Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

ARC Patent Looks Formidable

U.S. Patent Covers Automated Tools for Customizing Processor Cores

By Tom R. Halfhill and Rich Belgard {8/29/05-01}

A showdown may be looming between ARC International and archrival Tensilica over who invented the software tools and methods for customizing synthesizable microprocessor cores. Both companies have won important U.S. patents for the technology, and ARC's latest patent appears both broad and strong. Whether or not ARC and Tensilica come to legal blows, their growing patent portfolios should worry other companies working in the expanding field of configurable processors.

ARC won its latest patent, U.S. 6,862,563 (hereinafter referred to as the '563 patent), on March 1. (See the sidebar, "ARC Wins Key U.S. Patent" in *MPR 3/14/05-02*, "ARC's Preconfigured Cores.") In general, *Microprocessor Report* agrees with ARC that the patent lays claim to key technology for automating the configuration of synthesizable processors and other soft intellectual property (IP). However, the complex language and convoluted history of the patent defy easy analysis and interpretation.

After ARC filed the patent application in October 1999—following a provisional application a year earlier—the U.S. Patent and Trademark Office (USPTO) rejected all the claims three times, which normally is a strikeout. ARC rescued the application by arranging an extraordinary meeting between the patent examiner and ARC's attorneys and engineers. After explaining some terminology and making some changes to the claims, ARC convinced the examiner to approve the application, more than six years after the provisional filing date.

The unusual history of the '563 patent doesn't necessarily weaken it. However, it does offer valuable insight into ARC's claims and the limitations imposed by the USPTO. Even within those limits, *MPR* believes the '563 patent is broad enough and strong enough to concern ARC's competitors, such as ARM, MIPS Technologies, Silicon Hive, and Tensilica. Regular readers of *MPR* are aware that new configurable and reconfigurable processors keep appearing all the time. ARC is claiming fundamental rights to some automated configuration technology that other companies may find difficult to avoid or circumvent.

Tensilica has strong patents, too, and one is about to become stronger. In 2002, the USPTO granted Tensilica two U.S. patents for its system of automatically generating a custom processor core and compatible software-development tools. (See *MPR 12/9/02-01*, "Tensilica Patents Raise Eyebrows.") Within months, an anonymous party challenged Tensilica's 6,477,683 patent by asking the USPTO to reexamine the broadest claims and narrow their scope. (See *MPR 6/2/03-03*, "Tensilica Patent Challenged.") After reexamination, the USPTO rejected all 104 claims in Tensilica's '683 patent, ruling that the claims were invalid over the prior art or didn't represent a patentable advance. But Tensilica appealed the ruling. Just a few weeks ago, the USPTO informed Tensilica that all 104 claims will be reallowed.

Moreover, Tensilica took advantage of the reexamination to add 102 *additional* claims, for a total of 206. Although the revised patent isn't official until issued this fall, it's all but certain, and it's a major victory for Tensilica. Surviving a challenge and reexamination will make the '683 patent almost impervious to another challenge on the same grounds. With its patent portfolio restored to health, Tensilica is in a much better position to defend its configurable-processor

technology, should a defense become necessary. Other companies in this field and future arrivals must find their own armor.

A Brief History of Configurable Processors

Understanding ARC's '563 patent is easier after learning the history behind it. ARC, ARM, MIPS, Silicon Hive, Tensilica, and some other companies license IP in the form of synthesizable processor cores to customers developing ASICs and SoCs. ARC and Tensilica are particularly close competitors, because their 32-bit embedded processors are similar and come with proprietary automated tools for customizing the processors before logic-level synthesis. Customizable features include caches, registers, instructions, buses, interrupts, and other aspects of the architecture and microarchitecture. Licensees can readily modify and extend the processor to optimize it for a target application, then synthesize the design and manufacture it themselves or at a foundry.

One important point is that ARC's '563 patent does nothing to protect the architecture or microarchitecture of ARC's processor cores. Instead, it protects the automated tools and methods for configuring and extending the processors. Tensilica's '683 patent is similar in that regard. Although ARC and Tensilica also have patents on their processors, their baseline architectures and microarchitectures don't differ substantially from others. The most valuable IP is embodied in their automated processor-configuration tools, and their tools were born in very different ways.

A video game indirectly led ARC into the configurable-processor business. In 1993, ARC's corporate predecessor Argonaut Software designed a graphics-accelerator chip called Super FX for a Super Nintendo game (StarFox in the U.S. and Japan, StarWing in Europe). Encouraged by the success of Super FX, and recognizing the advantages of software-driven microprocessor design, Argonaut began designing embedded processor cores for other customers. An Argonaut engineer grew weary of repeatedly making changes to the same basic processor to satisfy customer demands, so he reworked the core to make it configurable. This improvement freed up his leisure time and suggested a new business opportunity to his bosses. Argonaut began licensing a configurable-processor core in 1997, and an early licensee was Fujitsu. In 1998, Argonaut spun off ARC (which originally stood for Argonaut RISC Computing) as a separate company.

In contrast to ARC's accidental entry into configurable processing, Tensilica was founded in 1997 solely for the purpose of creating and licensing configurable-processor cores. Former MIPS engineers designed the first configurable Xtensa processor and began licensing it in 1998. From the start, Tensilica created Xtensa and its configuration tool—the Xtensa Processor Generator—as an integrated system.

ARC's tools evolved more gradually. At first, ARC licensed its processor with command-line configuration tools. Customers asked for something easier to use. In response, ARC's first processor-configuration tool with a graphical user interface—an interactive "wizard"—appeared in 1998, a year after the debut of the configurable processor. Unlike Tensilica's system, ARC's first configuration utilities modified only the hardware design of the processor, not the associated software development tools. Later, ARC introduced an improved program called ARChitect, which continues evolving to this day.

The configuration tools from both ARC and Tensilica let customers modify the processor by clicking buttons and selecting predefined options, as Figures 1, 2, and 3 show. Customers can

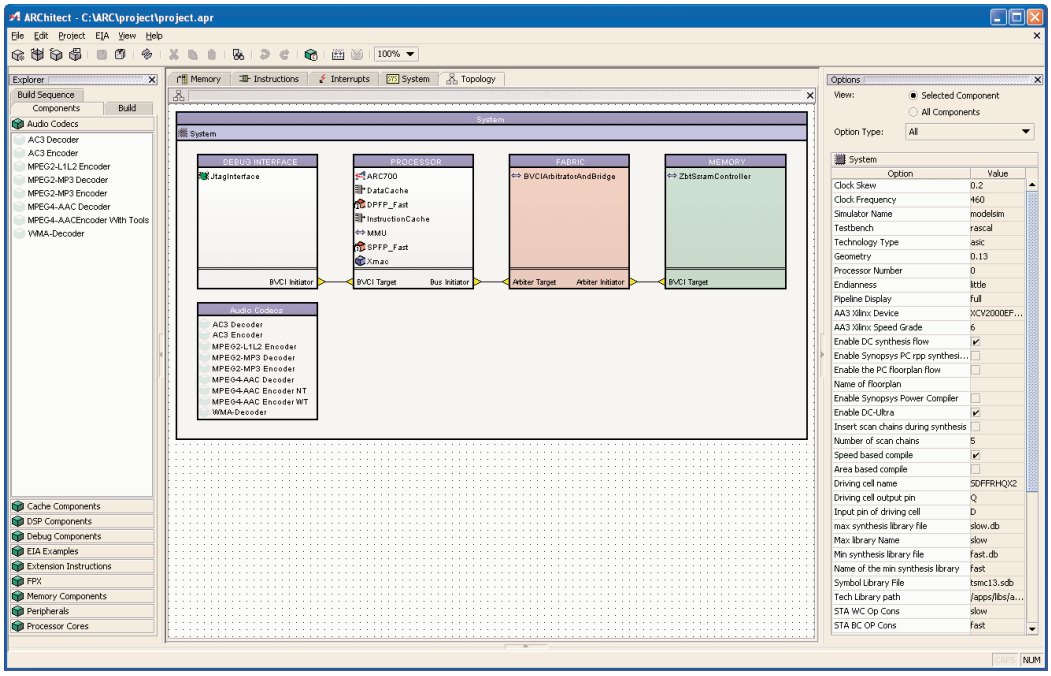

Figure 1. ARC's processor-configuration tool, now called the ARChitect Processor Configurator, runs on a PC or a Sun workstation. It has an easy graphical user interface that allows chip designers to rapidly customize an ARC processor core by choosing predefined options. It generates a synthesizable model of the customized processor by accessing prewritten hardware-description language (HDL) files in a locally stored library.

also write their own extensions from scratch, using a hardware-description language.

Skimpy Examples of Prior Art

Were ARC and Tensilica really the inventors of configurable microprocessors? The USPTO seems to think so. Or, more accurately, the USPTO says ARC and Tensilica invented their respective systems and methods for configuring processors, as described in their patents. In a general sense, microprocessors have always been "configurable," i.e., designers have always been able to modify them for better performance. What ARC and Tensilica have patented are automated configuration tools, and both companies put the tools into the hands of their customers.

One way to challenge a patent is to cite previous examples of essentially identical work, known as prior art. As a general rule, the USPTO assumes the invention claimed in a patent application was invented one year before the application was filed. Although ARC filed the application for the '563 patent on October 14, 1999, a provisional application was filed exactly one year earlier. The provisional filing implies that ARC invented the technology on October 14, 1997 (the "priority date"). Tensilica didn't introduce its configurable-processor technology until 1998 and didn't file any patent applications until 1999.

When *MPR* reported on the patents issued to Tensilica in 2002, we searched extensively for examples of prior art. For many years, corporate and academic researchers have explored ways of optimizing microprocessor architectures and microarchitectures for specific applications. Some of that research has either used or discussed using automated configuration tools. However, *MPR* was unable to find examples of prior art that foreshadowed Tensilica's complete processor-configuration system. (See the sidebar, "Earlier Configurable Processors: Close, But No Cigar," in *MPR 12/9/02-01*, "Tensilica Patents Raise Eyebrows.") ARC cited our report among the references to related technology in the '563 patent.

Tensilica filed the application for its '683 patent ("Automated Processor Generation System for Designing a Configurable Processor and Method for the Same") on February 5, 1999—eight months before ARC filed the '563 application but four months *after* ARC's provisional application. As mentioned above, the USPTO reexamined Tensilica's '683 patent after an anonymous challenge and rejected all 104 claims, but it has now reallowed them. Evidently, the USPTO decided that Tensilica's claims are indeed valid over the prior art and represent a patentable advance.

MPR suspects that ARC challenged the Tensilica patent and initially won by citing prior art claimed in the '563 application. By reversing that decision, and by issuing ARC's '563 patent, the USPTO has effectively established that both ARC and Tensilica invented key technology for configurable processors. Any future challengers will have difficulty attacking those patents on grounds of prior art.

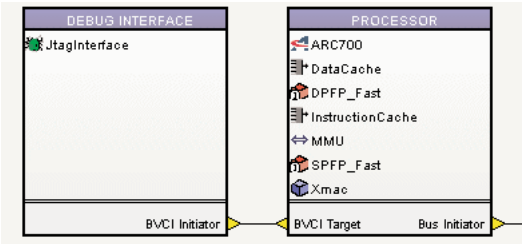

Figure 2. In this excerpt from Figure 1, ARChitect indicates that the user has configured an ARC 700 processor core with instruction/data caches, the fast versions of ARC's single- and double-precision floating-point units (SPFP_Fast and DPFP_Fast), a multiply-accumulate (XMAC) instruction, and a memory-management unit (MMU). There's also a BVCI bus and a JTAG debug interface.

ARC's Claims Are Broad

ARC's '563 patent—"Method and Apparatus for Managing the Configuration and Functionality of a Semiconductor

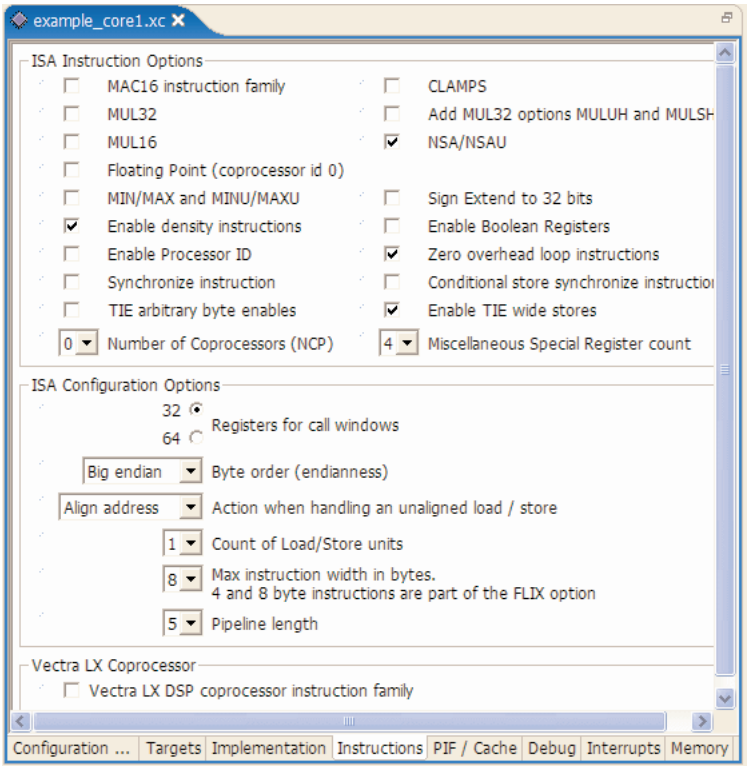

Figure 3. Tensilica's processor-configuration tool underwent major changes last year. The predefined configuration options for the Xtensa LX processor now appear in Tensilica's integrated development environment (IDE), called Xplorer, which runs on the customer's desktop workstation. With earlier Xtensa processors, the configuration options appear on a series of web pages accessible from the workstation. In either case, the workstation sends the user's choices over the Internet to a server program called the Xtensa Processor Generator, which produces the synthesizable model of the customized processor.

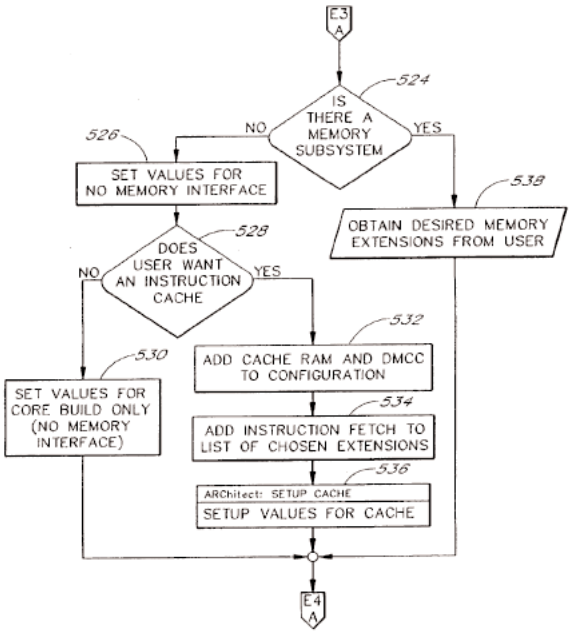

Figure 4. ARC's '563 patent has dozens of flowcharts like this one, showing how a processor-configuration program accepts user input to customize the synthesizable core.

Design"—is an impressive 90 pages long, half of them filled with diagrams, like the one in Figure 4. Even more impressive is the huge 863-page file history, which documents the patent's bureaucratic journey through the USPTO's application process. Most file histories are about 100 pages long. As mentioned before, the USPTO rejected the application three times, and ARC kept resubmitting it until finally winning over the patent examiner. The file history contains all the back-and-forth correspondence and revisions, a boon for anyone analyzing the patent.

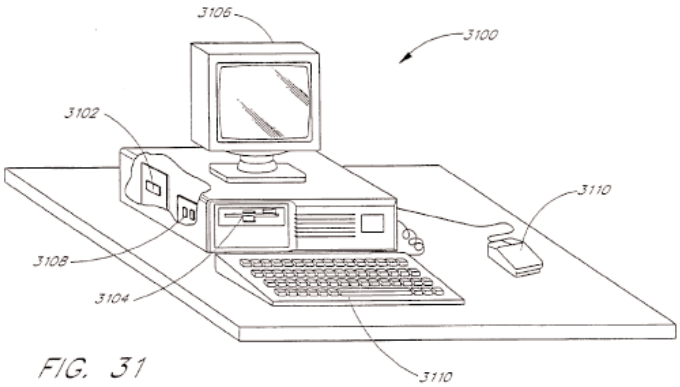

Figure 5. To eliminate any possible doubt about what constitutes a computer, ARC's patent describes the required components and illustrates them with this figure. The point of these claims is to protect the invention of a desktop workstation running processor-configuration software. Even the desktop is included.

U.S. patents can be design patents, plant patents, or utility patents. Most computer-technology patents are utility patents, although some are design patents. ARC's is a utility patent. This type of patent can make apparatus claims (which describe a physical invention) and method claims (which describe steps required to perform an invented process). Either type of claim can be an independent claim or a dependent claim. Independent claims are self-contained descriptions of an invention; dependent claims depend on other claims (which may be independent or dependent) and describe additional elements of an invention.

ARC's utility patent contains both apparatus claims and method claims. In all, it has 54 claims: 17 independents with 37 dependents. We believe it is a cleverly written patent that looks difficult to challenge. In addition, it could give ARC multiple options for pursuing alleged infringers. It's possible that ARC could take action not only against a party offering similar technology but also against that party's customers, or perhaps even the customer's foundry.

The patent's abstract neatly summarizes its claims. It begins, "A method of managing the configuration, design parameters, and functionality of an integrated circuit (IC) design using a hardware-description language (HDL)." Note the reference to ICs instead of microprocessors: this patent also applies to other forms of synthesizable IP, such as peripheral cores. The abstract continues, "Instructions can be added, subtracted, or generated by the designer interactively during the design process, and customized HDL descriptions of the IC design are generated through the use of scripts based on the user-edited instruction set and inputs." This sentence is more specific to microprocessors than to other forms of synthesizable IP lacking an instruction set.

"The customized HDL description can then be used as the basis for generating 'makefiles' for purposes of simulation and/or logic-level synthesis," the abstract continues. "The method further affords the ability to generate an HDL model of a complete device, such as a microprocessor or DSP. A computer program implementing the aforementioned method and a hardware system for running the computer program are also disclosed."

ARC's reference to a computer program obviously describes ARChitect or any similar configuration tool. But the reference to a "hardware system" seems odd at first, because ARChitect is software, not hardware. ARC doesn't make or sell CAD workstations. Instead, ARChitect runs on almost any computer with Microsoft Windows, Red Hat Linux, or Sun Solaris. Further analysis reveals the reason that ARC appears to be patenting CAD workstations for customizing synthesizable processors.

Analyzing ARC's Apparatus Claims

Descriptions of workstation hardware appear in several of the patent's independent and dependent apparatus claims. For example, independent claim 7 describes "an apparatus adapted to generate IC designs," which must include a CPU

capable of running software, a mass-storage device, and an input device—in other words, virtually any computer. As Figure 5 shows, the patent even goes to the length of illustrating a computer.

Independent claims 12 and 16 are similar to claim 7. Claim 12 protects "a system for generating IC designs at a high level of abstraction," which includes a hardware workstation and software like ARChitect. Claim 16 also protects "a system for generating IC designs at a high level of abstraction" and describes the software with additional detail.

In patent language, "system" equals "apparatus," so claims 7, 12, and 16 are apparatus claims. To infringe such a claim, the offender must make, use, or sell the apparatus containing each element of the claim. As claims 7, 12, and 16 are written, we believe it is mainly the chip designer that would have all the required hardware and software elements assembled together for configuring a processor. ARC's competitors don't sell complete CAD workstations, any more than ARC does. At most, they license only the processor-configuration software, which can run on virtually any PC or on a Sun workstation. When customers install the licensed software on their own hardware, all the necessary elements would come together to infringe the apparatus claims, assuming the software performs all the steps the claims describe.

Those steps are fairly simple. Independent claim 7 lists three. First, the chip designer must use the system to input at least one customized parameter (such as a new instruction). Second, the system must access a library file containing a "prototype description" and an "extension logic description." (In general, processor-configuration programs work by integrating the basic HDL model of the processor with prewritten libraries of extension logic, in response to user input.) Third, the system must generate a customized model of the IC by integrating the prototype description (such as the processor) with at least one extension-logic model (such as a new instruction) based on the designer's customized parameter. Almost any automated processor-configuration program will perform these steps in some fashion.

Claims 8 and 9 are dependent on independent claim 7, describing additional elements. Claim 8 says the IC model may be written in a hardware-description language. Claim 9 says the system may be capable of generating and running a simulation of the customized model. (Automatically generated instruction-set simulators are a prominent feature of the processor-configuration tools from ARC and Tensilica.)

Independent apparatus claims 12 and 16 are almost identical to independent apparatus claim 7. Claim 12 describes the same system that claim 7 describes but adds a new phrase and two dependent claims, 13 and 14. The new phrase says the IC design is modeled at a "high level of abstraction"—an important proviso we'll cover later. Dependent claim 13 says the system may offer the designer a choice of fabrication-process options (as do ARChitect and Tensilica's Xtensa Processor Generator). Dependent claim 14 says the system may receive input by reading a preconfigured data file (more on this later). Apparatus claim 16 describes the same system that claim 12 describes, including the phrase "high level of abstraction," but has no dependent claims.

Infringing ARC's Apparatus Claims

Because claims 7, 12, and 16 describe a specific hardware-software apparatus, chip designers who install and use processor-configuration software on their computers are more likely to infringe those claims than is the software vendor or the foundry manufacturing the chips. Of course, it's possible for the vendor to infringe the claims merely by testing the configuration software in house. The claims describe using the system to "generate IC designs," which could simply be an HDL model never fabricated in silicon. However, we think ARC would have more success asserting these claims against chip designers. Pursuing them would exert pressure on the vendor, in any case, especially if the vendor's license provides any legal indemnification to the customer.

If ARC intended these cleverly drafted apparatus claims to broaden the potential infringement base, we believe they slightly miss the mark, and the claims could have been drafted a little more cleverly. For instance, ARC could have drafted them to protect any "computer program product" capable of performing the steps described in the claims. Then ARC could have omitted the generic hardware elements of the claims, retaining only the novel elements. In addition to making the claims shorter—which is generally better in a patent—the claims could cover not only the assembled hardware-software system but also the processor-configuration software by itself. In that case, merely selling the software would infringe the patent—and the software vendor, rather than the customer, might be the infringer.

By describing the workstation hardware in some of its apparatus claims, ARC may have limited its options for asserting the '563 patent against its closest competitor, Tensilica. The reason is that Tensilica's processor-configuration software doesn't run entirely on a customer workstation. The option screens appear on a workstation at the customer's location, offering chip designers numerous choices for configuring Tensilica's Xtensa processor. But the workstation merely sends the user's input over the Internet to the actual Xtensa Processor Generator, which runs on servers at Tensilica's headquarters in Mountain View, California. At that location, the Xtensa Processor Generator builds the customized Verilog or VHDL model, the instruction-set simulator, the C/C++ compiler, and other tools in about an hour. It then sends the whole package over the Internet to the customer. By separating the acts of user input and processor generation, using a client-server model over a network, Tensilica might dodge the elements of ARC's workstation apparatus claims.

However, ARC has other apparatus claims to fall back on. Remember that independent apparatus claim 12 has two dependent claims, 13 and 14. Claim 14 says the system's "input receiving module" may receive the designer's input by reading a "preconfigured data file." This could be a trip wire

for Tensilica. The back-end Xtensa Processor Generator at Tensilica's headquarters receives a data file over the Internet from the front-end user interface running on the customer's workstation. If the back end is an "input receiving module," and the front end creates a "preconfigured data file," then Tensilica's distributed configuration software may satisfy the requirements of the system described in claim 14. The hardware described by independent claim 12 (which claim 14 modifies) doesn't explicitly exclude a networked client-server system.

Then there's independent claim 44, a good example of an apparatus claim that's stronger because it doesn't specify hardware. Indeed, *MPR* believes this claim is the strongest in the '563 patent. It describes an apparatus (which could be hardware or software) that generates a customized HDL model of a processor or peripheral, and it requires only four elements. First, the apparatus must receive user input, including at least one customized parameter for the processor or peripheral. Second, the apparatus must generate a customized HDL model of the processor or peripheral by modifying a prototype design to incorporate the customized parameter. Third, the apparatus must automatically generate test code associated with the customized model. Fourth, the customized HDL model must include "both functional and structural description-language descriptions" for the processor or peripheral. Claim 44 is broad enough to snag almost any automated tool for customizing a synthesizable processor.

If ARC tries to assert the '563 patent against Tensilica on those grounds, Tensilica could move its Xtensa Processor Generator server offshore in an attempt to place the main part of its automated tools beyond the reach of U.S. patent law. However, we view such a move as unlikely. For one thing, other U.S. laws guard against the importation of products that were substantially assembled offshore for the purpose of circumventing U.S. patents. ARC could plausibly argue that those laws apply to the output of Tensilica's tools. More important, the '563 patent appears to give ARC the option of pursuing a U.S. chip foundry or vendor, no matter where the HDL for making the chip originates.

ARC's Strongest Method Claims

In addition to making apparatus claims, ARC's '563 patent makes several method claims, and they appear strong as well. They are tightly written to close loopholes and discourage work-arounds. Note that infringing a method claim is different from infringing an apparatus claim. The offender doesn't have to make, use, or sell a specific apparatus. Merely performing the same steps described by the method claim is sufficient grounds for infringement, no matter what apparatus is employed.

Consider the independent method claims 17, 20, 21, 42, and 43. Claim 17 protects "a computer-implemented method of generating the design of an IC at a high level of abstraction using a description language." Claim 20 protects "a method of generating an extended processor design at a high level of abstraction." Claim 21 protects "a computer-implemented method of generating a customized description-language model of an IC design including at least one of a microprocessor or microprocessor peripheral device." Claim 42 protects "a computer-implemented method of generating a customized description-language model and associated test code of an IC design including at least one of a microprocessor or microprocessor peripheral device." Claim 43 protects "a method of designing an IC design including at least one of a microprocessor or microprocessor peripheral device."

Hear an echo? It's not happenstance that those claims sound alike. ARC is protecting many variations of the technology. All these claims use similar language to describe software tools that let designers configure, modify, or extend the synthesizable model of a microprocessor or other IC.

Claim 17 is representative. It describes a "computer-implemented" (e.g., software-driven) four-step process for generating an IC design. First, the process must start with a high-level model of a processor core. Second, the designer must choose a customized parameter from a list that includes at least one custom instruction, a cache configuration, and a memory-interface configuration. Third, the system must access a library file containing a prototype description and an extension-logic description for the IC model—the same requirement found in the apparatus claims. Fourth, the system must integrate the prototype description, extension description, and customized parameter to generate the final customized model, also as required by the apparatus claims.

The other method claims expand on the concept by describing variations of the same basic process, and some of the independent method claims have dependent claims. For example, method claim 42 describes automatically generating "test code" as part of the process. The test code might be a simulation of the customized processor, or it might be scripts or programs for testing the model on a simulator. The automated tools from ARC and Tensilica generate both types of test code.

ARC's Method Claims Cast a Wide Net

Method claim 42 has no dependent claims, but some other independent method claims do. For instance, claim 21 has five dependents. The independent claim describes the same basic process as claim 17, except it specifies that the IC design is a "microprocessor or microprocessor peripheral device," and it doesn't constrain the list of customized parameters to a custom instruction, a cache configuration, or a memory-interface configuration—any customized parameter will do.

Dependent claims 22–26 add new elements to claim 21. Claim 22 says the system may receive the designer's inputs through an "interactive process," which could describe a graphical user interface (GUI) or practically anything directly engaging the designer. Claim 23 says the system may receive the designer's inputs in a "non-description language format," which could describe the GUI of a processor-configuration tool such as ARChitect. Claim 24 says the

customized model of the processor may include "both functional and structural description-language descriptions," which could refer to high-level HDL models and gate-level netlists.

Claim 25 says the act of generating the customized model may include copying a prototype file, substituting new values, and merging it with additional HDL. That's the way almost all automated processor-configuration tools work. Claim 26 says the description language may be written in steps, which probably means building the customized HDL model gradually, as the designer chooses options in the processor-configuration tool, instead of generating the whole model at the end of the process.

From the preceding examples, it appears ARC has cast a wide net over configurable-processor technology by cleverly drafting apparatus and method claims describing many possible scenarios.

In addition, we believe claim 1 of the '563 patent is a particularly good example of creative claim drafting. It's a method claim written as an apparatus claim, and it describes any IC fabricated by a nine-step process. Some steps are the same as those listed in other claims: receiving user input for at least one customized parameter, accessing a prototype description, accessing an extension-logic description, and modifying the descriptions to incorporate the customized parameter. Other steps in claim 1 are more generic: generating a gate-level netlist for the IC, compiling the netlist, creating a circuit mask or FPGA configuration file, and fabricating the IC using the mask or FPGA configuration file, all at a "high level of abstraction."

Because claim 1 covers fabricating an IC, the potential infringer could be the foundry, the party installing the configuration file in an FPGA, or the company selling the finished chip. However, most configurable-processor licensees are fabless semiconductor companies that employ foundries for manufacturing. Typically (though not always), the fabless companies indemnify the foundries against patent infringement in a design. Although ARC could assert claim 1 against a foundry, the chip designer is more likely to be responsible for damages, because that's who substantially performs the steps to produce the infringing chip.

How the USPTO Limited ARC's Patent

As mentioned before, the USPTO rejected ARC's patent application three times, nearly dooming it to oblivion, before ARC called in the cavalry by arranging an unusual meeting between the patent examiner and the company's attorneys and engineers. To save the application, ARC had to make important changes to the claims. Those changes limit the scope of ARC's patent and use language that could become a point of contention if ARC tries to assert the patent against competitors.

By far the most important change was ARC's addition of the phrase "at least one customized parameter" to all 54 claims. That single alteration was instrumental in convincing the patent examiner to approve the claims, most likely because it distinguishes ARC's inventions from existing design-automation tools. The examiner alluded to the novelty of customized parameters by citing claim 12, an independent apparatus claim describing a system for generating IC designs. Claim 21 uses the phrase "at least one customized parameter" no less than four times. In a document found in the patent's voluminous file history, the examiner wrote, "Independent claim 12...[and] all other pending independent claims are allowable because they contain the same novel limitation, and all pending dependent claims are allowable for the same reason."

In other words, EDA companies whose automated tools generate IC designs by compiling HDL can breathe a little easier. High-level customization, coupled with automation, is ARC's key invention. Although the added language limits the scope of ARC's claims, the limitation isn't severe. Quite a few things could be construed as a customized parameter. And the limitation offers little solace to ARC's competitors, because customization is an inherent feature of configurable processors. Indeed, it's the defining feature that separates configurable processors from other processors.

Another point of contention emerges from the file history of the '563 patent: the phrase "high level of abstraction." Independent claims 1, 12, 15, 16, 17, 18, 19, and 20 contain that phrase, and their dependent claims include it by implication. During the patent application process, there was much confusion over the definition of this term. Initially, it appears the patent examiner thought the term meant HDL or a register-transfer-level (RTL) model—a logical assumption for anyone familiar with design-automation technology. Eventually, ARC explained to the examiner that the term means a level of abstraction *above* HDL or RTL.

What's more abstract than a hardware-description language? One possibility is the GUI of an automated design tool, such as ARChitect. In effect, ARChitect presents a feature-level model of a microprocessor. Other language in the patent also refers to levels of abstraction that are higher than HDL. For example, dependent claims 23, 29, 35, and 40 refer to systems or methods receiving user input in a "non-description language format." Dependent claims 47 and 51 both describe receiving user input from a "graphical user interface (GUI)." These phrases clearly refer to something more abstract than structural HDL—and probably more abstract than functional HDL, as well.

Our interpretation is that ARC is claiming rights to highly abstract views of synthesizable circuit designs that are user-customizable in the ways the patent describes. If so, there is little maneuvering room for ARC's existing competitors—and even less room for other companies entering the field.

Implications for Tensilica

Although we mention Tensilica several times as a potential target of ARC patent litigation, that's only because Tensilica's configurable-processor technology resembles ARC's

For More Information

ARC's latest patent, like all U.S. patents, is available online from the U.S. Patent and Trademark Office (USPTO). Enter the patent number (6,862,563) without commas in the search box at *http://164.195.100.11/netahtml/srchnum.htm*. The Tensilica patent discussed in this article is 6,477,683. Additional Tensilica patents related to configurable-processor technology include number 6,477,697, number 6,701,515, and number 6,760,888.

By clicking on the Advanced button on that web page, you can search for patents using other criteria, such as the name of the inventor or the assignee (which is often the inventor's employer). Patent file histories are not available online, but they are available from the USPTO for a hefty fee.

For company information, visit *www.arc.com* or *www.tensilica.com.*

technology in some ways. Both companies license synthesizable 32-bit processor cores and automated tools that allow developers to customize the design by choosing predefined options and adding new extensions. Both companies offer tools having similar customization options and easy-to-use GUIs. And both companies' licensees end up with customized Verilog or VDHL models ready for logic synthesis, plus matching instruction-set simulators and software-development tools.

One important difference is that Tensilica invented its own proprietary HDL, Tensilica Instruction Extension (TIE) language, for adding custom extensions to the Xtensa processor. ARC favors industry-standard languages, such as Verilog, VHDL, or System C. TIE is crucial to Tensilica's claims in the '683 patent. And so far, TIE has made it easier for Tensilica to offer more tool automation than ARC does. (See *MPR 7/12/04-01*, "Tensilica's Automaton Arrives.")

TIE could factor into a future patent dispute, either to Tensilica's advantage or to its disadvantage. A possible advantage is that TIE introduces major differences that could help insulate Tensilica against ARC's '563 patent. A possible disadvantage is that TIE makes Tensilica more vulnerable to the patent by fitting ARC's definition of a high level of abstraction beyond HDL—recall that the Xtensa Processor Generator converts TIE into Verilog or VHDL. Indeed, as Figure 6 shows, Tensilica promotes TIE as a higher-level HDL that eliminates many complexities of other HDLs.

All these considerations make the restoration of Tensilica's '683 patent even more momentous. That patent protects unique aspects of Tensilica's technology, including TIE. Although TIE provides a higher level of abstraction than do VHDL and Verilog, it's arguably still within the realm of HDLs. By restoring Tensilica's '683 patent, the USPTO seems to agree that TIE is an HDL. Apparently, the USPTO doesn't perceive a conflict between the "high level of abstraction" (above HDL) claimed by ARC and the TIE-related abstractions claimed by Tensilica. Except for minor grammatical changes, Tensilica didn't have to amend any of the 104 challenged claims. Surviving the challenge and reexamination in this way has significantly strengthened Tensilica's '683 patent.

Furthermore, Tensilica has reinforced the patent by adding 102 new claims during reexamination. *MPR* cannot analyze those claims until the USPTO issues the revised patent. However, it's rare for a patent to expand in this fashion. If anything, a reexamination usually results in the USPTO narrowing the scope of the claims or disallowing some claims altogether. That Tensilica managed to save all 104 original claims unscathed while tacking on 102 new ones is remarkable.

ARC and Tensilica now have such strong patent protection for their configurable technology that a legal battle between the companies would probably bleed them for years, which neither company can afford. One company might die, leaving the other to acquire the residual IP. Or they might cross-license their patents to avoid further litigation. In the end, we believe only their common competitors would benefit.

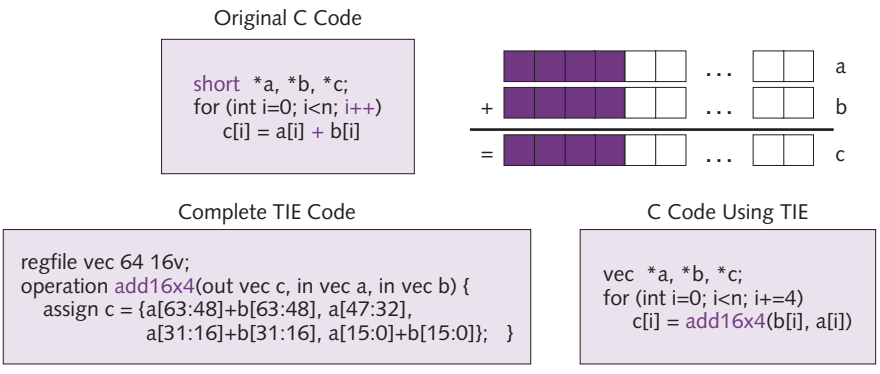

Figure 6. This example of creating a custom instruction in Tensilica Instruction Extension (TIE) language is from Tensilica's website. It explicitly promotes TIE as a higher-level alternative to standard design languages like Verilog and VHDL. In this example, a designer uses TIE to create a new SIMD instruction, and the TIE compiler automatically maps the instruction to an unused opcode and generates new load/store instructions for a new 16-entry register file. The Xtensa Processor Generator automatically generates RTL for this logic and modifies the C/C++ compiler, assembler, instruction-set simulator, debugger, and even an RTOS to support the new instructions and registers. TIE appears to provide a "high level of abstraction," as mentioned in ARC's patent—but it can still be considered an HDL.

Implications for Other Competitors

Other companies have more to worry about. ARC and Tensilica have staked out valuable territory in the field of configurable processors, and both companies have more patent applications pending. Competitors will have to tread

this territory carefully. But tread it they must, because we believe configurable processing is a strategic technology for optimizing performance in embedded systems—too important to ignore in the years to come, especially if Moore's law starts looking like Moore's wall.

MIPS Technologies should be concerned. Its CorExtend MIPS processors and tools are similar to the patented technology from ARC and Tensilica. (See *MPR 3/3/03-01*, "MIPS Embraces Configurable Technology.") ARM's general-purpose processors aren't significantly configurable, but ARM does license a configurable coprocessor core with automated design tools. (See *MPR 6/7/04-01*, "ARM's Configurable OptimoDE.") Silicon Hive is a relative newcomer that takes a slightly different approach to configurable processors, but if ARC and Tensilica get aggressive with their patents, those differences will be tested. (See *MPR 6/20/05-01*, "Busy Bees at Silicon Hive.") Altera's Nios II processor core has a user-extendable instruction set. (See *MPR 6/28/04-02*, "Altera's New CPU for FPGAs.") And that's just a partial list of companies competing against ARC and Tensilica in this field.

In the interest of world peace, *MPR* would rather see no one become a target of these patents. ARC and Tensilica are sufficiently well armed for mutual assured destruction, and we're sure both companies could use some incremental revenue by licensing their patented IP to others. Above all, we would like to see these patents wielded in ways that don't impede the continuing evolution of configurable-processor technology. ◇

*(**Editor's note:** Tom R. Halfhill is a senior analyst for* MPR. *Rich 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.)*