designers also must understand the limitations of FPGA architectures and augment design techniques accordingly.**EDN**
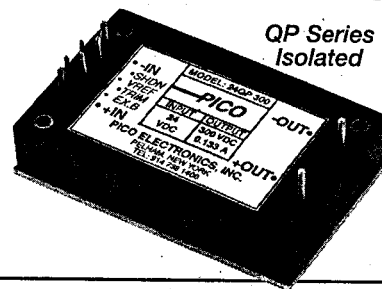
REFERENCES

■ Santarini, Michael, "FPGAs balance lower power, smaller nodes drip by drip," *EDN*, June 8, 2006, pg 58, www.edn.com/article/CA6339245.

■ Santarini, Michael, "ASIC prototyping: make versus buy," *EDN*, Nov 21, 2005, pg 30, www.edn.com/article/CA6283831.

■ Santarini, Michael, "Cosmic radiation comes to ASIC and SOC design," *EDN*, May 12, 2005, pg 46, www.edn.com/article/CA529381.

FOR MORE INFORMATION

Actel www.actel.com	**Mentor Graphics** www.mentor.com
Altera www.altera.com	**Quicklogic** www.quicklogic.com
Hewlett-Packard www.hp.com	**Sonoa Systems** www.sonoasystems.com
Intel www.intel.com	**Synplicity** www.synplicity.com
Lattice Semiconductor www.latticesemi.com	**Toshiba** www.toshiba.com
Magma Design Automation www.magma-da.com	**Xilinx** www.xilinx.com

You can reach Senior Editor Michael Santarini at 1-408-345-4424 and michael.santarini@reedbusiness.com.