

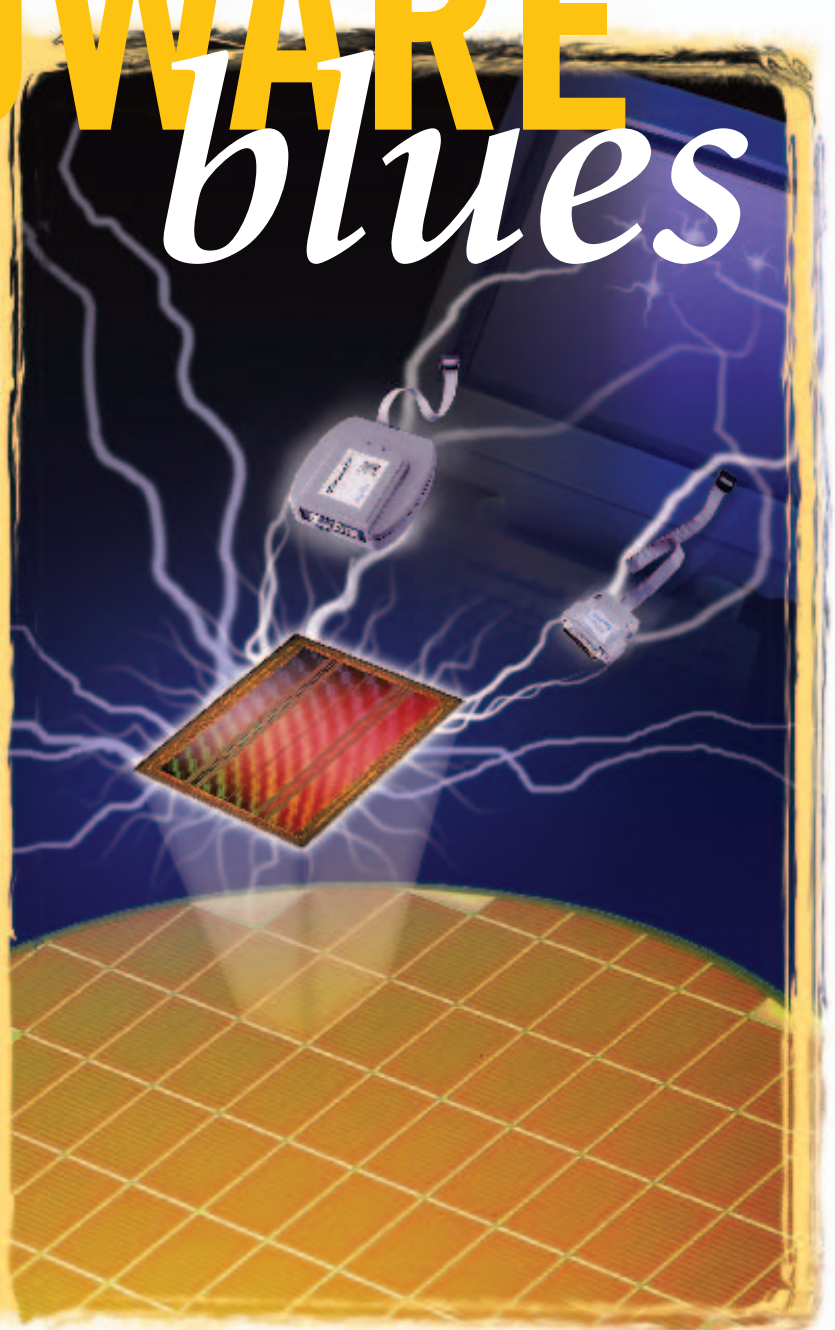
Tools help you lose the **HARDWARE** *blues*

EDA SOFTWARE SEEMS TO BE THE ONLY SILICON ASSISTANCE THAT SOME PROGRAMMABLE-LOGIC COMPANIES WANT TO TALK ABOUT NOWADAYS. WHAT OTHER TOOLS ARE AVAILABLE TO SIMPLIFY YOUR TASK OF—AS THESE SUPPLIERS SEEM TO HAVE FORGOTTEN—DESIGNING

HARDWARE?

<i>At a glance</i>	37
<i>A tools-table teaser</i>	38
<i>For more information</i>	42

Photo courtesy Altera



IN THE EARLY DAYS of semiconductors, digital-system design was almost completely a hardware-intensive-tools task. User-customizable-logic chips (with the exception of software-configurable microprocessors) just didn't exist. Instead, you dealt with

collections of transistors, subsequently followed by discrete-logic gates, and then by single-chip assemblages of multiple gates. You interconnected them on the system board to construct a desired function, maybe with the assistance of a DeMorgan Law logic-transformation program, but more likely with nothing more glamorous than a No. 2 pencil and paper. And when, invariably, the circuit's function didn't match what you expected, you powered up your trusty oscilloscope and logic analyzer to track down what went wrong.

The hardware-to-software tools shift began with PAL (programmable-array-logic) devices. Now, you could interconnect AND and OR gates, inverters, and flip-flops via on-chip (rather than on-board) routing resources. First low-level ABEL (advanced Boolean equation language), then higher abstraction-level design compilation and fitting software, created the bit streams that configured the devices' routing fuses, antifuses, and pass transistors. Next came design-simulation software, which let you test your circuits' functions before committing them to silicon, a capability of particular economic advantage if the device wasn't reprogrammable, and a misconfiguration, therefore, meant an expensive addition to the trash bin. And as the chips' routing architectures grew more hierarchical—culminating in the fine-grained and segmented FPGAs—dual-pass simulation, which combines a speedy first-pass functional test after design compilation with a more in-depth timing-inclusive simulation after design place and route or fitting, gained prominence.

Nowadays, multimillion-gate FPGAs and 1000-macrocell CPLDs, the epitome of VLSI, integrate so much and such diverse analog and digital circuitry and memory that they might, at first glance, seem to make hardware tools obsolete. Simulation software, howev-

er, covers only conditions that you explicitly describe in your test-vector file. What about those "real-world" combinations of input signals and sequences of those signals that you haven't predicted? Clearly, relying only on simulation is insufficient; old-school debugging techniques still have value. Throughout the four main stages of product development—definition, design, prototyping, and production—plenty of other opportunities for hardware help exist, too.

THE EVOLUTION OF EVALUATION

Before you can begin designing with a chip, you have to *pick* a chip. In appraising various vendors and the options within each vendor's product portfolio, it's often beneficial not only to compare data-sheet specifications, but also to see how well the device and its corresponding development-tool set tackle the types of circuits you commonly use in your designs (**references 1 and 2**). This hands-on assessment is the primary function of evaluation boards, as their name suggests (**Figure 1a**).

Take a look at any of the chip vendor's Web sites, which list both the vendors' and their partners' evaluation boards, or at an independent resource, such as the Programmable Logic Jump Station, and you'll discover long lists of products. They all contain some means of downloading

data to the PLD or the FPGA. They usually also offer some method of obtaining visible feedback that the chip is operating properly, such as blinking LEDs or a single- or multiple-segment LCD. Aside from the fact that the boards contain different device architectures and different device sizes within an architecture, what other feature disparities justify this multiplicity and, often, seeming duplicity?

If the device you're evaluating supports separate

AT A GLANCE

- ▶ Testdrive chip and software options during system definition before you proceed to design.
- ▶ Application-tuned device features have transformed evaluation kits into more focused but fuller featured reference designs.
- ▶ STAPL and IEEE 1532 further expand vendor-agnostic JTAG. Stand-alone programmers offer a higher performance alternative, particularly for production.
- ▶ As board traces have transformed into internal routing lines in the system-on-chip integration era, embedded-logic analyzers still let you probe around.

voltages for the core and I/O buffers, and if your design will take advantage of this feature, make sure that the board supports corresponding voltage flexibility. Some chips, in fact, offer multiple I/O banks, each capable of running at a different voltage range. Multiple clock inputs or onboard crystal oscillators provide additional evaluation-board-usage options. Test points enable you to monitor device inputs and outputs with an oscilloscope or a logic analyzer and are particularly valuable with hard-to-probe narrow- and numerous-lead packages, such as QFPs. Test points are *essential* when you use BGAs, because the package lids completely obscure their board-connection matrices (often to internal-layer board traces), making them otherwise unable to be probed.

Some evaluation boards contain a breadboard area to which you can attach other devices you plan to include in your end design. Socket adapters from companies such as Emulation Technology modify surface-mount packages to comprehend the through-hole requirements of this breadboard area. In addition to, or perhaps instead of, a supplemental prototyping area, some boards come with other preinstalled chips: banks of DRAM, SRAM, and flash memory; microcontrollers; UARTs; keyboard and mouse controllers; ADCs and DACs; and others. Other boards handle expansion via daughtercards in PC-104, PMC, and other standard formats.

Beyond the hardware differences among various boards, you find varying amounts of supplied documentation including, in some cases, schematics and Gerber-layout files. You also come across different mixes of bundled software. Whereas a chip manufacturer's board might reflect a partnership with Synopsys, for example, a third-party board might include a Mentor Graphics or Synplicity tool set. One board's tools might support only a single device or subset of



Figure 1

Evaluation boards let you test-drive product features before you buy and are applicable to both chips (a, courtesy Lattice Semiconductor) and embedded programmable logic (b, courtesy Actel). Some boards are tailored to specific product features and capabilities, such as “hard” (c, courtesy Atmel) and “soft” (d, courtesy Altera) CPU cores.

the device range, whereas another's might support an entire product family, multiple product families, or even products from multiple chip suppliers. Distributors are now even offering evaluation boards and corresponding technical support in the hope that when your design goes into production, they'll get your business.

Evaluation boards aren't restricted solely to programmable-logic chips. Actel provides a developer kit that lets you try out the company's Varicore embedded FPGA cores (Reference 3, Figure 1b). The evaluation board includes a 4x4-logic-block Varicore test chip with supporting logic, an LCD, both a breadboard area and a daughtercard connection, nonvolatile memory, serial- and

JTAG-configuration options, cables, documentation, and a user guide. Integrated Circuit Technology plans to offer similar boards for its long-delayed embedded-PLD technology, but embedded-FPGA supplier Adaptive Silicon currently has no evaluation aspirations.

REFLECTING ON REFERENCES

Programmable logic's generic nature has historically enabled it to serve a plethora of applications. In recent times, though, the suppliers have integrated additional features, such as embedded memory, DLLs, PLLs, high-speed I/O buffers, and arithmetic-function blocks, all with specific high-volume applications in mind. And, correspondingly, the devices' evaluation boards have become

A TOOLS-TABLE TEASER

Not all specs and prices for vendors' evaluation boards, reference designs, programmers, and logic analyzers arrived by press time. Check out the version of this article at www.ednmag.com for tables containing this information.

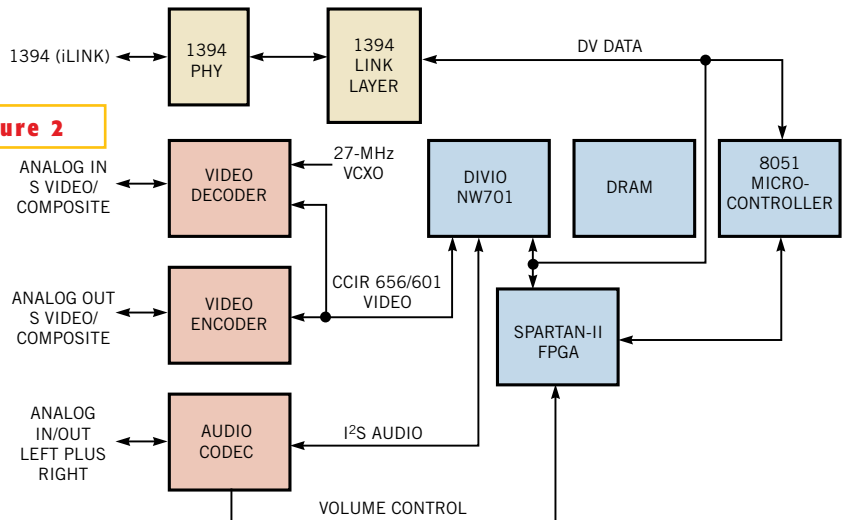
more optimized for applications and functions, transforming themselves into reference platforms applicable to both device evaluation and subsequent system design.

One of the first examples of this trend began when, in response to chips such as Xilinx's now-discontinued XC6200 series, a number of third-party vendors unveiled reconfigurable-logic boards. These development platforms are notable for their large amounts of on-board configuration memory (either nonvolatile or volatile), along with high-speed interfaces to external hardware. Instead of a serial JTAG cable, for example, they might support a parallel wiring harness or even take the form of an ISA or a PCI add-in card. All of these features—boost partial- and full-reconfiguration performance. An Ethernet controller, intended for remote configuration purposes, may even be onboard. These boards also typically include reconfigurable-computing-tailored design-and-debugging software, such as C compilers and other software-to-hardware migration utilities.

Xilinx has been aggressive in tailoring its collateral and tools for specific applications. The company devotes an entire area of its Web site to what it calls “eSPs” (emerging standards and protocols), areas in which the company sees compelling high-growth opportunities for its chips. At the eSP portal (www.xilinx.com/esp), you'll find information on wired interfaces, such as HomePNA, IEEE 1394, and USB 2.0; wireless interfaces, such as Bluetooth and IEEE 802.11; and other applications, such as encryption and decryption, and digital video encoding and decoding. You'll also find a number of reference designs, the fruits of Xilinx's partnerships with companies such as 4Links (IEEE 1355), Broadcom (Bluetooth), Convergent Design and Digital Harmony (IEEE 1394), Divio (video), and Kawasaki LSI and Mentor Graphics (USB 2.0) (Figure 2).

PLDs and FPGAs have always been a natural fit in other manufacturers' evaluation boards, which tend to have production runs of only dozens or hundreds of units, because, unlike ASICs,

Figure 2



A video-codec reference-design kit that Xilinx developed in partnership with Divio represents just one of the numerous applications that has captured the attention of the eSP program.

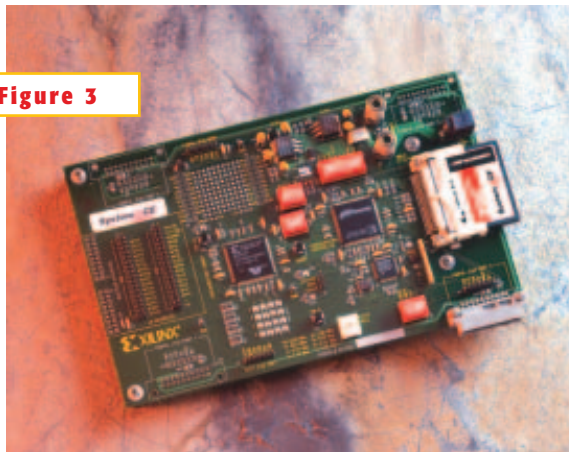
they don't have minimum-order quantities. In recent years, programmable-logic companies have taken a more active approach to ensuring their presence in others' reference designs in applications that represent large amounts of business potential. The advantage to those of you who work in one of these application areas, regardless of what company provides the reference design, is that you can directly apply the supplied schematics, design files, and software source code to your projects.

Reference designs might focus on a certain device feature that you can apply to a variety of applications rather than on a specific application. This trend first

gained prominence when devices got large and fast enough to house PCI initiator and target cores and has gained momentum as those and other cores have migrated onto dedicated ASIC silicon. QuickLogic, one of the first companies to move into the programmable-logic/ASIC “hybrid” space, with its ESP (Embedded Standard Product) line, lists on its Web site more than 10 reference-design kits that allow you to exercise the features of its QuickDSP, QuickFC, QuickMIPS, QuickPCI, and QuickSD products.

Other hybrid chip suppliers, such as Altera (Mercury and ARM-based Excalibur), Atmel (FPSLIC), Cypress (Programmable Serial Interface), Lattice (via its acquisition of Agere Systems' FPGA line), NEC (SoC Lite), Triscend (E5 and A7), and Xilinx (Virtex-II Pro), have similar ambitions, if not equally numerous board options (Figure 1c). Feature-optimized tool kits aren't restricted to “hard” CPU cores; both Altera (Nios) and Xilinx (MicroBlaze) directly and in partnership with distributors supply CPU soft-core reference designs (Figure 1d). And, as devices grow, nontraditional configuration devices, such as CompactFlash cards, become more appealing. Evaluation boards let you try out these emerging alter-

Figure 3



Evaluation boards also let you try out new configuration-download and -storage options (courtesy Xilinx).

natives, and, with the appropriate software stack, you might even be able to plug network cards into the Compact-Flash-module connector (Figure 3).

GET WITH THE PROGRAM

Now that you've got a design complete and ready to fire up, how do you get it into the programmable-logic device or, for SRAM-based parts, its companion configuration memory? IEEE 1149.1, commonly known as JTAG, is one of today's most popular device-programming techniques (figures 4a and 4b). After years of fractious debate between Altera and Cypress' Jam, Lattice's ispVM, and other vendors' proposals, the next-generation STAPL (Standard Test and Programming Language) in-system algorithm has secured broad-based industry support. JEDEC-approved STAPL defines a file format containing each chip's programming information. The companion IEEE-defined 1532 specification, a superset of 1149.1, is a hardware standard defining the programming-algorithm details (<http://grouper.ieee.org/groups/1532>).

JTAG's test-access port is, in many respects, an ideal programming interface—especially if the device internally generates all the “extra” voltages necessary to alter stored contents—because it's dedicated to programming and testing and doesn't interfere with other device and board functions. However, its serial and



Figure 4

JTAG is an attractive programming option for both CPLDs (a, courtesy Lattice Semiconductor) and FPGAs (b, courtesy Actel); stand-alone programmers offer a higher performance alternative (c, courtesy Lattice Semiconductor).

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, go to www.ednmag.com/info and enter the reader service number. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

Actel
1-408-739-1010
www.actel.com
Enter No. 318

Altera
1-408-544-7000
www.altera.com
Enter No. 319

Atmel
1-408-441-0311
www.atmel.com
Enter No. 320

Cypress Semiconductor
1-408-943-2600
www.cypress.com
Enter No. 321

Integrated Circuit Technology
1-408-434-0678
www.icpld.com
Enter No. 322

Lattice Semiconductor
1-503-268-8000
www.latticesemi.com
Enter No. 323

NEC
1-408-588-6000
www.necel.com
Enter No. 324

Quicklogic
1-408-990-4000
www.quicklogic.com
Enter No. 325

Triscend
1-650-968-8668
www.triscend.com
Enter No. 326

Xilinx
1-408-991-2000
www.xilinx.com
Enter No. 327

OTHER COMPANIES MENTIONED IN THIS ARTICLE
Adaptive Silicon
www.adaptivesilicon.com
Agere Systems
www.agere.com
Advanced RISC Machines (ARM)
www.arm.com

BP Microsystems
www.bpmicro.com
Broadcom
www.broadcom.com
Convergent Design
www.convergent-design.com
Digital Harmony
www.digitalharmony.com
Divio
www.divio.com
Electronic Trend Publications
www.electronictrendpubs.com
Emulation Technology
www.emulation.com

First Silicon Solutions
www.fs2.com
4Links
www.4links.co.uk
Kawasaki LSI
www.klsi.com
Mentor Graphics
www.mentorg.com
Programmable Logic Jump Station
www.optimagic.com
Synopsys
www.synopsys.com
Synplicity
www.synplicity.com

SUPER INFO NUMBER

For more information on the products available from all of the vendors listed in this box, enter no. 328 at www.ednmag.com/info.

low-speed attributes contribute to its low programming performance. These characteristics may not be a problem in the development lab and during prototyping, but they can be an expensive bottleneck in high-volume production (Reference 4). Instead, you might want to harness a wide and fast on-board programming interface (for which you'll have to comprehend function multiplexing in your system hardware and software designs). Alternatively, you can program the chips before installing them on the system board (Figure 4c).

Chip suppliers often partner with programmer manufacturers to offer a version of the manufacturer's product under the supplier's name. For example, Actel's single-site Silicon Sculptor and multisite Silicon Sculptor 6X antifuse FPGA programmers are variants of BP Microsystems' programmers, bundled with Actel chip-specific software. Actel's Flash Pro for the company's ProASIC and ProASIC Plus flash-based FPGAs is the fruit of Actel's partnership with First Silicon Solutions and supports STAPL and IEEE 1149.1. Before deciding on a silicon-vendor-sourced programmer, investigate whether you can later upgrade its firmware to support other vendors' products. Otherwise, if you switch chip suppliers in the future, you might be left with an expensive paperweight. Spending more money up-front for the original manufacturer's more versatile version of the programmer might be cost-effective in the long run.

LOGICAL ANALYSIS

You've programmed your chip, you turn on the system power, and...nothing happens. Or something happens, but it's not even close to what you expected. Your design has already passed the simulation tests with flying colors. What do you do now? Test points on the board, along with careful connection of test clips to the chip's package leads, might let you analyze its inputs and outputs. But what about signals running around *inside* the chip?

For a long time, your only option was to route internal nodes that you wanted to inspect out to device pins. Silicon vendors love this alternative, because, as the number of nodes increases, so do the required device pin count and price. It's not ideal to you for other reasons, too. Unless you employ the services of a fortuneteller,

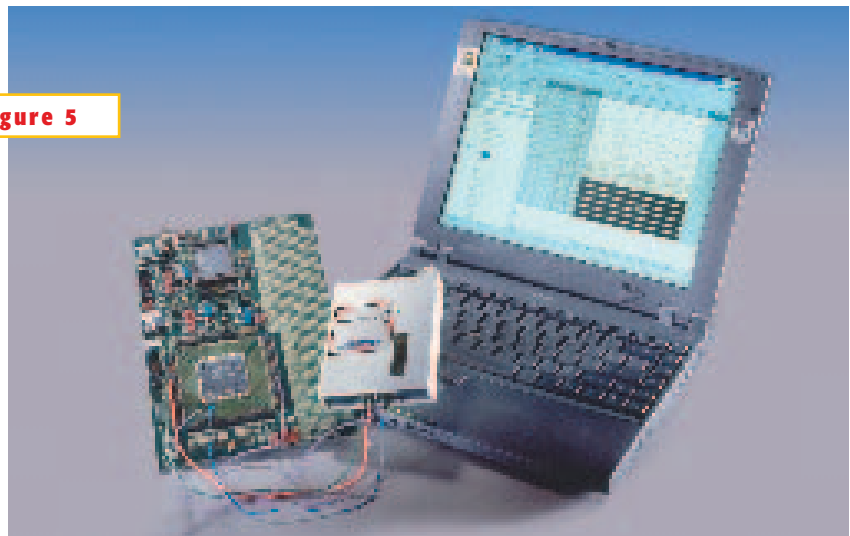
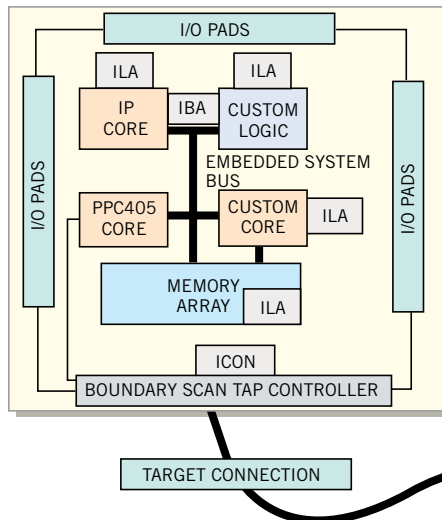


Figure 5

(a)



NOTES:
 ILA: INTEGRATED LOGIC ANALYZER.
 IBA: INTEGRATED BUS ANALYZER.
 ICON: INTEGRATED CONTROLLER.

(b)

Some embedded-logic analyzers employ devices' built-in logic circuits and routing resources (a, courtesy Actel); others take the form of a "soft" core (b, courtesy Xilinx).

you don't know in advance what errors your design has and, therefore, what nodes you want to bring out to pins.

Your lack of foreknowledge means that, to monitor internal nodes, you need to tap into incremental internal-routing resources. In other words, you need to recompile your design—a process that may change the design's timing and, inevitably, its function (obscuring the very phenomena you're trying to observe). Altera claims that the LogicLock and SignalProbe features of its latest Quartus II design software preserve your design and enable fast incremental compilation to bring nodes out to pins. But the Signal-

Probe documentation admits, "Signals might not always find successful re-route."

Keep in mind, too, that the added impedance of the logic analyzer or oscilloscope test probe alters the electrical characteristics of the internal node and, therefore, the circuitry that influences it and that it influences. Despite these potential pitfalls, routing nodes to pins might still allow you to gain valuable insight into what's wrong with the design and how to fix it. But now what do you do: Leave the added nodes-to-pins portions of the design alone and use a more expensive chip with a higher pin count

than you need, or remove the test circuitry and risk breaking the design again?

There has to be a better way. And Actel, followed by Altera and Xilinx, believes it has found it in the form of the embedded logic analyzer (Figure 5a). Actel's antifuse-chip implementations differ slightly from Altera and Xilinx's SRAM-based FPGAs, but the basic concept is the same: You tell the tool what nodes to monitor, for how many cycles, and with what trigger conditions. It communicates this information over the JTAG interface to circuitry within the chip, which sends the node data back out over JTAG or a dedicated trace port to a PC or logic analyzer for display.

CH-CH-CH-CH-CHANGES

Actel's Silicon Explorer 2 differs from the first-generation Silicon Explorer in several areas. (Ironically, Silicon Explorer was designed by a company that Altera subsequently acquired to develop SignalTap and the Nios CPU core.) It supports an external power supply, enabling it to probe devices with multiple operating voltages; delivers acquisition capability to 100 MHz; and offers four levels of triggering and decompression on download. Silicon Explorer 2 is unique among its

TABLE 1—SIGNALTAP PRIMARY-MODE LOGIC-ELEMENT USAGE

Channels	Apex 20K logic elements used*
One	136
Two	144
Four	160
Eight	192
16	256
32	384
64	640
128	1152

*Preliminary estimates

peers in that it can observe internal nodes in real time by tapping into hard-wired connections used for programming the FPGA antifuses. It outputs as many as two nodes' states over dedicated pins PRA and PRB and can monitor as many as 16 additional nodes via the FPGA's general-purpose I/O pins. Its Silicon Explorer 2 Lite variant enables only the Command module and relies on an external scope or logic analyzer for viewing signals.

Silicon Explorer 2 employs dedicated ActionProbe circuitry within the FPGA to implement the logic-analysis function. In contrast, Altera's SignalTap and Xilinx's ChipScope are soft cores, meaning that you have to compile them into the design (Figure 5b). Therefore, although they consume no incremental device-I/O buffers like the bring-the-node-to-a-pin technique or Actel's approach with more than two monitored nodes, they still consume incremental internal logic, memory, and routing resources (tables 1 through 4).

Once you debug your design using SignalTap or ChipScope, you probably

won't want to rip out the logic analyzer core and recompile the design. So, you still have a bigger chip than you'd otherwise need—but one containing a design that now works. Pick your poison, and ponder how big that logic-analyzer core will get when you start not just monitoring individual signal nodes, but also tracing entire buses (in conjunction with an embedded processor, for example). Actel's approach seems elegant in contrast, and, in many respects, it is, but keep in mind that you're dealing with nonrepro-

TABLE 2—APEX 20K LOGIC-ELEMENT RESOURCES

Device	Total logic elements
EP20K100	4160
EP20K100E	4160
EP20K160E	6400
EP20K200	8320
EP20K200E	8320
EP20K300E	11,520
EP20K400	16,640
EP20K400E	16,640
EP20K600E	24,320
EP20K1000E	42,240

grammable antifuse technology. Once you discover the problem, you have to discard the chip containing the faulty design and program a circuit that you hope is fixed into a fresh FPGA. And, unlike the more flexible soft-core approach, you can monitor only 18 signals at a time. Nothing's free, it seems. □

REFERENCES

1. Dipert, Brian, "Lies, damn lies, and benchmarks: The race for the truth is on," *EDN*, May 27, 1999, pg 54.
2. Dipert, Brian, "Synthesis shoot-out at the *EDN* corral," *EDN*, Sept 11, 1998, pg 95.
3. Dipert, Brian, "Do combo chips compute (or even compile)?" *EDN*, Feb 15, 2001, pg 101.
4. Dipert, Brian, "Focusing on chip programming," *EDN*, Jan 7, 1999, pg 58.

ACKNOWLEDGMENTS

Programmable Logic News and Views, edited by Murray Disman and published by *Electronic Trend Publications*, is a concise yet comprehensive monthly newsletter summarizing all that's new in the programmable-logic industry. Check out www.electronictrendpubs.com/plnbro.htm for sample content.

AUTHOR'S BIOGRAPHY



Technical editor Brian Dipert yearns for the good old days when programmable-logic design meant filling in Karnaugh maps and debugging meant probing low-pin-count DIPs (containing chips that he regularly blew up by inserting them in their sockets upside-down). You can reach Brian the Luddite at 1-916-454-5242, fax 1-916-454-5101, bdipert@pacbell.net, and www.bdipert.com.

TABLE 3—SIGNALTAP PRIMARY-MODE EMBEDDED-SYSTEM-BLOCK USAGE

Channels	Memory depth: samples				
	128	256	512	1024	2048
One					One
Two				One	Two
Four			One	Two	Four
Eight		One	Two	Four	Eight
16	One	Two	Four	Eight	16
32	Two	Four	Eight	16	32
64	Four	Eight	16	32	64
128	Eight	16	32	64	128

Embedded-system blocks required

TABLE 4—APEX 20K EMBEDDED-SYSTEM-BLOCK RESOURCES

Device	Embedded-system blocks
EP20K100	26
EP20K100E	26
EP20K160E	40
EP20K200	52
EP20K200E	52
EP20K300E	72
EP20K400	104
EP20K400E	104
EP20K600E	152
EP20K1000E	264