Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

TENSILICA TACKLES BOTTLENECKS

New Xtensa LX Configurable Processor Shatters Industry Benchmarks

By Tom R. Halfhill {6/1/04-01}

We have reached a watershed. With the RISC vs. CISC war a distant memory, and the sunset of some CPU architectures for workstations and servers, the most exciting architectural innovations are now appearing in embedded processors. This trend is even more amazing for a class of processors formerly derided as hand-me-down desktop designs or vestiges of old technology.

Among the architectural advances appearing first or most extensively in embedded processors are massively parallel processor arrays, on-chip interconnect fabrics, reconfigurable logic, DSP extensions, extendable instruction sets, and hardware-assisted simultaneous multitasking. One reason for this innovation is the great variety of embedded applications, which encourages creativity and experimentation. Another reason is the absence of a large installed software base for any particular embedded processor—unlike the desktop/server markets, which are dominated by monopoly operating systems and entrenched CPU architectures.

Tensilica Pushes the Envelope

Rarely has the ascendance of embedded processors been more evident than at the recent **Embedded Processor Forum**, where several companies announced products and features that must seem like tantalizing fantasies to the architects of staid PC processors. One company in particular, Tensilica, is continuing to pursue a farsighted corporate vision of architectural flexibility and automated design.

At **EPF 2004**, Tensilica announced new versions of its configurable microprocessor core and optional DSP engine, which are licensed as soft intellectual property (IP). When combined with the company's previously announced VLIW-like instruction extensions and next-generation development tools, they will redefine the possibilities for embedded processors.

The new Xtensa LX is a major upgrade of Tensilica's existing configurable processor core, the Xtensa V. (See *MPR 9/16/02-01*, "Tensilica Xtensa V Hits 350MHz.") Xtensa LX tackles three challenges vexing today's CPU architects: the architectural limitations on compute efficiency, the bottlenecks on I/O bandwidth, and rising power consumption. For SoC developers, Xtensa LX preserves the advantages of a customizable CPU architecture while laying the groundwork for future development tools that will further automate the task of creating an optimized SoC design.

Tensilica also announced at EPF a new configurable DSP engine called Vectra LX. Designed specifically for the Xtensa LX processor—Tensilica already offers a DSP engine for earlier Xtensa cores—Vectra LX uses 64-bit instruction words containing three issue slots for ALU, multiply-accumulate, and load/store operations. In all, Vectra LX supports about 200 instructions for 16-bit fixed-point signal processing. Vectra LX is included with Xtensa LX and adds a level of DSP performance unprecedented in a synthesizable RISC processor. (*MPR* will cover Vectra LX in detail in a future article.)

All this probably seems too good to be true. However, Tensilica can back up its claims with independently certified benchmark results. Xtensa LX clobbers every other benchmarked processor in its class—and even some processors out of its class. For instance, in the EEMBC consumer suite, Xtensa LX achieved the highest out-of-the-box ConsumerMark score ever recorded by a licensable CPU core: 171.6 when simulated at 330MHz. That's more than three times

higher than the previous out-of-the-box champ, the Philips TriMedia TM5250, which scored a ConsumerMark of 51.3 when simulated at 500MHz. (See *MPR 11/3/03-01*, "Philips Powers Up for Video.")

Tensilica also submitted Xtensa LX to Berkeley Design Technology Inc. for DSP benchmarking. Result: an optimized Xtensa LX core and Vectra LX DSP engine, simulated at 370MHz, easily outran every other licensable DSP or CPU core ever tested by BDTI. Xtensa LX scored a BDTIsimMark2000 of 6,150—about 70% higher than the previous champ, the CEVA-X1620 DSP, which was simulated at 450MHz. (See the sidebar, "How Tensilica Busted the Benchmarks.")

To achieve these extraordinary benchmark results with a small RISC processor, Tensilica has introduced some groundbreaking new technology and development tools. We believe it's only a matter of time before Tensilica's approach to configurability and design automation exerts more influence over the whole industry.

New Tools and FLIX Distinguish Xtensa LX

Tensilica's primary solution for boosting compute performance is well known: customers can create their own application-specific extensions to Xtensa's base instruction-set architecture (ISA). This approach—pioneered by ARC International and imitated by MIPS Technologies—has been validated by benchmark testing and proved by numerous designs in the field. (Tensilica currently has 160 design wins at more than 60 companies.)

For now, customers must manually write their extensions in Tensilica Instruction Extension (TIE) language. TIE is a proprietary hardware-design language (HDL) that works with the company's back-end tools to automatically generate register-transfer-level (RTL) Verilog or VHDL. It's a correct-by-construction tool chain that prevents the kinds of errors that often creep into extensions written in other HDLs. SoC developers can combine the resulting synthesizable model of the customized processor core with additional application-specific logic and on-chip peripherals, then port the design to any foundry or fabrication process.

Later this year, Tensilica will introduce next-generation processor-development tools that can automatically generate TIE extensions from application software written in C/C++. These remarkable tools, announced at EPF 2003, can analyze a customer's application code, rapidly generate hundreds or even thousands of possible extension instructions, and guide customers toward choosing the extensions that are the best match for the project's requirements. (See *MPR 6/23/03-01*, "Tensilica's Software Makes Hardware.") Xtensa LX is the first processor core designed to work with these tools, although Tensilica hasn't ruled out the possibility of making them compatible with Xtensa V. (The yet-unnamed tools will be an extra-cost option, beyond the usual tools that Tensilica provides; more details will be released in July.)

Xtensa LX is also the first processor core that can use Tensilica's Flexible-Length Instruction Xtensions (FLIX). Announced at Microprocessor Forum 2002, FLIX is a unique VLIW-like option for the Xtensa ISA. Customers can create special 32- or 64-bit instruction words containing multiple subinstructions. The subinstructions can be 1–28 bits or 1–60 bits long, depending on the length of the FLIX instruction word. (Four bits in each instruction word are reserved.) The simplest subinstructions are 1-bit commands that invoke operations in application-specific logic. Others can be sophisticated fused-ALU operations, compound load-compute operations, or single-instruction, multiple-data (SIMD) operations. Programs can freely mix FLIX instruction words with the standard 16- and 24-bit RISC instructions in the Xtensa ISA and with 24-bit extension instructions written in TIE. (See *MPR 11/25/02-06*, "FLIX: The New Xtensa ISA Mix.") In addition, Tensilica's next-generation processor-development tools can automatically generate application-specific FLIX instructions by analyzing C/C++ software code.

No other microprocessor architecture of any type—embedded, PC, workstation, server, DSP—can match the instruction-set flexibility of Xtensa LX with FLIX. Nor can any other company match Tensilica's end-to-end chain of processor-development tools.

ARC has impressive configurable-processor technology but lacks the same degree of tool-chain automation. Silicon Hive—a Philips-funded startup based in the Netherlands—has automated tools for generating synthesizable processors, but the tools lack a graphical user interface and aren't intended for use by customers. Furthermore, Silicon Hive's processor architecture is a massively parallel design for specialized applications, not a general-purpose RISC architecture. (See *MPR 12/1/03-02*, "Silicon Hive Breaks Out.") With the possible exceptions of Hewlett-Packard's secretive PICO and Lx projects, no other company is seriously in the race.

Xtensa LX is the highest realization both of Tensilica's strategic vision and of esoteric academic research into the potential of software-driven automated hardware design. This technology alone would make Xtensa LX a great leap beyond Xtensa V, but Tensilica hasn't stopped there.

Processing Power Strains I/O Bandwidth

As the EEMBC and BDTI benchmark results indicate, custom extensions can make all the difference for an otherwise ordinary 32-bit RISC processor. However, Tensilica realized that I/O bottlenecks and slow memory—problems familiar to all CPU architects—were limiting the effectiveness of custom extensions. The amount of processing power available to Xtensa developers was steadily increasing, but the core's existing I/O interfaces weren't keeping up with the larger amounts of data moving between the core, custom logic blocks, coprocessors, and memory (both on and off chip).

To break the I/O bottlenecks, Xtensa LX offers customers two new options: adding a second load/store unit and defining a virtually unlimited number of I/O ports to off-core

resources. In addition, to reduce the mismatch between the faster processor core and slower on-chip memory, Xtensa LX allows customers to lengthen the default instruction pipeline by two stages. It's up to customers to decide which options make sense for their SoC designs. All three options will be available in the next revision of Tensilica's Processor Generator, a graphical configuration tool that automatically creates the RTL code required to synthesize an Xtensa LX processor.

The second load/store unit, like the default load/store unit, connects to the Xtensa Processor Interface (PIF), the main I/O bus. The PIF can function as a local SoC bus or connect to a different SoC bus. Both load/store units can access all the processor's registers and caches plus the Xtensa Local Memory Interface (XLMI), which supports single- or multicycle access to on-chip memory, coprocessors, and custom logic. The I/O interfaces on both load/store units are configurable, up to 128 bits wide—unusually wide for a general-purpose embedded processor. If the processor has one load/store unit with a 128-bit interface, the maximum theoretical bandwidth at 350MHz is 5.2GB/s. Adding a second load/store unit doubles the bandwidth to 10.4GB/s. Figure 1 shows how the second load/store unit fits into the processor's bus architecture.

One limitation of the second load/store unit is that existing load/store instructions in the base instruction set don't recognize it. Customers must define special load/store instructions in TIE language. The special load/store instructions can be 24-bit extension instructions or FLIX instructions. However, this limitation can be turned into an advantage. With two load/store units, an Xtensa LX processor can perform two memory writebacks per cycle, as long as the writes don't conflict by trying to access the same memory ports. The longer instruction words available in FLIX format allow developers to define instructions that carry out multiple compute and memory operations in parallel. It's even possible to imitate the X and Y data-memory operations of a DSP.

Although Xtensa LX isn't a superscalar processor in the classic sense of having duplicate instruction pipelines, it can aspire to the same throughput as a superscalar machine. TIE language allows developers to create multiple pipelined function units that work independently of the main instruction pipeline. These function units can save results in their registers concurrently with other function units in the same clock cycle. With an optional second load/store unit, Xtensa LX can perform two memory writebacks per cycle as well, essentially matching the capabilities of a two-way superscalar processor.

Off-Core I/O Ports Limited Only by Routing

Another new option for expanding on-chip I/O bandwidth is to define special ports connecting Xtensa LX function units with off-core logic blocks and/or coprocessors, which may be additional Xtensa LX processor cores. After customers write a high-level definition of the ports in TIE language, Tensilica's processor-development tools automatically generate the preverified RTL required to implement the ports.

TIE ports connect directly to extension registers in the processor core and can transfer data at the core clock frequency. (Xtensa LX core frequencies will vary according to the design, of course, but Tensilica says a typical design can hit 350MHz under worst-case conditions in a 0.13-micron CMOS process.) Extension registers can be 1 to 1,024 bits wide—another configuration option—so the width of a TIE port can likewise range from 1 to 1,024 "wires."

Customers can define up to 1,024 TIE ports for an Xtensa LX processor. Therefore, the theoretical maximum number of I/O lines for TIE ports on one processor core is $1{,}024^2$, which would provide 350,000Gb/s of I/O bandwidth at a core frequency of 350MHz.

Obviously, the limitations of today's place-and-route tools (not to mention fabrication processes) rule out a design with a million I/O lines. However, TIE ports provide enough capacity and flexibility to match the I/O requirements of virtually any off-core resources. Ambitious customers can even create designs that weave dozens or hundreds of processors together in a sophisticated interconnect fabric. (Current designs by Tensilica's customers use an average of six Xtensa processor cores per chip, and one design has more than 150 processors.)

Note that TIE ports connect off-core logic blocks or coprocessors directly to TIE extension registers and other custom-defined state in the processor. This is crucial, because it means a TIE port looks like register porting to anything attached to the ends of the port. When an instruction stores a value in an extension register, that value can be immediately available to the attached logic block or coprocessor. Likewise, when a logic block or coprocessor writes to a TIE port (perhaps through an internal register of its own), the data appears in the connected function unit's extension register.

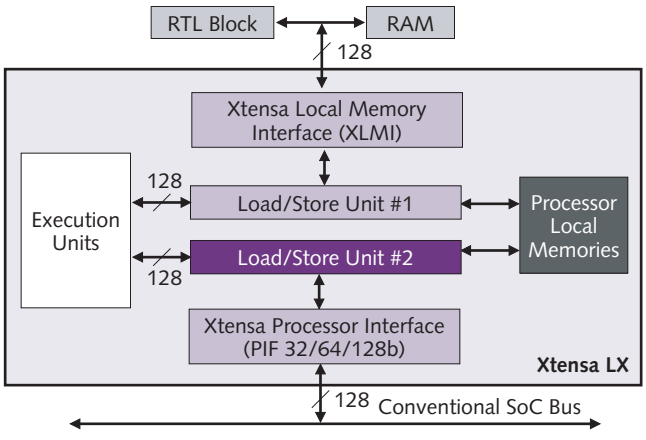

Figure 1. With an optional second load/store unit, Xtensa LX can provide twice as much I/O bandwidth as its predecessor, the Xtensa V. Both load/store units connect to the processor's Xtensa Processor Interface (PIF) and can optionally connect to the low-latency Xtensa Local Memory Interface (XLMI). All the interfaces are configurable, up to 128 bits wide.

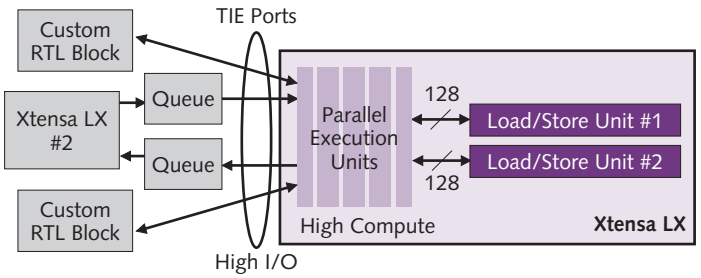

Figure 2. Configurable TIE ports offer a new option for attaching application-specific logic, coprocessors, and additional Xtensa processors to an Xtensa LX core. Theoretically, SoC developers can define as many as 1,024 TIE ports, each with as many as 1,024 I/O lines. Some ports can have queues to buffer time-of-flight or operation-latency delays. TIE ports appear as registers to the function units and off-core logic they connect together.

Therefore, moving data across TIE ports doesn't require special data-transfer instructions or even conventional load/store instructions. Data movement is automatic with any instruction that stores results in an extension register. An instruction can load an operand, perform a calculation, and save the result—all in a single clock cycle, making the result instantly available to any logic connected to the associated TIE port.

Sometimes, however, results can't be "instantly" available over a TIE port, due to wire delays or the latency of an operation at either end of the port. In those cases, customers can use TIE language to define a special queue for the port. Tensilica's processor-development tools can automatically generate the RTL for a queue with two-wire hardware handshaking. Each queue couples to a FIFO buffer that can transfer a block of data equal to the width of the TIE port (1–1,024 bits) per clock cycle. The head and tail of the queue still appear as registers to the attached function unit and off-core logic. Figure 2 shows how ports and queues can link together an Xtensa LX processor with other on-chip resources.

There's nothing quite like TIE ports in competing processors. Of course, it's always possible to design custom I/O interfaces in Verilog or VHDL and attach them to a synthesizable processor core. But the SoC developer is responsible for solving the synchronization, handshaking, buffering, latency, and other problems associated with a custom I/O interface. Tensilica's innovation is that developers can define a sophisticated I/O interface at a relatively high level in TIE language, then let Tensilica's back-end tools generate the down-and-dirty RTL. This feature of Xtensa LX could shave months off a development project.

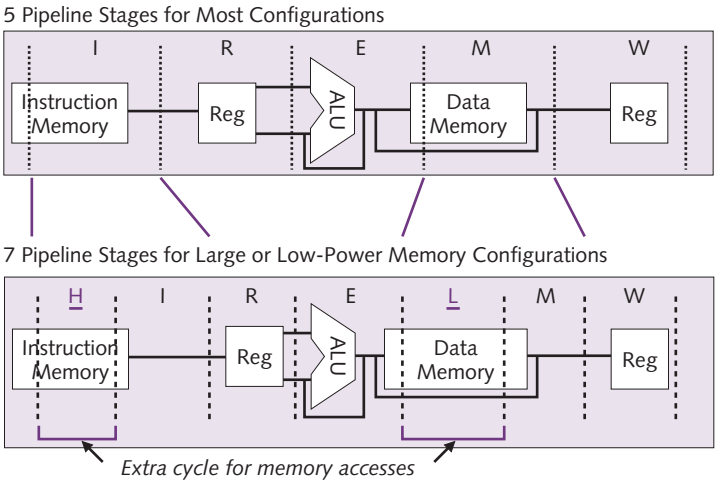

Figure 3. Another new feature in Xtensa LX is possibly the world's first point-and-click configurable instruction pipeline. By selecting options in Tensilica's Processor Generator tool, SoC developers can add a second instruction-fetch stage and a second data-memory stage to the default five-stage pipeline. The two extra stages prevent slower on-chip memories from stalling the processor.

Pipeline Is Configurable, Too

Tensilica's third new option in Xtensa LX is a configurable instruction pipeline—yet another unique feature. Although it's possible to modify the pipelines of other synthesizable processors, doing so would require manually modifying the Verilog or VHDL model. The customer would be responsible for resolving all issues related to pipeline synchronization, data forwarding, gate-delay distribution, and so forth. With Xtensa LX, modifying the pipeline is as easy as selecting a new configuration option in Tensilica's graphical Processor Generator.

The default Xtensa LX pipeline has five stages: instruction fetch, register access, execute, data-memory access, and register writeback. This is a canonical pipeline for a simple RISC processor, and it's sufficient for most purposes. But then, most RISC processors don't have configurable ISAs. To keep custom-defined multicycle instructions from stalling the main pipeline, all recent versions of TIE language allow developers to create separate decoupled pipelines up to 32 stages long. However, that doesn't solve the problem of slow local memories stalling the main pipeline during routine memory-access operations.

Mismatches between the pipeline and local memories might happen for different reasons. Low-power designs might use slower memory to save silicon and reduce power consumption. Conversely, high-performance designs might have large on-chip memories that cannot deliver single-cycle latency at high clock speeds.

As Figure 3 shows, Xtensa LX users can compensate for those mismatches by extending the default five-stage pipeline to seven stages, adding an extra instruction-fetch stage and an extra data-access stage. The extra stages can have one or two cycles of latency, so they can effectively double or triple the memory-access time at high clock frequencies.

The configurable pipeline is another impressive feature but falls a little short of making CPU architects obsolete. Although the Processor Generator automatically resolves the pipeline synchronization and data-forwarding issues, it doesn't redistribute the gate delays to balance

How Tensilica Busted the Benchmarks

This isn't the first time Tensilica has revealed eye-popping benchmark results. In 2001, the company startled the embedded world with certified EEMBC scores that put other processors to shame. (See *MPR 4/9/01-01*, "Stretching Silicon to the Max.") The trick was that Tensilica designed extension instructions for the Xtensa processor that dramatically accelerated some important test kernels in the EEMBC benchmark suites.

Tensilica reported its exceptional 2001 results under EEMBC's rules for "full fury" (optimized) benchmarking. At the same time, the company reported the rather ordinary results for a base configuration of the Xtensa processor under EEMBC's rules for "out-of-the-box" (unoptimized) benchmarking. It was like showing before-and-after pictures of a fat person in a diet advertisement. Tensilica wanted to show how custom extensions could make a huge difference in performance—in this case, a whopping 37× increase in the TeleMark score.

With the latest EEMBC scores, Tensilica is taking a different and perhaps controversial approach. Again, the company uses custom extensions written in TIE language to speed up some test kernels in the EEMBC benchmark suites. Again, the Xtensa scores blow the doors off every other benchmarked processor. But this time, Tensilica is reporting the scores under EEMBC's rules for *out-of-the-box* benchmarking, even though Xtensa LX was specifically optimized for the benchmark tests with custom TIE instructions.

Is this cheating? Not according to Tensilica, and not according to EEMBC, whose independent EEMBC Certification Lab verified the results. The reason: Tensilica has improved its Xtensa C/C++ Compiler (XCC) so it can automatically use custom TIE instructions without requiring programmers to modify the application program's source code. Until now, programmers had to use intrinsic functions to call TIE instructions from C. Making any modifications to the benchmark source code would violate EEMBC's rules for out-of-the-box testing. Because the improved XCC compiler can now use TIE instructions without requiring source-code modifications, Tensilica can use an optimized processor core to obtain out-of-the-box scores.

The key point is that EEMBC's rules, which were mostly written five years ago without configurable processors in mind, don't forbid an EEMBC member from benchmarking an optimized processor as an out-of-the-box system. In the EEMBC rulebook, the main difference between unoptimized and optimized benchmarking is that members must compile the test kernels from EEMBC's unmodified C source code for an out-of-the-box test; for a full-fury test, members are allowed to rewrite the source code or even substitute assembly language. The purpose of the bifurcated benchmarks is to reproduce the software optimizations common to embedded programming.

To some observers, it may seem unfair that Tensilica is using an optimized version of Xtensa LX to report out-of-the-box scores. Without the magic of XCC, the optimized Xtensa LX would produce the same out-of-the-box scores as an unoptimized Xtensa LX. Tensilica's argument is that there is no "standard" configuration of an Xtensa processor, because customers always optimize the base configuration for their applications. In this instance, the "application" was the EEMBC consumer benchmark suite. In effect, says Tensilica, any Xtensa configuration that emerges from the Processor Generator can lay claim to being an out-of-the-box configuration.

When Tensilica does modify the EEMBC benchmark code, the company will be able to report conventional optimized scores for Xtensa LX. Those scores should be even higher than the out-of-the-box scores, because they will result from optimized source code running on the optimized processor. Tensilica plans to release the double-optimized scores in July.

Unlike EEMBC, BDTI doesn't make a distinction between optimized and unoptimized benchmarks. BDTI's programmers always port their benchmark code to the target processor, so the scores are similar to EEMBC's full-fury scores. Note that the Xtensa LX core submitted to BDTI was different from the core submitted to EEMBC; it was generated with different configuration options and had 12 custom instructions.

Although Tensilica appears to be bending EEMBC's rules for out-of-the-box benchmarking, *MPR* doesn't consider it cheating. Tensilica is merely taking advantage of the vital difference between a configurable processor and a conventional processor. The same form of "cheating" is available to Tensilica's customers when they optimize Xtensa LX for their applications, so the benchmark scores reflect real-world experience. Moreover, XCC's new ability to use TIE instructions without requiring programmers to change the application code is a big step forward. It means any application program or programmer can automatically benefit from the superior performance of an optimized Xtensa processor.

the pipeline for higher clock frequencies. Usually, deeper pipelines enable faster clock speeds by minimizing the gate delays in any particular stage, but Xtensa LX's maximum core speed in a given fabrication process remains the same, whether the pipeline has five stages or seven. Instead, a deeper pipeline allows Xtensa LX to use slower memory without compromising its nominal clock frequency.

Reducing Power Consumption

Complaints about the high cost of mask sets for deep-submicron fabrication processes are rivaled only by the grumbling about power consumption as SoC designs grow larger and static leakage becomes acute. Here, too, Tensilica has made improvements. Xtensa LX and the revised Processor Generator can automatically implement clock gating for

Price & Availability

The Xtensa LX processor core with Vectra LX DSP engine is available for licensing now and will ship to customers in June. Upfront licensing fees for a single-processor design start at $550,000, including Vectra LX. Royalties are based on chip volumes. The licensing fee includes Tensilica's standard development tools, including the Processor Generator. The Xtensa C/C++ Compiler (XCC), instruction-set simulator, and TIE Compiler are priced separately. The next-generation development tools that automatically generate extensions by analyzing C/C++ application code will be another extra-cost option. For more information, see *www.tensilica.com*.

every functional logic block at the RTL level. All logic in the independently gated clock domains is inactive unless it's actually processing.

Functional logic blocks include not only the standard elements of the Xtensa LX core but also any custom extensions. In fact, every extension instruction can be a separate clock domain. Such aggressive clock gating is more common in hand-optimized hard cores than in synthesizable cores. A typical Xtensa V configuration might have a few dozen clock domains, whereas Xtensa LX might have hundreds. The Xtensa LX configuration for the BDTI benchmarks has 431 domains.

More important, when a customer selects the clock gating option, the Processor Generator implements it automatically. No manual RTL insertions are required. This feature simplifies the design phase of an SoC project and virtually eliminates the need to manually tune the clock circuits after layout.

Thanks to aggressive clock gating, Tensilica estimates that the base configuration of the Xtensa LX core will consume 37% less power than the base Xtensa V processor in the same fabrication process, even though Xtensa LX has 11% more gates. Surprisingly, the 2,000-gate difference between the cores (20,000 vs. 18,000) isn't mainly due to the overhead of the clock gating itself. Instead, it's almost entirely due to the additional control logic required for FLIX. If the FLIX option is disabled, Xtensa LX is virtually the same size as Xtensa V.

Tensilica estimates that the 20,000-gate minimum base configuration of Xtensa LX will consume 0.04mW per megahertz when manufactured in TSMC's 0.13-micron LV process, assuming nominal operating conditions (1.0V, 25°C). A slightly larger configuration with caches, capable of supporting a real-time operating system, will consume about 0.076mW per megahertz under the same conditions.

These power-consumption numbers compare favorably with those from Tensilica's archrival, ARC. A minimum 27,000-gate configuration of the ARC 600 core consumes 0.04mW per megahertz when fabricated in a similar process—the same as Xtensa LX. However, the ARC 600 has a little less performance headroom, topping out at 290MHz. (See *MPR 12/15/03-01*, "ARC Alters Trajectory.") ARC's highest-performance processor is the new ARC 700, which can hit 400MHz in a 0.13-micron process. But the ARC 700 base configuration requires about 100,000 gates and consumes 0.15mW per megahertz, nearly four times as much as Xtensa LX. (See *MPR 3/8/04-01*, "ARC 700 Aims Higher.")

In reality, making these power/performance comparisons between base configurations of configurable processors is like splitting hairs. The strength of a configurable processor is what it can achieve when customized for a specific application. Benchmark results from both ARC and Tensilica show that raw clock speed is much less important for high performance than custom extensions, and that gate counts of custom extensions are a larger factor in the processor's total power consumption.

Note, as well, that while Tensilica's extensive clock gating can dramatically reduce active power, it does nothing to reduce static current leakage. Static leakage is becoming a larger power-consumption factor in smaller fabrication processes. Unfortunately, a soft-IP provider like Tensilica can do little about static leakage, because it's a physics problem inherent in lower-voltage CMOS.

Perhaps someday, Tensilica can offer a solution similar to Transmeta's recently announced Enhanced LongRun, which reduces leakage by varying the threshold voltages of transistors. (See *MPR 2/9/04-19*, "Better, Faster, Cheaper: Take All Three.") But a solution at that level would almost certainly require a process- or foundry-specific synthesis library, which would limit choices for Tensilica's customers.

Xtensa LX Offers Unparalleled Flexibility

The Xtensa LX processor, Vectra LX DSP engine, and automated processor-development tools set a new standard—not just for configurable or embedded processors but for computing. Some computer scientists have long dreamed of application software shaping the hardware instead of rigidly conforming to it. Over the years, software-driven automated hardware design has been the subject of numerous academic papers and experiments. The ultimate expression of the concept is run-time reconfigurable processing. (See *MPR 5/3/04-01*, "Viewpoint: Microprocessor Sunset.") Tensilica doesn't go that far, but no other company has refined and marketed design-time configurability to the same degree.

Credit the strategic vision of Tensilica's founders, a technical management team that has remained largely intact since the company's inception in 1997. Although ARC was the first company to license a configurable processor in 1993, Tensilica was the first company purposely founded to exploit the concept. Doing this gave Tensilica the opportunity to learn from ARC's experience and start with a blank slate.

In addition, Tensilica has focused exclusively on configurable-processor IP, whereas ARC has diluted its resources by licensing peripheral IP, system software, and

unrelated software-development tools in addition to processor IP. (After another management shakeup, ARC recently began divesting itself of ancillary product lines, focusing more tightly on processors, too.) Because of this history, Tensilica's product line is more integrated and automated than ARC's.

Automation is the important difference. Both ARC and Tensilica provide general-purpose 32-bit RISC processors and graphical configuration tools that shield SoC developers from the core's RTL. Both companies' tools are sophisticated and allow developers to configure and extend the cores in hours, often with startling results. But Tensilica's tools are more automated, whereas ARC encourages a more freewheeling attitude.

Developers who want to go beyond the canned options and who are comfortable manipulating RTL will be attracted to ARC, because the processor's Verilog model is part of the package and is open to modification. Tensilica has an aversion to RTL hacking and protects the synthesizable models of its processors, allowing customers to modify the core only using the graphical Processor Generator or by writing extensions in proprietary TIE language. That also means Tensilica accepts responsibility if the Processor Generator churns out an Xtensa configuration that doesn't work, because TIE language is supposed to be correct by construction.

MIPS Technologies, which didn't introduce a configurable processor until last year, has not yet matched the configurability or tool automation of either ARC or Tensilica. However, the MIPS microprocessor architecture is more widely used than is ARC's or Tensilica's. A more distant competitor is Silicon Hive, which prefers to license preconfigured processor cores for vertical applications, reserving its processor-generation tools for in-house use.

If truth be told, all these companies suffer more at the hands of ARM than from competing with each other, even though ARM doesn't offer a configurable processor core and says it never will. ARM is by far the most successful vendor of processor IP, because it gained an early lead and established its architecture as a de facto standard for low-power embedded systems. In contrast, ARC has never been profitable; MIPS is popular but struggling financially; Tensilica is still a private company whose finances are opaque; and Silicon Hive is just getting started.

In the technology race, however, Tensilica's start-to-finish processor-development system sets the company apart from the pack. There are now plenty of benchmark scores and field-proven designs to validate the concept of a configurable processor architecture. The remaining challenges are to make the technology easier to use for customers and financially profitable for the vendor. Until Tensilica goes public, we cannot speak to the latter. But Tensilica has the former challenge well in hand. ◇