

M I C R O P R O C E S S O R

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

ARM BLESSES FPGAs

New Cortex-M1 Processor Core Is Optimized for FPGA Integration

By Tom R. Halfhill {3/19/07-01}

In a radical departure from past policy, ARM will allow licensees to synthesize some of its embedded-processor cores in FPGAs and is optimizing these cores for programmable-logic fabrics. Until now, with one exception, ARM has permitted licensees to synthesize

ARM processors in FPGAs for development purposes only, not for product deployment.

At the same time, ARM is announcing its first synthesizable processor core specially designed for FPGAs: the Cortex-M1. This small 32-bit core is intended for microcontrollers and deeply embedded applications. ARM says additional FPGA-optimized cores will follow. The first FPGA vendor approved for Cortex-M1 synthesis is Actel. Other vendors—including industry leaders Altera and Xilinx—also have ARM's blessing. ARM is currently tuning the Cortex-M1 for their programmable fabrics and plans to make the RTL available in 2Q07.

Until now, the only synthesizable ARM processor approved for product deployment in FPGAs was a special version of the ARM7TDMI-S licensed to Actel, a relatively small FPGA vendor. In a unique arrangement forged with ARM in 2005, Actel began offering the ARM7TDMI-compatible CoreMP7 soft processor core for integration in some Actel programmable-logic devices. Actel's customers don't need an ARM license and don't pay licensing fees or chip royalties to ARM. Instead, they get the CoreMP7 with a pass-through license from Actel, which includes all costs for the CoreMP7 in the prices of its chips. This streamlined business model is attractive, particularly to startups and small companies that can't afford a conventional ARM license and want to reach the market quickly.

ARM is now expanding this arrangement with Actel to include the new Cortex-M1 processor. ARM has optimized the Cortex-M1 for new Actel Fusion and ProASIC3 devices,

which use flash-based configurable logic. (See *MPR 12/19/05-02*, "Actel Releases First Fusion Chip.") As with the CoreMP7, Actel will offer the Cortex-M1 to customers without requiring an ARM license or chip royalties. Most other FPGA vendors will probably follow a more conventional course by requiring their customers to obtain a Cortex-M1 license directly from ARM. In that case, customers will pay an upfront licensing fee and royalties to ARM. Some FPGA vendors may offer both licensing options. But even the conventional licensing model will be surprisingly affordable. ARM is offering a Cortex-M1 license for less than \$100,000—a very low licensing fee, by ARM standards.

Overall, the big news is that ARM will no longer stop embedded-system developers from deploying synthesizable ARM processors in finished products. *Microprocessor Report* has encouraged ARM to open the market in this way. The rising costs of developing custom chips are exerting great pressure on developers, while the falling unit costs of FPGAs are creating new opportunities. As time goes by, FPGAs make more sense for deployment, not just for prototyping. But ARM's new strategy, though welcome, also has dangers. Perhaps that is why ARM is testing these waters carefully with only one processor, the Cortex-M1. For now, FPGAs are still off limits to other ARM processors.

Why FPGA Integration Is Important

Don't confuse ARM's new strategy with its past practice of allowing Altera to integrate hardened ARM cores in FPGAs.

Some of Altera's older-generation Excalibur devices have an ARM922T hard core, an option missing in Altera's later-generation Stratix devices. (Altera's archrival Xilinx offers some Virtex-II Pro FPGAs with Power 405 hard cores.) Integrating a hard core directly into an FPGA may seem more sensible than synthesizing a soft core in the configurable fabric, because the hard core runs faster and leaves the whole fabric available for other purposes. However, the hard-core option lacks flexibility and hasn't attracted enough customers to justify more products of this type.

Soft processor cores for FPGAs offer several advantages over hard cores. First, developers can customize the processor for specific applications, assuming the core has some degree of configurability. Second, developers can readily modify the design to fix problems, add features, or adapt it for different products. (In some cases, developers can even modify a design already deployed in the field.) And third, FPGA vendors don't need to manufacture multiple versions of their chips, with and without hard cores. ARM's new Cortex-M1 offers all these advantages to developers and FPGA vendors.

ARM has been reluctant to permit soft-core integration in FPGAs for fear of losing control of its valuable intellectual property (IP). ARM's successful business model is based on selling IP licenses and collecting downstream royalties on customers' chips. Likewise, many developers worry about losing control of the application-specific logic they wrap around the processor. In theory, an IP thief could steal the binary image of the design in two ways: by extracting the image from the nonvolatile memory chips that program the FPGA during boot-up; or by intercepting the bitstream between the memory chips and the FPGA. The thief could then study the stolen IP, learn its secrets, and perhaps even reconstruct a synthesizable model for unauthorized reuse. Consequently, ARM has restricted licensees to synthesizing ARM processors in FPGAs only for development, testing, and verification. Product deployment in FPGAs was forbidden—and still is—for all ARM processors but the new Cortex-M1 and Actel's CoreMP7.

However, the semiconductor industry is changing in ways that virtually force ARM to amend its policy. Every new advance in fabrication technology comes at the price of higher development costs for the programmable-core ASICs and SoCs that ARM's customers create. Nonrecurring engineering (NRE) costs are skyrocketing, as are the costs of deep-submicron mask sets. Meanwhile, time-to-market pressures keep building. In some highly competitive markets, users expect new products every 6 to 12 months, whereas developing an ASIC requires 12 to 24 months. To justify developing an expensive custom chip, vendors must sell higher volumes of their products, but the rapid market turnover limits the volumes of any particular model of those products.

Squeezed by these forces, some embedded-system developers are turning to FPGAs. Not many years ago,

FPGAs were too costly for anything but prototyping—or for deployment in the most expensive products. However, the unit prices of FPGAs keep falling, even as the capacities of their configurable fabrics keep growing. There is a point at which deploying an FPGA in a product is more economical than developing and manufacturing an ASIC for that product. Although the crossover point depends on the product's specifications and projected sales volume, each year the point moves inexorably in favor of FPGAs.

Unfortunately for developers, the world's most popular embedded-processor architecture wasn't available for FPGA synthesis—until now. Whether ARM wants to or not, it must adapt to these changes in the semiconductor industry. First with the CoreMP7, and now with the Cortex-M1, it's apparent that ARM is moving strategically but carefully into this new territory.

Actel Offers IP Protection

ARM's concern about stolen IP explains why Actel is the launch pad for this new strategy. First, Actel's lower sales volumes (relative to larger FPGA vendors) will limit the damage if something goes wrong. More important, Actel can protect ARM's valuable IP with strong encryption. Unlike most FPGAs, Actel's devices use integral flash memory, not SRAM, to configure the fabric. At boot-up, Actel's nonvolatile FPGAs don't need to load a binary image of the design from off-chip flash or ROM. Therefore, there's no external binary image to extract or bitstream to intercept. To prevent IP thieves from extracting the design directly from the FPGA, Actel's devices can lock the fabric using Advanced Encryption Standard (AES) cryptography. The synthesized processor in the fabric is unencrypted and runs unimpaired, but external access requires a 128-bit encryption key.

Another safety feature, though most likely a temporary one, is that ARM isn't risking too much IP at first. ARM's initial experiment was to permit Actel to sublicense the CoreMP7, a minor variation of the 12-year-old (though still popular) ARM7TDMI processor. ARM's investment in the ARM7TDMI is well amortized by now. Although the IP certainly isn't expendable, losing control of it wouldn't deal the company a deathblow. ARM's Cortex-M1 is basically an update of the ARM7TDMI. Although the new processor isn't expendable, either, it gives ARM a chance to tweak its strategy before permitting licensees to deploy more-valuable ARM processors in FPGAs.

Eventually, for ARM's strategy to succeed, additional ARM processors must follow the path of the Cortex-M1. Some FPGAs besides those from Actel can protect synthesized IP with encryption, but not all of them can. ARM says it will begin licensing processors for deployment in unencrypted FPGAs in 2Q07, starting with the Cortex-M1. In other words, ARM is willing to accept some risk of IP theft. Developers with application-specific IP to protect will have to make up their own minds in this regard.

Another reason for moving gradually into these waters is that ARM needs time to optimize its processor cores for FPGAs. Although it's possible to synthesize almost any high-level model of a processor for a programmable-logic device, the results are often disappointing. Logic gates implemented in the lookup tables of FPGAs aren't as efficient as the standard-cell logic gates in ASICs. To achieve the best performance—in terms of both throughput and power efficiency—the synthesizable model of a processor must be optimized for configurable logic. So far, the CoreMP7 and Cortex-M1 are the only ARM processors adapted for this purpose.

To complicate things further, programmable-logic fabrics vary from one FPGA vendor to another. Actel's fabrics consist of flash-based "tiles" made of three-input lookup tables (LUT3 gates). Altera and Xilinx commonly use SRAM-based "logic cells" made of LUT4 gates. The new top-of-the-line Xilinx Virtex-5 uses LUT6 gates. These and other differences require ARM to optimize its processor cores differently for each FPGA vendor and perhaps even for specific product lines of FPGAs from a single vendor. To cope with this complexity, ARM has designed the Cortex-M1 in two parts: a high-level generic model of the processor and a lower-level device-specific layer that's interchangeable during logic synthesis. The synthesis script tells the synthesis compiler which lower layer to mate with the higher-level core.

Of course, it takes time to develop all the device-specific layers for the FPGAs that ARM wants to support. That's another reason the Cortex-M1 is available first on Actel FPGAs. ARM and Actel already laid much of the groundwork when porting the ARM7TDMI-S to Actel's devices.

Cortex-M1 Updates the ARM7TDMI

The Cortex-M1 joins the Cortex-M3 as the latest addition to ARM's Cortex-M series, which is designed for integration in microcontrollers and deeply embedded systems. The Cortex-M announcement in 2004 stirred up some controversy because the series isn't fully compatible with other 32-bit ARM processors and software. (See *MPR 11/29/04-01*, "ARM Debuts Logical V7.")

Instead, Cortex-M processors have an instruction set consisting largely or entirely of Thumb and Thumb-2 instructions. They are 32-bit processors—their general-purpose registers, function units, and datapaths are 32 bits wide—but most instructions are only 16 bits long, to achieve greater code density in low-memory systems. Essentially, the Cortex-M1 is an updated ARM7TDMI with a subset of Thumb-2 instructions. The vast majority of Thumb-2 instructions are 16 bits long, but a few instructions required for system-level operations are 32 bits long.

The ARM7TDMI is certainly worth updating. Since its debut in 1995, it has become ARM's biggest-selling processor core, although most new designs favor the ARM9 and ARM11 families. The ARM7TDMI was the first processor to support 16-bit Thumb instructions. (See *MPR 3/27/95-01*, "Thumb Squeezes ARM Code Size.") The synthesizable ARM7TDMI-S

has been available since 1998. But whereas the ARM7TDMI supports the 32-bit ARM instruction set in addition to Thumb, the Cortex-M1 supports only Thumb-2, an improved version of Thumb. (See *MPR 6/17/03-02*, "ARM Grows More Thumbs.")

In ARM nomenclature, the Cortex-M1 supports the ARMv6-M instruction-set architecture (ISA). ARMv6, introduced in 2002, added mixed-endian modes, better exception handling, and more-flexible interrupts, among other things. (See *MPR 11/26/01-03*, "ARM Drives V6 to MP Forum.") As implemented in the Cortex-M1, ARMv6-M is both a subset and superset of ARMv6. Missing are some single-instruction multiple data (SIMD) instructions and other 32-bit operations; added is Thumb-2 and a few 32-bit instructions required for system functions. Table 1 lists the complete Cortex-M1 instruction set.

The ARMv6-M ISA will deter some ARM7TDMI customers from reengineering their designs for the Cortex-M1, because developers will have to rewrite and reverify the portions of their software written for the 32-bit ARM ISA. Programs written entirely in high-level languages should port easily, requiring little more than recompilation. But the kinds of deeply embedded systems that use an ARM7TDMI often have some code written in assembly language, and the 32-bit portions of that code will require rewriting for the Cortex-M1. ARM and Actel minimize this obstacle, contending that most ARM7 code—especially the critical routines—are probably written with Thumb instructions, which the Cortex-M1 executes. But some routines, even in programs that make heavy use of Thumb, contain standard 32-bit ARM instructions. Typical examples are exception handlers, which cannot be written with 16-bit Thumb instructions. The additional 32-bit instructions in the Cortex-M1 allow it to perform such tasks, if developers don't mind porting their code.

ARM7TDMI customers who are reluctant to rewrite their software for the Cortex-M1 but who still want to implement their designs in an FPGA should consider using Actel's CoreMP7 instead. It's fully compatible with the ARM7TDMI. The drawback of that alternative is that the CoreMP7 is available only for Actel FPGAs, whereas the Cortex-M1 will soon be available for other programmable-logic devices as well.

Cortex-M1 Is Mildly Configurable

Another factor that may deter some ARM7TDMI customers from adopting the Cortex-M1 is that the new processor isn't a huge improvement over the 12-year-old one. For instance, the Cortex-M1 has the same three-stage pipeline as the ARM7TDMI; a modern x86 processor needs three stages just to think about decoding an instruction.

Of course, it's by design that the Cortex-M1 isn't festooned with all the architectural advances of the past 12 years. The new core is intended for memory-challenged embedded systems that need a minimal amount of 32-bit processing with high power efficiency. It's not designed to

break benchmark records. At the same time, the Cortex-M1 does offer worthwhile improvements over the ARM7TDMI, such as enhanced interrupts (up to 32, with four priority levels), support for little- or big-endian memory addressing, a choice of two 32-bit integer multipliers (one that's faster but larger, and another that's slower but smaller), and a debug block supporting four breakpoints and two watchpoints (through a JTAG or SWD interface).

Like the ARM7TDMI, the Cortex-M1 is a cacheless processor. However, it offers the option of tightly coupled memories (TCM) for holding critical instructions and data. Each TCM is configurable and can range in size from 1KB to 1MB. Developers usually prefer TCMs over self-managed caches for real-time applications, because their behavior is more deterministic. Note that when the Cortex-M1 is synthesized in a configurable fabric, the TCMs will be implemented in the FPGA's block RAMs, not in conventionally compiled SRAMs.

Developers can choose the Cortex-M1's configuration options at synthesis time. Table 2 shows the options available. However, the initial version of the Cortex-M1 that Actel offers for its Fusion and ProASIC3 devices isn't internally configurable. Instead, Actel will deliver its new Fusion M1AFS600 and ProASIC3 M1A3P1000 chips with an

encrypted "black box" version of the Cortex-M1, preconfigured with little-endian addressing, the small multiplier, the debug block, only one interrupt-priority level, and no TCMs. This base-configuration core will run at clock rates up to 72MHz in the Actel devices and occupy about 4,300 tiles of configurable logic. To put that size in perspective, the Fusion M1AFS600 will have about 14,000 tiles and the ProASIC3 M1A3P1000 about 24,000 tiles.

Configurations and Fabrics Influence Performance

Depending on which configuration options a developer chooses, the size and performance of the Cortex-M1 will vary greatly. Another difficulty of assessing this processor is that its size and performance depend heavily on the configurable-logic fabric for which it is synthesized. As mentioned above, different fabrics use different kinds of logic cells and LUTs, and they are manufactured in different fabrication processes. Comparing fabrics across multiple FPGA vendors isn't straightforward, and relating those gate counts to NAND-equivalent ASIC gates is even dicier.

Consider this example of performance variance. Actel says the Cortex-M1 will run at clock speeds up to 72MHz in its new ProASIC3 and Fusion FPGAs, which are manufactured in a 0.13-micron process. Altera and Xilinx use SRAM-based

Instruction	Width	Description	Instruction	Width	Description
ADC	16b	Add with carry	MUL	16b	Multiply
ADD	16b	Addition	MVN	16b	Move not
ADR	16b	Form address from PC	NEG	16b	Negate
AND	16b	Logical AND	NOP	16b	No operation
ASR	16b	Arithmetic shift right	ORR	16b	Logical OR
B	16b	Branch to label	POP	16b	Pop from stack
BIC	16b	Bit clear	PUSH	16b	Push to stack
BKPT	16b	Software breakpoint	REV	16b	Reverse bytes in word
BL	32b	Branch with link	REV16	16b	Reverse bytes in both halfwords
CMN	16b	Compare negative	REVSH	16b	Reverse bytes in low halfword
CMP	16b	Compare	ROR	16b	Rotate right
CPS	16b	Change processor mode	RSB	16b	Reverse subtract
CPY	16b	Same as MOV	SBC	16b	Subtract with carry
DMB	32b	Data memory barrier	SEV*	16b	Set event (NOP)
DSB	32b	Data sync barrier	STM	16b	Store multiple
EOR	16b	Logical exclusive-OR	STR	16b	Store word
ISB	32b	Instr sync barrier	STRB	16b	Store byte
LDM	16b	Load multiple	STRH	16b	Store halfword
LDR	16b	Load word	SUB	16b	Subtraction
LDRB	16b	Load byte	SVC	16b	Service call to operating system
LDRH	16b	Load halfword	SXTB	16b	Sign-extend byte to word
LDRSB	16b	Load signed byte	SXTH	16b	Sign-extend halfword to word
LDRSH	16b	Load signed halfword	TST	16b	Update CPSR flags
LSL	16b	Logical shift left	UXTB	16b	Unsigned-extend byte to word
LSR	16b	Logical shift right	UXTH	16b	Unsigned-extend halfword to word
MOV	16b	Move data	WFE*	16b	Wait for event (NOP)
MRS	32b	Move PSR to register	WFI*	16b	Wait for interrupt (NOP)
MSR	32b	Move register to PSR	YIELD*	16b	Yield to alternative thread (NOP)

Table 1. Cortex-M1 instruction set. The Cortex-M1 is a 32-bit processor core, but, to save memory, most instructions are only 16 bits long. ARM added some 32-bit instructions that allow the Cortex-M1 to perform system-level tasks in Thumb-2 user code. For example, the MRS and MSR instructions access the processor status registers, allowing exception handlers to run in Thumb mode. *The Cortex-M1 doesn't fully support a few instructions in this table. These instructions won't trigger an illegal opcode exception, but the processor treats them as NOPs.

fabrics that are inherently faster than Actel's flash-based fabrics. ARM says the Cortex-M1 will run at clock frequencies exceeding 174MHz in a Xilinx Virtex-5 device, which is fabricated in a leading-edge 65nm process. That's a difference of 2.4× for similar configurations of the same processor. Naturally, the state-of-the-art Virtex-5 chips will be much more expensive than the Actel chips.

Other performance trade-offs are less intuitive. For instance, Actel says the smaller, "slow" 32-bit multiplier sometimes delivers better overall throughput than the larger, "fast" multiplier, because it allows the synthesized core to run at a higher clock rate that overcomes the "fast" multiplier's advantages. This effect will likely vary from one type of FPGA fabric to another. Yet another factor is that the device-specific layer of the Cortex-M1 may be better optimized for some fabrics than for others.

Fortunately, developers using the Cortex-M1 can rapidly modify and test their designs to measure these differences. It's better than sending an ASIC design off to the fab and nervously waiting a few months—only to receive an unpleasant (and costly) surprise when the design falls short of the performance specifications or fails to work altogether.

Of course, processor performance is a relatively minor factor to consider when implementing a design in an FPGA. Typically, developers surround the processor core with application-specific logic and peripherals, which can have a much greater effect on overall system performance than the processor's raw throughput. To facilitate such connections, the Cortex-M1 has an AMBA-Lite bus. In addition, ARM is adapting some of its AMBA PrimeCell peripherals and macro blocks for programmable-logic fabrics. At this time, ARM's OptimoDE configurable coprocessor isn't licensable for FPGA deployment, but ARM is investigating the possibility. (See *MPR 6/7/04-01*, "ARM's Configurable OptimoDE.")

Competing With Rock-Bottom Prices

Earlier we mentioned some hazards of ARM's new FPGA strategy. IP theft is only one; another is competition. Of course, there's always competition. In this case, however, at least three competitors offer a price advantage too great for ARM to match without endangering its business model. And those three competitors are the very same FPGA vendors allied with ARM's new strategy: Actel, Altera, and Xilinx.

Altera and Xilinx, the FPGA biggies, offer synthesizable 32-bit embedded-processor cores specially designed for their programmable-logic fabrics. Altera has the three-member Nios II family, and Xilinx has the new MicroBlaze v5.0 and still-available MicroBlaze v4.0. In terms of features, throughput, size, and power consumption, these processors are similar to ARM's Cortex-M1. The big difference is price. Altera and Xilinx use their processors as loss leaders to drive sales of their FPGAs. Consequently, their licenses cost only \$495, including software-development tools—and no chip royalties. (See *MPR 11/13/06-01*, "Xilinx Revs Up MicroBlaze," and *MPR 6/28/04-02*, "Altera's New CPU for FPGAs.")

Price & Availability

The ARM Cortex-M1 processor core will be available in 2Q07 for Actel, Altera, and Xilinx FPGAs. Actel's initial version of the Cortex-M1 isn't internally configurable and will be encrypted in new Fusion M1AFS600 and ProASIC3 M1A3P1000 devices, which are scheduled to sample in 3Q07 and ship in 4Q07. Later, Actel will offer a configurable version of the processor for Fusion and ProASIC3 devices, and perhaps for lower-priced Igloo FPGAs as well. All Actel Cortex-M1 FPGAs will include the cost of the processor and don't require customers to obtain a separate ARM license or to pay royalties. Implementing the Cortex-M1 in Altera and Xilinx devices will probably require customers to obtain an ARM license and pay royalties to ARM. However, ARM is offering a relatively low-cost Cortex-M1 license for less than \$100,000 plus royalties. For more information, please visit www.arm.com and www.actel.com.

Actel has entered this field with a relatively low-cost processor, too. Actel has adapted the Gaisler Research LEON3 processor to its FPGAs. LEON3 is a synthesizable 32-bit embedded-processor core based on the Sun Microsystems SPARC V8 architecture. The processor's VHDL source code is freely available for research and education through a GNU Public License (GPL). For commercial use, a LEON3 FPGA license from Gaisler Research costs 20,000 Euros (about \$26,400). Better yet, Actel offers the same deal for the LEON3 that it does for the CoreMP7 and Cortex-M1. Commercial customers implementing LEON3 in Actel FPGAs don't need a commercial license from Gaisler, because the licensing costs are built into the unit price of the chips.

ARM Cortex-M1 Feature	Configuration Options
Tightly Coupled Memory (Instruction)	0K–1024K (1K, 2K, 4K...1024K)
Tightly Coupled Memory (Data)	0K–1024K (1K, 2K, 4K...1024K)
Multiplier (32b Integer)	Fast or small function unit (Default = small)
Vector Interrupt Controller (Nested)	1–32 interrupts (4 priority levels)
Endianness	Big or little (Default = little)
OS Extensions	System timer Software interrupts
Debug Extensions	Removable (4 breakpoints, 2 watchpoints)

Table 2. Cortex-M1 configuration options. Developers can modify the register-transfer-level (RTL) model of the processor to select from these options before logic synthesis. These choices will measurably affect the core's size, throughput, and power consumption.

Historically, ARM has been very secretive about its licensing fees. It is known that a single-project license typically costs hundreds of thousands of dollars plus royalties, while unlimited-use licenses with multicore privileges can cost \$1 million or more. With the Cortex-M1, however, ARM is publicly disclosing a relatively low-price license for less than \$100,000 plus royalties. That license allows developers to use the Cortex-M1 in single-core projects. A multicore license costs more, but the surcharge “is not linear,” ARM says. Although the Cortex-M1 is an astonishing bargain by ARM standards, it’s still significantly more expensive than the LEON3, MicroBlaze, and Nios II cores. Table 3 summarizes the features of these competing cores.

ARM’s Advantages

How can ARM compete with processors that are practically free? *MPR* has been anticipating this collision for years. Yet despite the huge price disparity, ARM has selling points in its favor.

First, ARM virtually owns the industry standard for 32-bit embedded processors. The ARM architecture isn’t quite as dominant as the x86—the embedded market is too diverse for that. Nevertheless, ARM-based processors are found in more than five billion cellphones, every Apple iPod,

and myriad other products and systems. ARM is making impressive inroads into the world’s fastest-growing markets of China and India. ARM is not as strong in some market segments, such as automotives, but it is actively pursuing those opportunities. (See *MPR 10/30/06-01*, “ARM Thumbs a Ride.”) Although thousands of developers (including students and the merely curious) have taken LEON3, Nios II, and MicroBlaze licenses, these processors aren’t nearly as ubiquitous as ARM’s. They don’t have the same thick catalog of development tools, and customers don’t receive the same level of hands-on tech support.

Second, the Altera and Xilinx processors are restricted to synthesis in their vendor’s FPGAs. (LEON3 is agnostic in this regard.) Developers are forbidden to implement a Nios II processor in a Xilinx device or a MicroBlaze processor in an Altera device. Choosing an Altera or Xilinx processor locks the developer into buying that vendor’s chips. In contrast, ARM’s Cortex-M1 will be synthesizable for many different FPGAs. Customers can switch FPGA suppliers without rewriting their software. True, developers will have to resynthesize the core and any associated application-specific logic if they switch. That’s not a trivial task, but at least it’s possible.

Third, if a particular product design outgrows the FPGA—or if the FPGA-based product succeeds so well that its

Feature	ARM Cortex-M1	Altera Nios II/f	Altera Nios II/s	Altera Nios II/e	Gaisler LEON3	Xilinx MicroBlaze v5.0	Xilinx MicroBlaze v4.0
Architecture	ARMv6-M	Nios II	Nios II	Nios II	SPARC V8	MicroBlaze	MicroBlaze
Primary FPGA Targets	Fusion, ProASIC3, Stratix, Virtex-4/5, Cyclone, Spartan	Stratix, Cyclone, HardCopy	Stratix, Cyclone, HardCopy	Stratix, Cyclone, HardCopy	Any	Virtex-5	Virtex-4 Spartan-3E
Configurable ISA	—	Yes	Yes	Yes	—	—	—
Pipeline Depth	3 stages	6 stages	5 stages	1 stage*	7 stages	5 stages	3 stages
I-Cache	—	0–64K	0–64K	—	0–1MB	0–64K	0–64K
D-Cache	—	0–64K	0–64K	—	0–1MB	0–64K	0–64K
Local Memory	0 or 2 1K–1024K each	0–8 Configurable	0–4 Configurable	—	0 or 2 Configurable	0 or 2 256K each	0 or 2 128K each
32-Bit Multiplier	Two options	Optional	Optional	—	Yes	Optional	Optional
32-Bit Divider	—	Optional	Optional	—	Yes	Optional	Optional
Barrel Shifter	Yes	Optional	Optional	—	Yes	Optional	Optional
FPU	—	Optional 32 bits	Optional 32 bits	Optional 32 bits	Optional 32 / 64 bits	Optional 32 bits	Optional 32 bits
Branch Predict	—	Dynamic	Static	—	—	—	—
Privilege Levels	1	2	2	2	2	1	1
Core Freq (Max)	Up to 72MHz [†] >170MHz [‡]	205MHz	165MHz	200MHz	Up to 125MHz in FPGAs	220MHz	205MHz
Int. Perf (Max)	0.8 Dmips / MHz	225 Dmips	127 Dmips	31 Dmips	106 Dmips	240 Dmips	166 Dmips
FP Perf (Max)	n/a	n/a	n/a	n/a	125 MFLOPS	50 MFLOPS	33 MFLOPS
FPGA	4,300+ LUT3 tiles	1,800	1,150	600	~3,500** (Base config)	960–1,700	950–2,400
Logic Cells	(~1,900 LUT4 cells)						
Introduction	4Q07	2004	2004	2004	2004	Oct 2006	May 2005
Price	<\$100,000 (ARM) Free (Actel)	\$495	\$495	\$495	\$26,400 (FPGA) \$46,000 (ASIC)	\$495	\$495

Table 3. The Cortex-M1 faces competition from the same FPGA vendors that are key allies of ARM’s new FPGA strategy—Actel, Altera, and Xilinx. Actel offers the LEON3 processor from Gaisler Research. Altera offers the Nios II family, and Xilinx has its MicroBlaze line. All are similar to ARM’s Cortex-M1 and in some cases offer distinct advantages. For instance, LEON3, Nios II, and MicroBlaze have optional FPUs, caches, and 32-bit hardware dividers. Nios II lets developers create application-specific custom instructions. The biggest advantage of these competing cores is lower-price licenses, unless developers obtain the Cortex-M1 from Actel, which does not require a separate ARM license. *Nios II/e has a six-stage pipeline, but it works like a one-stage pipe. †Estimate for synthesis in an Actel ProASIC3 or Fusion FPGA. ‡Estimate for synthesis in a Xilinx Virtex-5 FPGA. **Estimate for synthesis in an Altera Stratix-II or Xilinx Virtex-4 FPGA. (n/a: not applicable.)

volumes justify developing a custom chip—ARM offers a better migration path. Although ARM licenses the Cortex-M1 only for FPGAs, developers can switch to a higher-end ARM core, such as the Cortex-M3, which is optimized for ASICs and is upward compatible. Software written for the Cortex-M1 will run on any ARM Cortex-family processor, including those supporting the full 32-bit ARM ISA. Although it's possible to synthesize a Nios II or MicroBlaze for an ASIC, those cores aren't designed for that purpose, and neither Altera nor Xilinx offers higher-end processors. Altera's HardCopy initiative provides developers with the option of porting Nios II to a structured ASIC, but that's not quite the same as a custom ASIC. LEON3 is suitable for ASICs but doesn't provide an upgrade path as broad as ARM's.

Fourth, developers can forgo the expense of an ARM license by obtaining the Cortex-M1 directly from Actel. Remember, Actel offers the equivalent of a royalty-free ARM sublicense with the sale of each FPGA, which greatly lowers the barrier of entry. Other FPGA vendors may offer a similar licensing model. Of course, Actel offers the same deal for the LEON3, too.

Other Competitors Must Follow ARM

In a sense, it doesn't matter if ARM's advantages are great enough to outweigh the price disparity between its processors and the almost-free competitors. *MPR* believes ARM is

compelled to move its processors into FPGAs because of the relentless trends in the semiconductor industry noted earlier. The rising costs of spinning custom chips are colliding with the falling unit prices (and growing gate counts) of programmable-logic devices. Every year, the decision whether to deploy designs in FPGAs instead of ASICs tilts a little further in favor of FPGAs. ARM can't stop it. Only a fundamental change in semiconductor technology can alter the balance, and no such change appears imminent.

ARM's other competitors must heed those trends, too. ARC International, MIPS Technologies, and Tensilica all make licensable 32-bit embedded-processor cores intended primarily for synthesis in fixed logic. Those companies don't absolutely forbid customers to deploy their designs in FPGAs, as ARM once did, but they don't encourage it, either. Nor have they optimized their processors for the idiosyncrasies of configurable fabrics.

MPR believes that ARC, MIPS, and Tensilica eventually must follow ARM by introducing FPGA-optimized versions of their cores. They will then meet the same low-price competition that ARM now faces, except without at least one of ARM's advantages—a processor architecture that's nearly ubiquitous. For all processor-IP companies, the onslaught of FPGAs is a thorny business challenge, but it's also a great opportunity. ♦

To subscribe to Microprocessor Report, phone 480.483.4441 or visit www.MPRonline.com