Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

EXTREME CPUS DEFY CONVENTIONS

Radical Designs Attempt Quantum Leaps in Performance

By Tom R. Halfhill {2/9/04-15}

Microprocessor architects love a challenge. But the greatest challenge may lie in finding a challenge. The world already has plenty of general-purpose CPU architectures—too many, some say—and their performance differences are relatively minor. Another new RISC, CISC, or VLIW instruction set won't start a revolution. What's an ambitious CPU architect to do?

An increasingly popular answer is to throw convention out the window and invent something radically different. The trick is to find an application that can benefit from a radical CPU design and has enough market potential to justify a risky development project. Then the architect needs a cool idea, a talented design team to execute it, and a great deal of funding to pay for it. When all those planets align, the result is an extreme processor.

MPR coined the term "extreme processor" a few years ago to describe unusual architectures or unusual implementations of conventional architectures. Because extreme processors tend to serve niche applications, they will never become as widespread as general-purpose processors. Yet they continue to appear regularly at Microprocessor Forum and Embedded Processor Forum—so many of them that *MPR* has created a special Extreme Processors category for our annual Analysts' Choice Awards. Why have so many extreme designs been surfacing in recent years?

One reason is that visionaries who are determined to create a new CPU architecture recognize the futility of competing directly with the established vendors of conventional architectures. To survive, a new architecture must be truly new and different. The last attempt to crack the PC market was in 2000, when Transmeta introduced an unorthodox VLIW architecture that relies on hardware-assisted software emulation for x86 compatibility. Despite a power-efficiency advantage, Transmeta is struggling to win more than a tiny market-share percentage.

The embedded-processor market has attracted more-lively competition, mainly because it isn't dominated by a single CPU architecture like the x86. However, it is overrun with general-purpose architectures, and any more would be redundant.

A secondary reason for the surge in new extreme-processor architectures is the growing demand for high performance in fairly narrow applications. For just one prominent example, take a telephone network, which used to consist of electromechanical switches that routed analog signals across miles of wires to stationary phones containing little more than a microphone, a speaker, and a pulse-tone dial. Now, in the most recent telephone networks, the switches are digital routers, the signals are data packets, the wires have been replaced by wireless base stations, and the phones may also function as two-way text pagers, Web browsers, walkie-talkies, PDAs, and digital cameras. This startling evolution is generating demand for new types of processors that deliver high performance and/or low power consumption for specific tasks.

Of course, Moore's law is another driver of extreme-processor architectures. As transistor budgets keep rising, the easiest response is simply to build larger caches into general-purpose processors. Lately, we've seen CPUs that are more properly described as memory chips with integrated processing; the SRAM arrays dominate the die photos. Larger caches do boost performance, at least for some applications. However, maverick CPU architects would rather spend the extra transistors on something more creative than rows and columns of SRAM cells spewed out by a memory compiler. One result of their efforts: a growing proliferation of massively parallel architectures.

For the 2003 *MPR* Analysts' Choice Awards, we have nominated the five **Best Extreme Processors** introduced last year. All challenge the status quo, and they meet our requirement of existing as sample chips or customer-deliverable cores by the end of 2003.

Our nominees are the ClearSpeed CS301, Cradle ECE3400/MPE3400, Intrinsity FastMath, Elixent ET1, and Xelerated Xelerator X10q. ClearSpeed and Xelerated announced and sampled their processors in 2003. The Elixent ET1 is actually a Toshiba chip based on the D-Fabrix processor core from U.K.-based Elixent, which announced and delivered the hard core to Toshiba in 2003. Intrinsity announced FastMath in 2002 and began shipping the processor in 2003. Cradle announced its processors in 1999, although the chips weren't named or introduced until 2003.

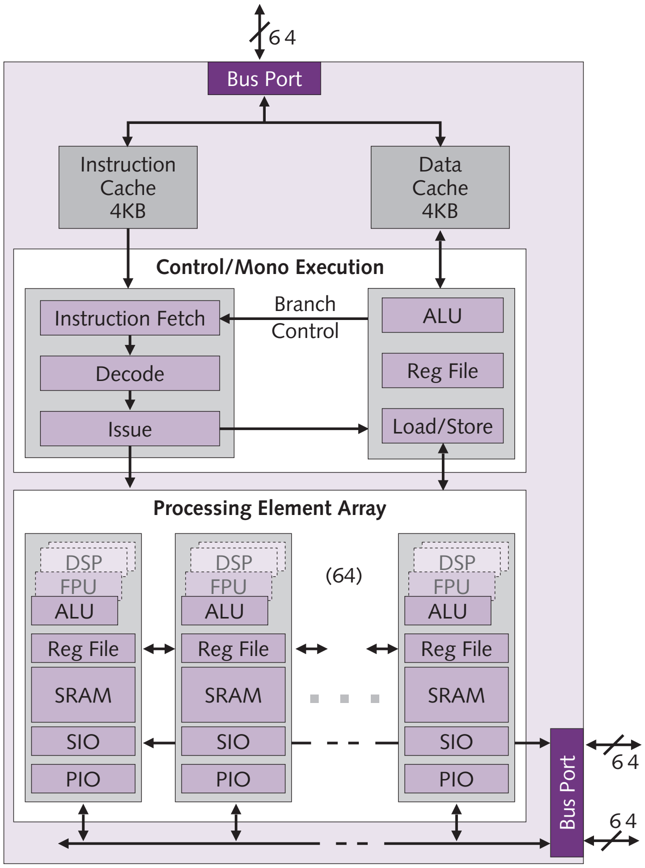

Figure 1. This block diagram of the ClearSpeed CS301 shows the control/mono execution unit, a processor-within-a-processor that executes some instructions itself and passes other instructions along to the array of parallel-processing elements.

This article describes all the nominated extreme processors and summarizes the attributes that make each special. For the winner, see the accompanying article, *MPR 2/9/04-16*, "Best Extreme Processor."

ClearSpeed Accelerates Floating-Point Math

At Microprocessor Forum 2003 last October, ClearSpeed's presentation sparked a frenzy: afterward, *MPR* received calls from reporters as far away as Brazil. Oddly, ClearSpeed generated the buzz by reviving the concept of the floating-point math coprocessor, a product category that faded away in the 1990s when Intel and others began integrating FPUs into their microprocessors.

Perhaps ClearSpeed's performance claims stirred up the excitement. The company says its **CS301** coprocessor can execute 25.6 peak GFLOPS (billion floating-point operations per second) at a clock rate of only 200MHz while consuming less than 2W of power. (See *MPR 11/17/03-01*, "Floating Point Buoys ClearSpeed.")

However, as we noted in our original report, the CS301's dazzling potential is limited by a 64-bit, 200MHz bidirectional I/O bus that provides only 1.6GB/s of off-chip memory bandwidth in each direction. That's significantly less bandwidth than a general-purpose PC processor and not nearly enough capacity for a math coprocessor that must crunch through large datasets. The CS301 will be further impeded when used as a PC coprocessor on PCI cards—one of the target applications—because a 32-bit, 33MHz PCI bus provides only 133MB/s of bandwidth.

Nevertheless, the CS301 has much to offer in compute-intensive (as opposed to data-intensive) applications that operate on manageable data sets. Each CS301 has special bridge ports that can link multiple coprocessors without additional loading on the main I/O bus. A single PCI card may contain two or three coprocessors, and a single system can use multiple coprocessor cards. ClearSpeed has demonstrated such a setup on an ordinary PC with six CS301 chips on three cards, sustaining about 30 GFLOPS. (See *MPR 1/12/04-02*, "ClearSpeed Hits Design Targets.")

The CS301 achieves its remarkable performance with a massively parallel architecture that has 64 independent processing elements, including 128 FPUs. Figure 1 shows a high-level block diagram of this design. ClearSpeed's proprietary architecture, called the Multithreaded Array Processor (MTAP) architecture, is best suited for programs that repeatedly apply an algorithm to a relatively small data set. One company demo is a drug-testing program that attempts to match the elements of a candidate drug with protein molecules.

Hardware-controlled multithreading is another salient feature of the MTAP architecture. It's similar to Hyper-Threading in Intel's Pentium 4 and simultaneous multithreading (SMT) in other processors, but it's primarily intended to allocate the CS301's compute and I/O resources more efficiently, not to eliminate the pipeline bubbles associated with context switching. The CS301 can simultaneously execute eight threads. Programmers can reserve one thread for system-level operations and another thread for compute tasks, and can use one or two threads for asynchronous I/O.

An instruction-set lookup table encoded in on-chip SRAM allows programmers to customize the CS301's instructions, even while a program is running. This is a rare and potentially powerful feature. Furthermore, the microarchitecture is customizable at design time, because, in addition to offering the CS301 as a standard part, ClearSpeed is licensing the synthesizable model as soft intellectual property (IP).

ClearSpeed hints that a follow-on design will boost the I/O bandwidth and have a more efficient die layout. Owing to last-minute design changes, the CS301 is a little larger than it needs to be and has some unused pins. The future chip will be something to watch, but even the CS301 is a daring and inventive design.

Cradle's Baby Finally Born

After eight years of labor, Cradle Technologies finally brought its unusual reconfigurable processor to silicon in 2003. The lengthy project started at Cirrus Logic in 1995, moved outside in 1998, and was publicly revealed at Microprocessor Forum in 1999. The first production samples came back from TSMC around the middle of 2003. (See *MPR 10/6/03-03*, "TSMC Delivers Chip to Cradle" and *MPR 10/6/99-05*, "Cradle Chip Does Anything.")

The design's complexity and the evolving market for protocol processing are likely reasons for the delay. From the outset, Cradle wanted to create an "ASIC killer"—a processor so powerful and flexible that customers would turn away from custom ASICs in video-communication and image-processing applications. Obviating the need for a costly ASIC project is the goal of many extreme-processor designs. Cradle's unique solution is a radical architecture optimized for data parallelism and reconfigurable I/O. In addition to multiple processing elements and DSPs, it has reprogrammable logic associated with the I/O pins, so developers can configure the I/O logic for different communication protocols.

Cradle's first two chips based on this architecture are the **ECE3400** and the **MPE3400**. *MPR* has nominated both processors for an Analysts' Choice Award because they are actually the same chip. At run time, special boot software (written in BOOL, a logic-design language) configures the processor's reprogrammable logic and 128 I/O pins for the appropriate protocols. The ECE3400 is configured for video applications, such as H.264 or G.729 videoconferencing, advanced set-top boxes, surveillance cameras, or Internet Protocol videophones. The MPE3400 is configured for image processing in multifunction laser printers, color copiers, and similar products. Customers can reprogram the logic of either processor for other I/O protocols and can write their application software in assembly language or in C.

The ECE3400/MPE3400 is a highly integrated device. It has six RISC-like processing elements, eight DSP cores, 192K of on-chip memory for instructions and data (not counting the reprogrammable I/O logic), and an SDRAM controller. As Figure 2 shows, a global on-chip bus ties everything together. The Cradle chip is large for an embedded processor—33 million transistors—but not excessively large for an extreme processor with reconfigurable logic. Like many extreme processors, it relies on a complex parallel design, not high clock frequency, to achieve high performance. The ECE3400/MPE3400 runs at 220MHz and consumes only about 3W at 1.2V.

High integration and reconfigurable logic can be good or bad. They're good if the target application can use a substantial portion of the processor's capabilities. They're bad if the application leaves too much of the processor lying fallow. The ECE3400/MPE3400 is best suited for data-intensive applications that leverage the parallelism and reconfigurable I/O: for example, systems that support evolving I/O protocols or are likely to need protocol updates in the field. Otherwise, customers are paying for reconfigurable logic they won't need.

At $50 in production quantities, Cradle's ECE3400/MPE3400 can be an attractive alternative to an ASIC in lower-volume applications. In high volumes, an ASIC would still pay off, but Cradle's chips are available as standard products, saving a year or two in ASIC development time. Now that Cradle has actual chips to test and benchmark, the tradeoffs will be easier to evaluate.

Intrinsity's Extreme Implementation

Intrinsity's **FastMath** is an oddball, even among extreme processors. While other nominees in this category have extreme architectures implemented in conventional logic, FastMath has a conventional architecture implemented in extreme logic. This is a wholly different approach that pursues

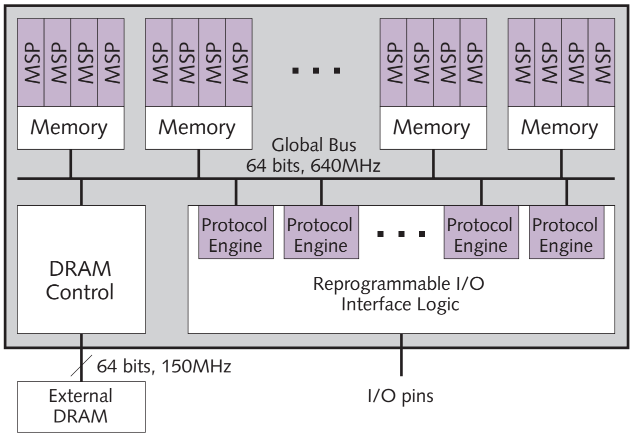

Figure 2. All function units and other internal elements of Cradle's ECE3400 and MPE3400 communicate over a single global bus. This bus moves code and data among external memory, the local memory blocks within each processing element, and the I/O protocol engines.

high performance by means of fast clock speeds while preserving compatibility with a standard instruction-set architecture (ISA).

For its unique attributes, FastMath won the *MPR* Analysts' Choice Award for the best extreme processor in 2002. (See *MPR 2/18/03-05*, "Extremely High Performance.") Because FastMath remains competitive, we nominated it again for a 2003 award.

The FastMath processor is based on the popular MIPS32 ISA. FastMath has a MIPS32-compatible core and a tightly coupled coprocessor called the Matrix Engine, which includes a 16-unit matrix of 32-bit DSPs. The MIPS core and Matrix Engine are connected to a 4GB/s on-chip bus, a 1MB integrated Level 2 cache, a double-data-rate SDRAM controller, a DMA controller, and dual RapidIO ports. This hybrid MIPS/DSP architecture allows FastMath to run both the control code for an embedded system and the math-intensive application code. Most other extreme processors must work alongside a general-purpose processor that runs the control code.

What really sets FastMath apart is Intrinsity's unique Fast14 logic. Fast14 is a circuit implementation that uses 1-of-N dynamic logic to maximize the speed of critical paths by assigning one wire or line to each binary value rather than encoding it as a 2× representation. Figure 3 shows an example of an OR gate implemented with this logic. Although Fast14 requires four wires instead of two wires to represent a two-bit binary value, its all-zeroes state helps precharge the faster dynamic logic and reduce the number of wires that must change level to represent different data. In addition, Fast14 uses four-phase clocking to avoid the race conditions common with other dynamic-logic circuits. (See *MPR 8/13/01-02*, "Intrinsity's Dynamic Designs.")

Thanks to Fast14, the most recent FastMath attains 2.5GHz at 1.2V when fabricated in TSMC's 0.13-micron process. (The original FastMath reaches 2.0GHz at 1.0V in the same process.) That's about ten times faster than some other extreme processors having more-complex architectures. Such stratospheric clock frequencies are taken for granted in desktop/server processors but rarely encountered in the embedded world. FastMath is truly a speed demon.

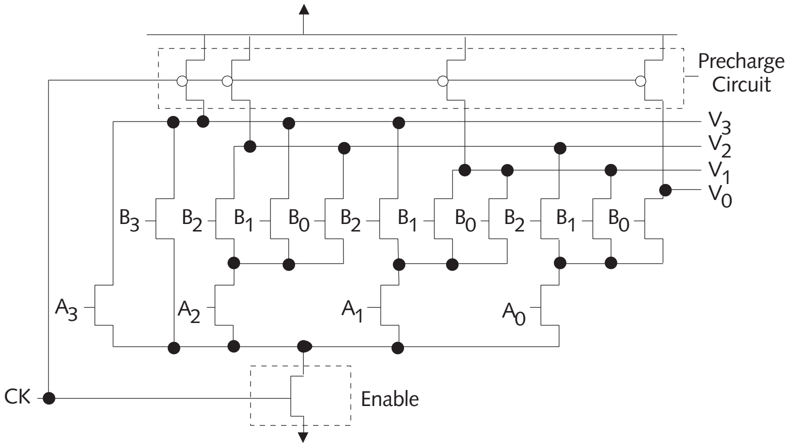

Figure 3. Intrinsity's Fast14 logic departs from the norm. This circuit diagram, taken from an Intrinsity patent, shows an OR gate with two 2-bit inputs and one 2-bit output, all represented using 1-of-4 coding.

However, high clock frequencies also boost power consumption (and prices), so Intrinsity has introduced somewhat slower and cheaper versions of the chip for customers not requiring maximum performance or unable to afford the 2.5GHz chip's 24W power envelope. (Intrinsity says it is updating its pricing, but it hasn't publicly announced anything.)

The 2.0GHz FastMath, already in production, consumes 16W at 1.0V. According to preliminary specifications, which may change when the chips enter production, the 1.5GHz FastMath consumes about 13W at 1.0V, and the new 1.0GHz FastMath-LP consumes about 6W at 0.85V. (See *MPR 5/27/03-03*, "Update on Intrinsity Fast Products.") Obviously, FastMath-LP isn't "low-power" in the ARM sense, but it is competitive with other extreme processors and with conventional embedded processors in the same performance range.

Toshiba Collaborates With Elixent

Not invented here? Who cares! Dismissing NIH prejudice, Toshiba has joined forces with a U.K.-based startup, Elixent, to create the **ET1** media processor. Their hybrid design starts with a 32-bit synthesizable RISC core: the Toshiba Media Embedded Processor (MeP), which includes a VLIW coprocessor. (See *MPR 6/10/02-02*, "New Processors For New Media.") The extreme part of the design is Elixent's D-Fabrix processor array, a massively parallel proprietary architecture. (See *MPR 7/21/03-01*, "Elixent Expands SoCs.")

Toshiba's MeP is the ET1's tightly coupled on-chip controller, much like the MIPS32-compatible core in Intrinsity's FastMath. However, Elixent's D-Fabrix is more exotic than FastMath's DSP Matrix Engine. It's a dense interconnected fabric of small processors, each having its own local instruction memory and register file. Elixent describes D-Fabrix as reconfigurable, but *MPR* prefers to call it reprogrammable or run-time programmable, because the configuration of the fabric is fixed at design time—there's no reconfigurable logic in the sense of an FPGA.

Instead, application developers "configure" the ET1 by programming the numerous processors in the fabric. Because each processor has its own instruction memory and registers, it can execute a task locally without continuously fetching instructions from off-chip memory. This architecture frees up more bandwidth for data on the fabric.

Elixent licenses D-Fabrix as a GDS-II macro with register-transfer-level (RTL) Verilog and VHDL models. It's more flexible than a conventional hard macro, because chip developers can use an Elixent tool called the Array Generator to configure the size of the fabric at design time. The basic D-Fabrix building block is a tile, which has two four-bit ALUs and six local registers. (Multiple four-bit ALUs can work in concert to process larger operands.)

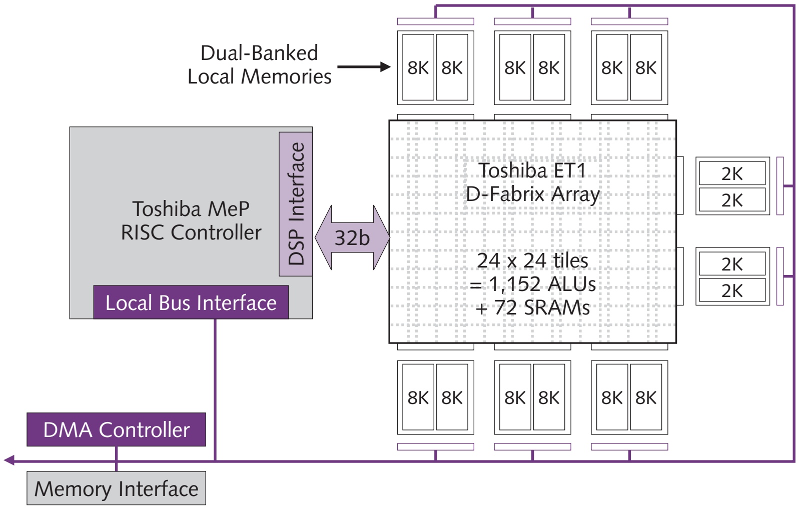

Figure 4. This block diagram of the Elixent ET1 shows all the processing and local-memory components of the D-Fabrix architecture. SoC developers can license this architecture as a hard macro, scale it to fit their applications, and integrate additional components.

As the block diagram in Figure 4 shows, Toshiba configured the ET1 to have a 24- × 24-tile fabric, with a total of 1,152 ALUs. In addition to the local memory in the tiles, dual-banked SRAMs surround the edges of the fabric; the ET1 has 104KB. Application software can use this memory as an I/O buffer, as temporary local data storage, or as storage for program code.

Neither Elixent nor Toshiba is saying much about the ET1. Elixent's application examples and informal benchmarks indicate that Toshiba will probably push the ET1 for applications like digital-video compression (MPEG), still-image compression (JPEG), and software-defined radio. If so, the ET1 will encounter stiff competition—both from other award nominees in this category and from dozens of other processors vying for a piece of the lucrative communications and media-processing pies.

Xelerated's Superduperpipelined NPU

Microprocessor design doesn't get much more extreme than a single chip that has 200 processor cores and a 1,040-stage pipeline. That's not a misprint. If you laid 52 Hyper-Pipelined Pentium 4 processors end to end, they would have a pipeline as long as the one in Xelerated's new **Xelerator X10q**.

More important, the X10q is the first 40Gb/s packet processor. It can perform forwarding and filtering functions on 100 million packets per second while running at a core clock frequency of only 200MHz. If that's overkill, a slightly slower 160MHz version of the chip can handle 60 million packets per second, enough for four 10Gb Ethernet channels. Although the X10q's applications are rather narrow—mainly, layer 2–4 packet processing—at least it has a sharply defined market, unlike some other extreme processors. (See *MPR 8/18/03-01*, "Xelerated's Xtraordinary NPU.")

If a 1,040-stage pipeline seems impractical, remember that the X10q isn't running general-purpose software that branches to a new address every four or five instructions. The chip's 200 identical processor cores are chained together in series between the four 10Gb/s input ports and the four 10Gb/s output ports. Distributed throughout this pipeline are 11 "engine access points"—essentially, intersections that allow the processors to access the other on-chip function units as well as off-chip memories and coprocessors. Each processor core has 44 bytes of packet memory; larger packets are divided into fragments. Therefore, even the smallest packet will occupy at least two stages of the pipeline. Figure 5 shows a high-level block diagram of the Xelerator X10q.

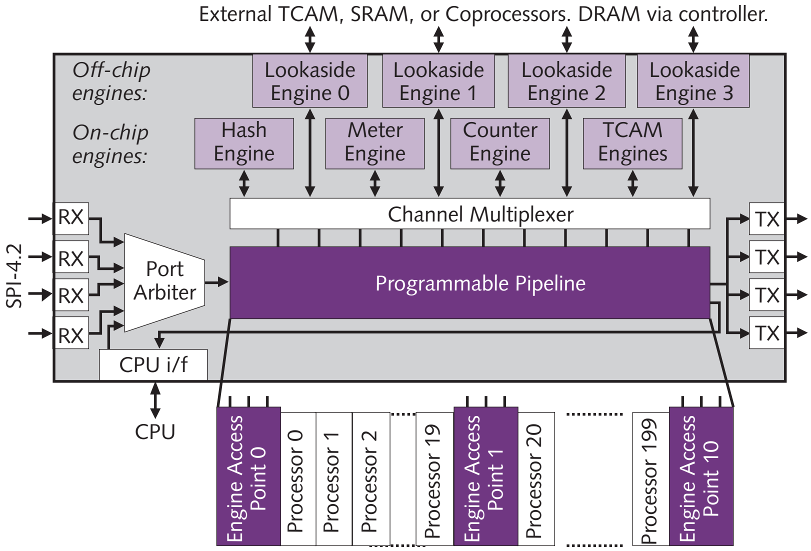

Figure 5. The Xelerator X10q is a single-chip NPU with 200 packet-processor cores and a superpipeline more than 1,000 stages long. At 200MHz, the X10q can process up to 100 million packets per second on a 40Gb/s network.

The 200 identical processor cores in the X10q have a VLIW instruction set that encodes as many as four operations per instruction word. Each processor has 64 words of local instruction memory and can execute one instruction per packet. Under best-case conditions, the X10q can execute 80 billion operations per second at 200MHz. Despite its extraordinary performance, the 200MHz part typically consumes only 9.5W when fabricated in TSMC's 0.13-micron process.

Unlike some massively parallel processors, Xelerated's massively pipelined chip shouldn't be massively difficult to program. Although the microarchitecture inspires shock and awe, the instruction-set architecture is relatively straightforward. Ignore the pipeline and think of the X10q as a processor that provides 200 instruction slots (containing as many as 800 operations) for each packet. Programmers can write code as if the X10q were a single-core, single-threaded processor. Xelerated provides compilers, simulators, and code libraries for common packet-processing algorithms.

With most NPU vendors concentrating on 2.4Gb/s and 10Gb/s packet processors, Xelerated faces little direct competition in 2004. The downside is that few customers need 40Gb/s performance right now. Even so, Xelerated will be ready when the communications market comes back to life, and the scalable nature of the X10q's architecture bodes well for future implementations. ◇