

LEXRA'S NETVORTEX DOES NETWORKING

MIPS-Like CPU Architecture Is Designed for Packet Routing

By Tom R. Halfhill {7/17/00-01}

If you're not satisfied with any of the network processors (NPUs) that everyone from C-Port and IBM to Intel and Sitara has announced in recent months, Lexra has an alternative: license NetVortex and build your own. NetVortex is the first licensable microprocessor

architecture designed for packet processing. Lexra disclosed the first technical details of NetVortex last month at **Embedded Processor Forum**.

NetVortex (formerly code-named Monadnock) is also Lexra's first move beyond the traditional embedded realm, although last year's LX5280 core with DSP-type Radiax instructions indicated the company wants to explore new territory (see [MPR 8/23/99-05](#), "Lexra Adds DSP Extensions"). By deciding to pursue network processing, Lexra appears to be joining a madding crowd of other companies stampeding into the same market. Some companies have even changed their names from "XYZ Semiconductor" to "XYZ Communications"—a letterhead upgrade that seems calculated to attract attention from an investment community soured on dot-coms.

But Lexra's unique spin is that NetVortex is a do-it-yourself solution for ASIC designers who want to wrap special peripherals, coprocessors, and interfaces around one or more processor cores to create custom NPUs. Because NetVortex allows designers to integrate from 1 to 16 cores on a die, it's suitable for a wide range of applications—everything from home-network gateways at the low end to OC-192 core routers at the high end.

Direct competitors that license 32- or 64-bit embedded-processor cores include MIPS Technologies, ARC Cores, and Tensilica. However, their general-purpose cores lack the special features Lexra has added to NetVortex. ARC and Tensilica offer configurable cores that allow designers to duplicate some

or all of NetVortex's features, but those features aren't part of the base configurations, and the proprietary ARC and Tensilica architectures don't enjoy the broad tool support of Lexra's MIPS-like cores.

Available in Multiple Flavors

The NetVortex LX8000 packet-processor core is based on Lexra's LX4189, an R3000-class embedded-processor core with a single six-stage pipeline (see [MPR 5/8/00-06](#), "Lexra Introduces LX4189 Core"). Both the LX8000 and the LX4189 support the popular 32-bit MIPS-I architecture, except for the patented unaligned load/store instructions that are part of an ongoing legal battle between Lexra and MIPS Technologies (see [MPR 12/6/99-03](#), "MIPS vs. Lexra: Definitely Not Aligned").

The basic LX8000 will be available as a soft core and as an optimized hard macro, which Lexra calls a SmoothCore. (SmoothCores save porting time because they come with physical databases for major foundries, and the licensing fee includes porting costs.) Lexra plans to deliver the soft-core version of the LX8000 this quarter and expects processors synthesized from that model to run at 250MHz (worst case) in a typical 0.15-micron IC process, such as TSMC's.

In 4Q00, Lexra plans to deliver RTL for the complete NetVortex architecture, which allows designers to make processors with multiple LX8000 cores, shared memory, and an optional block-transfer controller. The SmoothCore version of NetVortex is scheduled to ship in 1Q01, and Lexra

expects processors based on that macro to reach 450MHz (worst case) in a 0.15-micron process. For some applications, such as access concentrators, Lexra suggests a clock rate of 427MHz, which is an even multiple of the fundamental SONET frequency (19.44MHz). Even the hard macro is fully static and will run at slower clock speeds.

By itself, an LX8000 core would occupy about 2.5mm² of die area when generated from the synthesizable model. Lexra estimates that a full-blown 16-core NPU with 256K of on-chip instruction memory and 256K of on-chip data memory would occupy 64mm² and consume only 6.8W (worst case). Such a processor could route packets on SONET optical backbones at OC-192 data rates (about 10Gb/s), with plenty of headroom left over for other tasks.

To reach that level of performance, NetVortex has several improvements over the standard MIPS architecture. These improvements include new instructions, additional registers, faster buses, and a method for switching contexts among multiple threads of execution with zero-cycle delays. The instantaneous context switching is the most important technique, because it allows a single NetVortex core to process as many as eight packets simultaneously, even though the core has only one execution pipeline.

Multithreading Prevents Stalls

Multiple register files in the NetVortex architecture make rapid multithreading possible. Normally, an operating system

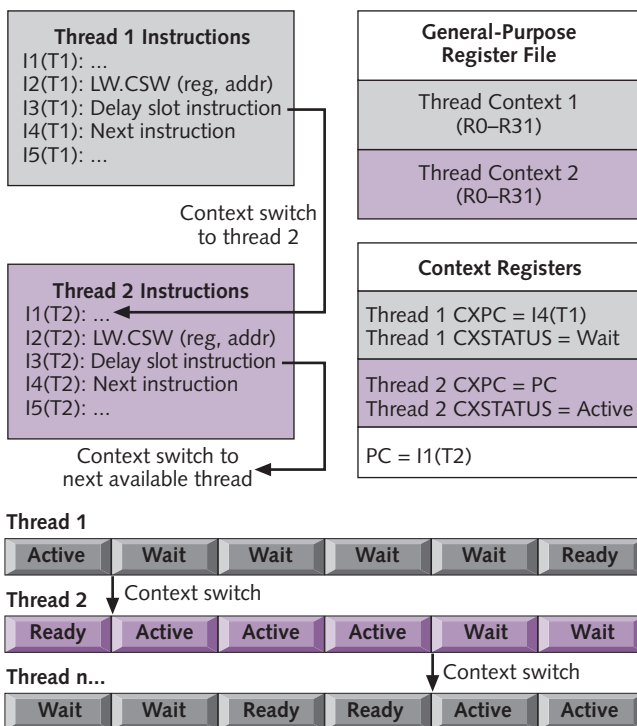


Figure 1. The secret to fast context switching is multiple register files. Each NetVortex LX8000 core can have up to eight independent register files and switch among them with a zero-cycle effective delay.

that supports multithreading or multitasking must save the state of each thread or task during a context switch by copying the contents of general-purpose registers (GPRs) and status registers to memory. The OS must then copy the state of the new thread or task from memory to the registers before the processor can resume execution. To avoid all those slow memory accesses, a NetVortex core can have as many as eight independent register files, each one dedicated to a different thread. Each register file has a complete set of MIPS-like GPRs (32 registers, 32 bits wide), status registers, and a program counter.

Figure 1 shows how NetVortex makes the switch. The second instruction in thread 1 is LW.CSW (load word with context switch), which might have to wait several cycles for the processor to fetch the data from memory. Like the MIPS architecture, NetVortex provides a delay slot after a branch or memory reference that allows one more instruction to execute while the processor is fetching data. (If the compiler or programmer cannot fill the delay slot with a nondependent instruction, the slot must contain a NOP. The delay slot may not contain a branch, jump, or load.) Without multithreading, the processor would stall after the delay slot until the data arrives. Instead, NetVortex switches to the next available thread—in this example, thread 2—and initiates the load, all in the same clock cycle.

Because each thread has its own register file, there's no need for the OS to save and restore any state during the context switch. NetVortex simply flips to thread 2's registers. In addition to the GPRs, each context has a 16-bit status register (CXSTATUS) that indicates whether it's actively executing, is inactive and waiting for data, or is inactive but ready to resume. Each context also has a 32-bit program counter (CXPC) that points to the address of the next instruction in its thread. In effect, the CXPC becomes the processor's global program counter while that context is active.

When thread 2 encounters a context-switching instruction, the whole procedure repeats itself—thread 2 becomes inactive and the next available thread becomes active. The CPU doesn't skip a beat. Context switches don't force the CPU to flush its pipeline, as conditional branches can, because the CPU discovers a context-switching instruction early enough in the pipeline to avoid creating bubbles. Unlike conditional branches, which the CPU must resolve in the execute stage, all context-switching instructions are unconditional. There is nothing to resolve and they always execute. The CPU discovers a context-switching instruction in the decode stage and always executes the following instruction (in the delay slot) as well. The next instruction the CPU fetches is from the newly active thread. Only if no other thread is ready to resume execution does the pipeline stall.

Global memory accesses by inactive threads don't stall active threads, because a separate bus (described below) handles data transfers, and NetVortex can fetch data from local memory in parallel with one outstanding global access per thread. The local memory includes 16K of instruction

memory and 16K of dual-ported data memory per LX8000 core. These software-managed memories are substitutes for primary caches, because the indeterministic behavior of ordinary caches would cause misses that could interfere with wire-speed packet processing.

Instant context switching hides load-use delays and other stalls caused by access to shared resources. It also allows a single, uniscalar NetVortex core to juggle as many as eight packets without the delays of conventional context switching or the massive overhead of control logic that an eight-issue superscalar design would impose. In comparison, the number of latch transistors required for the extra register files is miniscule—each additional context adds only about 0.35mm² of silicon to a single-context LX8000 core. (If a soft-core implementation needs fewer than eight contexts, the synthesis tools will generate only the number of register files required. The hard core has the full complement of eight register files.)

Multithreading is ideal for routing because packets are self-contained units that don't have mutual data dependencies. They can also be processed out of order, although it's highly desirable to send them along their way in the same order they were received. NetVortex preserves order either by tagging the packets with an identifier before processing them or by handling the packets in a round-robin fashion, although the latter method requires twice as much packet-buffering memory.

New Instructions Switch Contexts

To make the multithreading technique work, Lexra added new instructions that explicitly switch contexts during operations that would normally stall the pipeline. At the same time, Lexra added some new bit-field instructions that make it easier to analyze packet headers, which often contain fields that aren't word-aligned or even byte-aligned. As Table 1 shows, some of the context-switching instructions are unusually complex for a RISC architecture.

To use the new instructions, programmers must write the most important sections of a program in assembly language, although they can use a higher-level language for the surrounding code. This doesn't necessarily put NetVortex at a disadvantage against other NPU architectures. Programmers usually write the most critical packet-processing code in assembly language anyway, and some NPUs—such as Intel's IXP1200—aren't supported by any high-level programming tools at all.

Note that some new instructions (such as LT.CSW) manipulate 64-bit "twinword" operands. To boost performance, Lexra doubled the width of the standard 32-bit

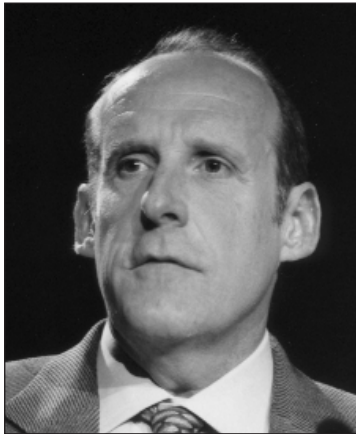
system bus found on other Lexra cores, stretching it to 64 bits in the LX8000. NetVortex processors can load and store twinwords in a single bus cycle. The new bus also handles split transactions, tagging each transaction with a core/context identifier.

Some additional instructions (such as WD) write descriptors to devices on the system bus. Descriptors are control values that manage peripherals, such as the NetVortex block-transfer controller (BTC). The optional BTC transfers data between the local data memory on each LX8000 core and external devices. The BTC has 16 transmit ports and 16 receive ports, each 8 bits wide. Under software control, the ports can be combined into 16-, 32-, or 64-bit-wide ports.

Block transfers between the BTC and the cores travel on a 64-bit internal bus called the VortexBus. A single die may have up to four of these buses, and each bus can run at a 1:1 or 1:2 ratio to the core frequency. That yields up to 14.4GB/s of peak bandwidth in a 450MHz NetVortex SmoothCore.

Figure 2 shows what a NetVortex NPU designed for an OC-192 router might look like. (Lexra has no plans to make standard parts, so NetVortex implementations are up to licensees.) This example has 12 LX8000

cores (only two, highlighted in purple, are shown in the figure), plus another CPU core to handle control functions. The hypothetical chip also integrates several peripherals:



MICHAEL MUSTACCHI

W. Patrick Hays, CTO of Lexra, describes the NetVortex network-processor architecture at Embedded Processor Forum.

| Instruction | Description |
|--|---|
| Context-Control Instructions | |
| MYCX | Read my context |
| POSTCX | Post event to a context |
| CSW | Context switch |
| LW.CSW | Load word with context switch |
| LT.CSW | Load twinword* with context switch |
| WD | Write descriptor † to device |
| WD.CSW | Write descriptor to device with context switch |
| WDLW.CSW | Write descriptor to device, load word with context switch |
| WDLT.CSW | Write descriptor to device, load twinword with context switch |
| Bit-Field Instructions | |
| SETI | Set subfield to ones |
| CLRI | Clear subfield to zeroes |
| EXTIV | Extract subfield and prepare for insertion |
| INSV | Insert extracted subfield |
| ACS2 | Dual 16-bit ones complement add for checksum |
| Cross-Context Access Instructions | |
| MFCXG | Move from a context general-purpose register |
| MTCXG | Move to a context general-purpose register |
| MFCXC | Move from a context-control register |
| MTCXC | Move to a context-control register |

Table 1. The NetVortex architecture extends the MIPS-I instruction set with 18 new instructions, including 6 instructions that switch among different contexts (threads). *Twinwords are 64-bit values. †Descriptors are commands sent to devices.

a BTC with four 64-bit VortexBus ports, a 64-bit PCI interface, a 133MHz DDR-SDRAM controller, a content-addressable-memory (CAM) interface for router-table lookups, and a pair of interfaces to the OC-192 switch fabric.

A Highly Scalable Architecture

For maximum performance, NPU designers could integrate as many as 16 LX8000 cores on a single die, each supporting as many as eight contexts. The number of contexts per core and the number of cores per chip depend on the system requirements, which are not exclusively determined by raw wire speeds.

Basic routing requires fewer instructions to process a packet than more sophisticated services. For instance, a core router that simply glances at the outer-layer headers and forwards the packets to the next hop toward their destinations could get by with less processing power than a more intelligent edge router that looks deeper into the packets for quality-of-service filtering. Therefore, the number of NetVortex cores needed per chip depends on the number of instructions needed to process each packet as well as the wire speed of the switching fabric.

Normally, a NetVortex NPU allocates one thread per packet instead of sharing packet processing among multiple threads. To avoid dropping packets at high wire speeds, it's necessary to distribute the processing of a packet among multiple cores—in effect, multiple-instruction, multiple-data (MIMD) processing.

Table 2 shows Lexra's estimates of the number of cycles required for some common packet-processing tasks. By using a similar table, NPU designers can calculate the number of nanoseconds required to process a packet (based

on the LX8000's clock frequency) and the number of cores required to match a given wire speed without dropping packets. The summary section of the table shows how much capacity the common packet-processing tasks would use on the 12-core NetVortex processor illustrated in Figure 2.

Note that the hypothetical 12-core NetVortex processor doesn't come close to running out of bandwidth when carrying out the tasks in Table 2, even at OC-192 speeds. The processor would typically use only 25–30% of its capacity for those tasks. But the extra bandwidth isn't necessarily wasted. It leaves plenty of headroom for more sophisticated packet processing, such as quality-of-service routing that assigns a higher priority to preferred traffic.

The additional analysis required to filter packets for priority routing (is the packet carrying an important financial transaction or a teenager's MP3 download?) would increase the number of instructions needed to process a packet. An NPU optimized for basic routing wouldn't have enough capacity left over for deeper packet filtering, so a little overdesign is wise protection against early obsolescence.

The Allure of Network Processing

With NetVortex, Lexra may have chanced upon the best business model for network processing seen to date. Instead of competing head-to-head with numerous other NPU vendors, Lexra merely licenses NetVortex to other companies that bear the risk of designing and marketing the NPUs. Lexra gets paid whether its customers succeed or fail—although a successful product would obviously earn more royalties for Lexra (\$1 to \$2.50 per core, depending on volume).

The difficulty of designing a hit NPU probably explains

Lexra's unusually hefty license fees for NetVortex: \$695,000 for the soft core and \$995,000 for the hard core. Those compare with \$350,000 for the LX4189 soft core and \$179,000 for the LX4180 soft core. Perhaps Lexra isn't counting on NetVortex to generate a huge royalty stream. Of course, even a successful NPU won't be as popular as, say, a CPU for a Nintendo machine, so Lexra needs to make more of its revenue up front.

Staying out of the thickest part of the fray is a sound strategy for Lexra. The lineup of NPU vendors is formidable: IBM (see *MPR 10/6/99-en*, "IBM, C-Port Network Processors Challenge Intel"), Intel (see *MPR 9/13/99-01*, "Intel Network Processor Targets Routers"), Motorola (see *MPR 3/6/00-03*, "Motorola Buys C-Port: Smart Move"), and Sitera (see *MPR 5/29/00-02*, "Sitera Samples Its First NPU"), to name just a few.

SiByte announced a powerful MIPS-compatible NPU core at **Embedded Processor Forum** on the same day Lexra unveiled NetVortex,

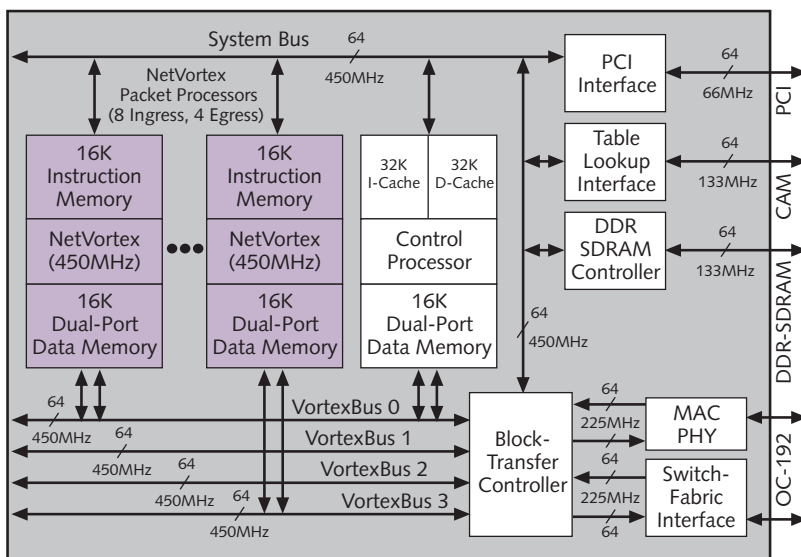


Figure 2. This example of a NetVortex-based NPU for OC-192 routing integrates 12 LX8000 packet-processor cores with an LX4189 control processor, a block-transfer controller, four VortexBuses, and peripheral interfaces.

| Typical Packet-Processing Task | Cycles |
|--|--------|
| Ingress Packet Processing | |
| Get next 64-byte packet from MAC/PHY | 11 |
| Check frame, packet-header syntax, frame checksum | 20 |
| Lookup and tag packet with new MPLS* label, switch output port and priority queue, drop priority | 12 |
| Forward tagged packet to switch-fabric ingress queue manager | 7 |
| Egress Packet Processing | |
| Get next 64-byte packet from switch-fabric egress queue manager | 11 |
| Decrement TTL value, recompute IP header checksum, write new MAC address | 11 |
| Forward packet to MAC/PHY for transmission | 7 |
| Summary | |
| Ingress packet-processor loading | 24% |
| Egress packet-processor loading | 27% |
| System bus loading | 30% |
| Transfer bus loading | 29% |
| Table-lookup unit loading | 30% |

Table 2. Basic routing tasks would consume less than a third of the capacity of the 12-core NPU shown in Figure 2, leaving plenty of headroom for more complex packet processing. Note that ingress processing requires more cycles than egress processing, which is why the chip in Figure 2 has an asymmetrical configuration of eight ingress cores and four egress cores. *MPLS = multiprotocol label switch, which some routers use to forward packets toward the next hop.

but SiByte is designing its own family of NPUs and has no plans to license the core to anybody else (see [MPR 6/26/00-04](#), “SiByte Reveals 64-Bit Core for NPUs”). Microprocessors based on other MIPS cores are commonly found in routers and other networking equipment, but those cores don't have the special packet-processing features that distinguish NetVortex. That's why SiByte resorted to creating its own core after licensing the MIPS64 instruction-set architecture. In fact, MIPS licensees such as SiByte would like to see MIPS add new instructions similar to those in NetVortex.

ARC and Tensilica come the closest to competing against Lexra in this market. Both companies license configurable CPU cores that customers can modify for a wide variety of applications, including network processing. ARC has several design wins in the field, including Chameleon (see [MPR 6/12/00-01](#), “Chameleon Crosses CPU, FPGA”), Cisco (which has patented at least two of its extensions to the ARC core), Hyperchip (which has integrated more than

Price & Availability

Lexra plans to deliver the synthesizable model of the basic LX8000 core this quarter. RTL for the complete NetVortex architecture is scheduled for delivery in 4Q00. The SmoothCore (hard macro) is scheduled to ship in 1Q01. A single-project license costs \$695,000 for the soft core and \$995,000 for the hard core. Royalties start at \$2.50 per core and decline to \$1, based on volume.

a dozen ARC cores on an NPU), and Pixelfusion (which has created a 76-million-transistor chip with 24Mb of embedded DRAM). Meanwhile, Tensilica has licensed its Xtensa core (see [MPR 6/19/00-02](#), “Vector DSP, FPU Extend Xtensa”) to Fujitsu, NEC, and TranSwitch for undisclosed communications chips, and to Zilog for that company's Cartezian family of chips for routers and switches.

However, neither ARC nor Tensilica offers a core that's as ready-made for packet processing as NetVortex. ARC and Tensilica leave it up to their customers to add the new instructions, registers, buses, memories, and bus-transfer controllers that set NetVortex apart from the crowd. ARC's cores are more flexible in this regard because customers can define new condition codes and complex instructions—at least, they can if they're adept at Verilog or VHDL—but the base definition of an ARC core doesn't match the features that are standard equipment with NetVortex. Perhaps that situation will change if ARC or a third-party provider creates an NPU extension (see [MPR 6/19/00-03](#), “ARC Cores Encourages ‘Plug-Ins’”).

Furthermore, the ARC and Tensilica architectures are proprietary and have only a few development tools. Lexra's MIPS-like architecture is widely understood and supported by many third-party tools. EPI is developing an in-circuit emulator for NetVortex—something that even Intel's IXP1200 doesn't have.

So for now, Lexra offers the best solution for network-processor wannabes who want a quicker path to market than designing their own cores from scratch. Lexra has already signed up one undisclosed customer. If the NPU stampede continues, more will surely follow. ♦

To subscribe to Microprocessor Report, phone 408.328.3900 or visit www.MDRonline.com