Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

TENSILICA PATENTS RAISE EYEBROWS

Legal Protection of Configurable-CPU Technology Could Frustrate Competitors

By Tom R. Halfhill and Rich Belgard {12/9/02-01}

Tensilica has been granted two U.S. patents for its system of automatically generating a custom microprocessor core and compatible software-development tools, and the company has 16 more patent applications pending. If archcompetitor ARC International is granted U.S. patents for dozens of similar applications now pending—in addition to the international patents it already holds—the result could be a legal minefield for any other companies that try to offer configurable-processor technology.

The new Tensilica patents are numbers 6,477,683 (filed February 5, 1999) and 6,477,697 (filed May 28, 1999), both issued November 5, 2002. They describe Tensilica's Xtensa microprocessor core and related tools, which were first announced the same year the patents were filed. (See *MPR 3/8/99-02*, "Tensilica CPU Bends to Designers' Will.") Xtensa is a synthesizable embedded-processor core with an extendable architecture: system-on-chip (SoC) developers can add new instructions, registers, and other features to the base architecture to customize it for specific applications. Because the architecture is so flexible, the software-development tools must also be customizable, enabling the compiler, assembler, debugger, and simulators to recognize and use the extensions.

Tensilica licenses a complete system for achieving this goal. SoC developers can customize Xtensa by choosing predefined options with a graphical tool called the Processor Generator, and they can create entirely new extensions with a proprietary instruction-description language called Tensilica Instruction Extensions (TIE). Tensilica's back-end system generates a register-transfer level (RTL) model of the customized processor in Verilog or VHDL, which customers then convert into a gate-level netlist using a standard synthesis compiler.

The system also generates a batch of tools and utilities for the customized processor, including a C/C++ compiler, linker, assembler, disassembler, debugger, cycle-accurate simulator, synthesis scripts, place-and-route scripts, test benches, and verification diagnostics. Tensilica encourages developers to experiment with multiple iterations of their design until they get a processor that offers the best balance of application performance, power consumption, and die size.

Tensilica has one previous patent (number 6,282,633), issued in August 2001, but it's a fairly narrow patent on the Xtensa architecture. The new patents make more sweeping claims. ARC has at least three international patents and 39 applications pending in various regions, including the U.S., Europe, and Asia. Those applications—some filed provisionally as early as 1998—will probably start issuing in the next 12–18 months.

Because ARC filed its applications before, during, and after Tensilica's applications, the first question is whether ARC is trying to protect the same technology. Beyond that is another question of larger concern for the rest of the industry, especially ARM and MIPS Technologies: Will the Tensilica and ARC patents leave enough room for anyone else to introduce a configurable-processor system in the future?

Some Tensilica Claims Are Broad

Tensilica's '683 patent has a straightforward title: "Automated Processor Generation System for Designing a Configurable Processor and Method for the Same." It has 104 claims, but only two (1 and 104) are independent claims. Independent claims are important because they draw the outlines of what the patent protects. Dependent claims include an independent claim but require more detail by adding other limitations. To be infringed, a dependent claim's related independent claim must also be infringed.

Of the 104 claims in the '683 patent, 102 depend on claim 1: "A system for designing a configurable processor, the system comprising: means for, based on a configuration specification, generating a description of a hardware implementation of the processor; and means for, based on the configuration specification, generating software development tools specific to the hardware implementation."

In patent language, two key terms are "system" and "means for." The "means for" language is commonly interpreted as the specific system described in the patent specification—or its equivalent. This language would seem to narrow the definition of what claim 1 and its 102 dependent claims protect.

However, other language in this claim is broader: "a description of a hardware implementation of the processor" and "software development tools." The dependent claims provide (and require) more detail. For example, claim 13 mentions "a detailed HDL hardware implementation description," which implies a hardware-design language (HDL) such as Verilog or VHDL. Several other claims mention specific software-development tools, such as a compiler, assembler, linker, and debugger.

Claim 104 in the '683 patent claims: "A method of designing a configurable processor, the method comprising: generating a description of a hardware implementation of the processor based on a configuration specification; and generating software development tools specific to the hardware implementation based on the configuration specification."

This appears to be almost identical to claim 1, except for the key term "method," which in patent language is commonly interpreted as "any method." In other words, Tensilica claims protection for any method that achieves substantially the same result. That's an exceptionally broad claim—so broad we doubt it could survive a rigorous challenge.

However, breaking the '683 patent would be a daunting task because of the large number of dependent claims. Knocking out a few of them might be relatively easy, but invalidating all 102 would be difficult. The research phase alone would probably cost at least $20,000 per claim, which raises the prospect of a $2 million effort, even before the lawsuit gets to a judge or jury.

There are different styles of writing patent applications. Some applicants prefer to write a larger number of independent claims, with relatively few dependent claims. Writing an application with a large number of dependent claims is one way to discourage challengers that are unwilling to risk substantial amounts of money.

Narrow Claims Protect TIE

Although aspects of Tensilica's '683 patent appear unusually broad, the company's '697 patent seems too narrow to protect anything but Tensilica's specific technology. It begins with a long-winded title: "Adding Complex Instruction Extensions Defined in a Standardized Language to a Microprocessor Design to Produce a Configurable Definition of a Target Instruction Set, and HDL Description of Circuitry Necessary to Implement the Instruction Set, and Development and Verification Tools for the Instruction Set."

This patent has 20 claims, of which only one is independent: "A system for designing a configurable processor, the system comprising: means for generating a configuration specification having a user-definable portion, the user-definable portion of the configuration specification including a specification of user-defined processor state, and at least one user-defined instruction and a user-defined function associated therewith, the function including at least one of reading from and writing to the user-defined processor state; and means for, based on a configuration specification, generating a description of a hardware implementation of the processor."

Again, the key terms are "system" and "means for," which probably will limit the protection to the Tensilica technology disclosed in the patent specification. In contrast to the '683 patent, this patent has no broad method claim. Indeed, the lone independent claim goes into such detail that it might exclude a remarkably similar system. For example, a system that allows users to add instructions but not registers may not infringe on this patent, because there wouldn't be any "user-defined processor state" for the instruction to read from or write to. Instead, the instruction could use the processor's architectural registers, which are not user-defined.

In addition, it's not obvious what this claim means by a "user-defined function" associated with a user-defined instruction. Does it mean a C/C++ function that calls the custom instruction? If so, then a nearly identical system that requires users to invoke a custom instruction in assembly language, instead of with a high-level-language function, would appear to be safe from infringement.

The specific language and ambiguities in the '697 patent make us believe its protection is relatively narrow in scope, tailored to Tensilica's specific technology and implementation. That was indeed the intention, according to Beatrice Fu, Tensilica's vice president of engineering. Fu says Tensilica wrote these claims not to protect *any* configurable-microprocessor system but to protect Tensilica's own system, which the company believes is unique.

In particular, Tensilica's system depends on its proprietary instruction-description language, TIE. That language sets it apart from ARC's existing system, for example, because ARC developers use industry-standard VHDL or Verilog to write extensions for the ARC microprocessor core.

Exploring the Prior Art

Tensilica says TIE and the role it plays in the automatic generation of a custom processor with compatible development tools is the key factor that differentiates Tensilica's technology from prior art. The differentiation is crucial, because patents can be challenged and invalidated if they attempt to

Earlier Configurable Processors: Close, But No Cigar

Neither ARC nor Tensilica invented the concepts of customizable microprocessors or customizable processors with compatible software-development tools. Many other companies and researchers have conceived similar systems. However, *MPR* has been unable to find previous examples that duplicate Tensilica's whole system of using an instruction-description language to automatically and simultaneously generate a custom processor with user-defined extensions and matching software tools.

Some examples come close. In 1996, almost three years before Tensilica filed its patent applications, Hewlett-Packard Labs published a paper titled "Custom-Fit Processors: Letting Applications Define Architectures" by Josh Fisher, Paolo Faraboschi, and Giuseppe Desoli. Their system "automatically designs realistic VLIW architectures highly optimized for one given application (the input for this system), while running all other code correctly." The system also included a "product-quality compiler that generates very aggressive VLIW code."

In December 1999, HP Labs and STMicroelectronics announced they were converting this "custom-fit processor" technology into a product. Their system, later named Lx, can rapidly generate application-specific VLIW processors with compatible development tools, simulators, and real-time operating system kernels. (See *MPR 1/24/00-03*, "HP and ST Collaborate on VLIW.") HP and ST formed their partnership in 1997, the same year Tensilica was founded. Unfortunately, HP and ST haven't publicly disclosed enough details about Lx to compare it with Tensilica's patented technology. The 1996 paper indicates that one missing element is an instruction-description language, like TIE, that automatically configures the compiler. For now, HP isn't taking a position on the Tensilica patents.

Lx should not be confused with a similar project at HP Labs known as PICO (Program In, Chip Out). PICO is a separate research project that doesn't involve ST and isn't a shipping product, according to Dave Berman, HP Labs media relations manager. PICO's primary inventor was Bob Rau, an HP engineer. Again, not enough is known about PICO to make a detailed comparison with Tensilica's system. In any case, HP didn't file applications for some key patents on the PICO technology (such as 6,385,757, 6,408,428, and 6,457,173) until August 1999, a few months after Tensilica filed.

Other early explorers in this field were JRS Research Labs (founded 1982), Quantitative Technology Corp. (QTC, founded 1985), and some academic researchers, including Dr. Hans Mulder (a former professor at Delft University in the Netherlands who is now at Intel) and Dr. Henk Corporaal (also a professor at Delft).

In 1988, JRS introduced an interesting system called IDAS (Integrated Design Automation System), which once attracted the attention of Lockheed Martin. IDAS could start with application software written in C or Ada and use it as a behavioral model to synthesize an application-specific VLIW processor with a matching C or Ada compiler. However, IDAS had several limitations, which are briefly described in patent 6,226,776, filed in 1997 by Synetry. In addition, it lacked an instruction-description language like TIE.

Unfortunately, JRS seems to be out of business. QTC, which at one time sold a configurable cross-compiler called Software Foundry, appears to have suffered the same fate. *MPR* was unable to locate either company or to find any relevant patents by them, although other patents cite their work.

Mulder's research at Delft University in the late 1980s was also prescient. He worked on a hardware/software codesign system that would feed an application analysis into a metacompiler to generate a customized RISC architecture and compiler. Mulder's team completed part of the system—the architecture framework, called SCARCE—but not the whole system.

Later, Mulder and Corporaal developed the Move architecture, which tried to go a step further. Move was based on a configurable VLIW architecture with only one instruction: move. All operations were side-effects of moving operands into particular registers. After Mulder left Delft for Intel in 1991, Corporaal continued working on Move and even created a few simple processors for commercial embedded applications. However, Mulder says the processor customization was still done manually, not automatically, a significant difference from Tensilica's system. Furthermore, Corporaal and Mulder didn't obtain a patent, although they did publish several papers on their technology from 1989 to 1991.

protect something already patented or publicly known before the application was filed. Virtually all inventions build on existing technology, so a good patent shows how the new invention is different from this prior art.

Patent applicants must cite all the relevant prior art they know about, but they aren't required to search for prior art. That's the responsibility of the government's patent examiner. Some examiners do a more thorough job than others.

Tensilica's patents describe some related technology from companies like ARM, Lexra, MIPS, and Synopsys, as well as a few academic sources. But one obvious example barely rates a mention: ARC. The '683 patent skims over ARC's technology with one paragraph, and the '697 patent doesn't mention ARC's technology at all. This is curious, because ARC was arguably the first company to license a customizable processor core and to file patent applications for the technology.

For More Information

To look up Tensilica's patents in the U.S. Patent and Trademark Office database, go to *patft.uspto.gov/netahtml/srchnum.htm* and enter the patent numbers (6,477,683 and 6,477,697). Tensilica's announcement is at *www.tensilica.com/html/pr_2002_11_11b.html*.

ARC was founded in 1996 as Argonaut RISC Cores after a spinoff from Argonaut Software, a British videogame company. Argonaut created a customizable CPU core in 1993, after customers kept asking for similar versions of Argonaut's synthesizable core. ARC's current flagship products are the customizable ARCtangent-A4 and ARCtangent-A5 processors, both 32-bit RISC cores for embedded applications.

Tensilica was founded in 1997, and two years later it introduced the customizable Xtensa processor, also a 32-bit RISC core for embedded applications. Tensilica's current product is the Xtensa V, announced in August. (See *MPR 9/16/02-01*, "Tensilica Xtensa V Hits 350MHz.") ARC and Tensilica are direct competitors, and they make similar marketing claims.

Tensilica's '683 patent summarizes the customizable features of the ARC processor and dismisses it with one sentence: "The ARC design has no facility for implementing an instruction set description language, nor does it generate software tools specific to the configured processor."

That statement is at least half true. The first part refers to Tensilica's TIE language, and ARC, as mentioned before, currently uses Verilog and VHDL to define instructions. Verilog and VHDL can describe logic at the behavioral or structural levels, whereas TIE is a behavioral language. However, TIE has semantics for explicitly describing the relevant characteristics of a machine instruction, such as the number of operands it may take, the data types of the operands, the registers it may use, the status flags it affects, and so forth. Those explicit semantics are missing from Verilog and VHDL. Tensilica's system uses the explicit semantics in TIE to extract information about user-defined instructions and to automatically modify the software-development tools.

We are aware of no technical obstacles that would prevent ARC from creating an instruction-description language like TIE. However, Tensilica's '683 patent appears to erect a legal obstacle that could block ARC and other companies from introducing such a language in the future—especially if it plays a role in automatically configuring the software tools.

Playing Tug-of-War With Tool Chains

The second part of Tensilica's statement in the '683 patent—which claims that ARC's system doesn't generate software tools matching the customized processor—is more questionable. Indeed, this strikes to the heart of the fierce marketing battle between the companies. Tensilica has always claimed its system is more automatic and complete than ARC's.

Marketing squabbles aside, the issue is whether ARC's system generates custom software tools for a configured processor. Both companies freely bandy the term "generates," which is somewhat misleading, because neither company's system literally generates a compiler, linker, assembler, debugger, or simulator. Rather, they start with base versions of those tools, written by programmers to support the base architectures of their respective microprocessors. During the customization process, both companies' systems modify the tools to match the user's CPU configuration. The degree of automation varies.

For instance, if an ARC user clicks a checkbox in the ARChitect configuration tool to add a barrel shifter to the ARCtangent processor, the Verilog or VHDL model assembled by ARChitect will include the predefined files required to add the barrel shifter and five new instructions to the base architecture. The C/C++ compiler and assembler can recognize and use the new instructions, although programmers sometimes must invoke the instructions by using automatically generated intrinsic functions or macros.

That's generally the way the system works for the instructions ARC calls "standard extensions"—predefined extensions offered in the graphical ARChitect tool. ARC's system requires a little extra work for custom instructions created from scratch, because the possibilities are boundless: developers can create virtually any instruction in Verilog or VHDL. After writing a structural description of the instruction in one of those languages, developers must write brief compiler pragmas and assembler macros to define the instruction's parameters for the software tools. Again, the compiler and assembler encapsulate the new instruction as an intrinsic function or macro.

Tensilica's system is similar, but one significant difference is the role TIE plays. Instead of writing a structural description of a custom instruction in Verilog or VHDL, users write a behavioral description in TIE. Thanks to its explicit instruction semantics, TIE can pass the instruction's characteristics (opcode, number of operands, etc.) to the back-end part of Tensilica's Processor Generator. The Generator translates the behavioral description into RTL Verilog or VHDL and automatically creates intrinsic functions for the compiler. Users don't have to write their own macros or pragmas, eliminating a possible source of manual errors.

One trade-off is that TIE is more restrictive than Verilog or VHDL, although Tensilica has gradually improved the language's versatility. (For instance, the original version of TIE didn't allow multicycle instructions.) In return, this part of Tensilica's system is more automated than ARC's is.

From one point of view, ARC's system is subtractive, not additive. That is, the software-development tools support a superset of the processor's base instruction set plus the standard extensions; if the user doesn't select a particular standard extension, ARChitect omits the HDL files, compiler pragmas, and assembler macros required to support that extension. In this sense, ARC's system more closely resembles conditional compilation than automatic tool generation. Nevertheless, at least for the standard extensions, it can be argued that ARC's system does "generate software tools specific to the configured processor"—contrary to Tensilica's statement in the '683 patent.

However, it's obvious why Tensilica's '683 patent describes tool generation as an inherent part of the customization process: it's a key point of differentiation. ARC's current system doesn't have an instruction-description language like TIE for creating new instructions and telling the software tools how to use them.

From what *MPR* has learned about ARC's international patents and numerous patent applications, none describes a system quite like Tensilica's. For example, one of ARC's key applications is number WO 00/22553, published by the World Intellectual Property Organization on April 20, 2000. The international filing date was October 14, 1999—several months after Tensilica filed its patent applications—but the "priority date" stems back to a U.S. application filed on October 14, 1998, several months before Tensilica's filings. ARC's detailed 110-page application, which includes a step-by-step 42-page flow chart, thoroughly describes the process for generating a custom microprocessor. It doesn't, however, describe a process for generating custom software-development tools to work with the processor. In fact, it hardly mentions software tools at all.

Unless ARC's other pending patent applications have much more to say about automatically configuring software tools, they probably won't conflict with Tensilica's patents, even if ARC filed the applications first.

The Best Defense: No Offense

When Tensilica recently announced its patents, the press release quoted Tensilica president and CEO Chris Rowen as saying, "These patents recognize Tensilica's technology lead and make us the only company legally entitled to deliver these capabilities to the industry." Taken literally, the second part of his statement is a legal definition of the protection afforded by a patent. It could also be interpreted as a veiled threat against the only company claiming to deliver similar capabilities today: ARC.

Nevertheless, Tensilica says it has no plans to assert its patents against ARC or any company that doesn't substantially duplicate Tensilica's technology. Tensilica says ARC's existing system for configuring software tools isn't automatic enough to infringe on the patents. But if ARC or another competitor introduces a system more like Tensilica's, the gloves may come off.

It's the potential new competitors that should worry. If Tensilica's patents protect the automatic method of configuring software tools, and ARC's existing and future patents protect the manual method, what's left for anyone else? Soft-core providers, such as ARM and MIPS, that are edging toward greater configurability might find themselves locked out altogether. It will be easier to assess this situation when patent offices begin issuing the dozens of patents ARC has pending.

Meanwhile, we think Tensilica is steering a wise course. Sometimes, it's better to avoid risking the validity of a patent by starting an action and provoking a challenge. Tensilica's patents are better held in reserve for defense, in case anyone copies Tensilica's specific technology. ◇

(Editor's note: Halfhill was a technical writer/analyst at ARC International from 2000 to 2002 before rejoining In-Stat/MDR in August. Belgard is a patent consultant and a member of the MPR *editorial board. Neither is an attorney, and this article does not constitute legal advice.)*