Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

ARM Reveals Cortex-R4

Deeply Embedded Processor Core Inches Toward Configurability

By Tom R. Halfhill {5/16/06-01}

At this week's **Spring Processor Forum** in San Jose, ARM revealed the first member of its Cortex-R family—the Cortex-R4, a synthesizable 32-bit processor core for deeply embedded applications. With this debut, ARM has now introduced initial members of all three of the new Cortex families announced in 2004. The Cortex-R4 is available for licensing now and is already in the hands of lead customers.

ARM's long-term Cortex strategy is to create new families of 32-bit processor cores for three major segments of the embedded market. At the bottom end, the Cortex-M family relies exclusively on Thumb-2 instructions to enable the smallest possible 32-bit cores for microcontrollers; the Cortex-M3 was the first example. (See *MPR 11/29/04-01*, "ARM Debuts Logical V7.") At the high end, the Cortex-A family is reaching for higher performance in application processors without abandoning ARM's traditional strength of low power consumption; the superscalar Cortex-A8 was the first example. (See our two-part coverage in *MPR 10/25/05-02* and *MPR 11/14/05-01*, "Cortex-A8: High Speed, Low Power.")

Tucked between those two families is the midrange Cortex-R family, exemplified by the new Cortex-R4. This processor duplicates the relatively high performance and relatively low power consumption of the existing ARM9, ARM10, and ARM11 families while incorporating the latest features of the ARMv7 architecture. It's intended primarily for deeply embedded applications, such as hard-disk controllers, automotive systems, wireless modems, inkjet printers, and home network-gateway devices. As Figure 1 shows, ARM expects consumers to buy 1.25 billion of those products in 2008, growing to about 1.6 billion units in 2010.

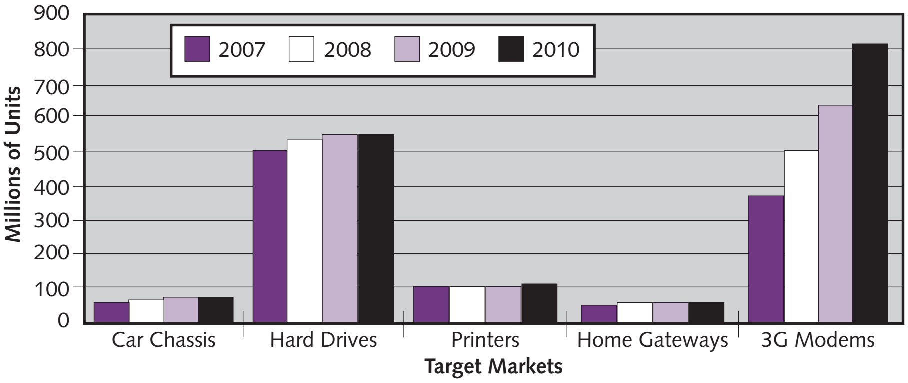

Figure 1. Of the primary target markets ARM identifies for the new Cortex-R4, hard-disk controllers and wireless modems account for the largest unit volumes. Other than wireless modems, however, these product categories aren't experiencing particularly strong growth. (Data source: ARM)

The Cortex-R4 bridges the ARM9, ARM10, and ARM11. Although it's similar to the ARM946E-S in some respects, it bears a stronger resemblance to the ARM1156T2-S. (See *MPR 1/5/04-01*, "ARM Expands ARM11 Family.") Among the features inherited from the ARM1156T2-S are an improved interface for local memories, error correction for

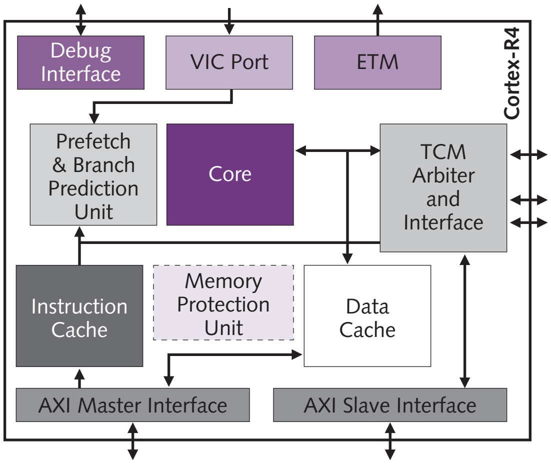

Figure 2. Cortex-R4 block diagram. This 32-bit synthesizable processor resembles the two-year-old ARM1156T2-S, but it's more configurable and has several differences that make it better suited for its intended role as a deeply embedded controller.

those memories, Thumb-2 instructions, nonmaskable interrupts, faster responses to interrupts, better memory protection, AMBA-3 AXI, and a deeper instruction pipeline for higher clock frequencies.

However, the Cortex-R4 also has several important differences from the ARM1156T2-S. It's more configurable, so developers can choose which features to include. Fully configured, it's still smaller than the ARM1156T2-S. The redesigned pipeline is more compact and clock-efficient, in some cases allowing two instructions to execute simultaneously. The I/O interfaces for tightly coupled memory (TCM) are more flexible, and this is the first ARM processor with an AXI slave port for external access to the TCMs. Like the Cortex-M3, the Cortex-R4 has a hardware divider that performs integer operations two or three times faster than other ARM processors.

Config. Feature	Options	Logic Gates Required
I-Cache	0–64K	10k (controller)
I-Cache Parity	Yes/No	TBD
D-Cache	0–64K	10k (controller)
D-Cache Parity	Yes/No	TBD
TCM Ports	0–3	1.5k gates per port
I-TCM	0–8MB	—
I-TCM Parity	Yes/No	TBD
D-TCM	0–8MB	—
D-TCM Parity	Yes/No	TBD
MPU	0, 8, 12 regions	1k gates per region
Breakpoints	2–8	850 gates per point
Watchpoints	1–8	850 gates per point

Table 1. The Cortex-R4 takes a small step toward user configurability by offering these options in prewritten scripts for the synthesis compiler. Caches and tightly coupled memories have always been somewhat configurable in ARM processors, but the Cortex-R4 offers new options, such as parity checking. The memory-protection unit (MPU) is another significant option. (TBD: to be determined.)

The CoreSight debug port is better than JTAG for debugging multicore designs, and ARM has improved the processor's overall power and cycles-per-instruction efficiency.

ARM Stretches Toward Configurability

Figure 2 is a block diagram of the Cortex-R4—or rather, one possible configuration of the processor. ARM is taking a small but welcome step in the direction of configurability. Other possible configurations of the Cortex-R4 could omit the memory-protection unit (MPU), substitute as many as three TCMs for the caches, or add TCMs as well as caches. The debug interface is configurable, too, supporting variable numbers of breakpoints and watchpoints.

In the past, *Microprocessor Report* has chided ARM for overpopulating its product line with too many nearly identical processor cores. Sometimes the difference between two members of the same processor family is as trivial as a single feature. This practice confuses customers and leads to cryptic product names, such as the ARM1156T2-S and ARM-1156T2F-S, which differ only by the absence or presence of an FPU. ARM's main competitors—ARC International, MIPS Technologies, and Tensilica—avoid this problem by offering configurable processor cores. Developers can add or subtract features at will, often with a few mouse clicks, by using GUI design tools.

Although the Cortex-R4 isn't nearly as flexible as the configurable processors from competitors, it does offer some important options, which developers can choose by modifying the synthesis scripts. This dab of configurability allows ARM to provide a single processor core that's readily adaptable for different purposes—without the confusing redundancy and product names of the past. Of course, some features of ARM processors, such as the sizes of the instruction and data caches, have been configurable for years. That hasn't changed. What's new are options for parity checking, the number of TCMs, an MPU, the number of memory regions the MPU can protect, the number of breakpoints, and the number of watchpoints. Table 1 shows the Cortex-R4's configurable options and their costs in logic gates.

Multiple Interface Options for TCMs

Notice that the Cortex-R4 supports both caches and TCMs for storing instructions and data. TCMs are managed by the application software and don't exhibit the unpredictable behavior of caches, so they are better for hard real-time systems requiring fast, deterministic responses to interrupts. Otherwise, caches are usually preferable, because the processor automatically manages instructions and data. But each cache requires a 10,000-gate controller, so that's another trade-off.

In the Cortex-R4, each cache can range in size from 4KB to 64KB. TCMs can be as large as 8MB, which is useful for attaching large amounts of flash memory in automotive systems. (In most other applications, TCMs are about the same size as caches.) The Cortex-R4 supports one, two, or three TCM ports. A dedicated AXI slave port to external memory—the first in any ARM processor—provides direct memory access (DMA) to the TCMs. The AXI interface supports out-of-order transactions, improving throughput. TCMs can segregate instructions and data, or they can function as unified memories that store instructions and data together.

The Cortex-R4 can assign TCM ports to a pair of dedicated interfaces: TCM-A has one 64-bit port, and TCM-B can have one or two 64-bit ports. This grouping gives developers a few additional choices. A very simple configuration of the Cortex-R4 could attach one TCM to interface A and use it as a unified local memory for both instructions and data. A higher-performance unified configuration could attach two TCMs to interface B and use them as interleaved memories. While the processor is fetching instructions or data from one TCM, the other TCM can make DMA transfers from main memory.

A third option is to attach one TCM to each interface, either to segregate the instructions and data or to use one TCM as a buffer for memories other than DRAM (such as flash memory or ROM). For even greater performance, the Cortex-R4 can have three TCMs. While the processor prefetches instructions from a TCM on interface A, it can also retrieve data in an interleaved fashion from a pair of TCMs on interface B. Although choosing the number of TCMs and their port configurations is a design-time decision, developers no longer must decide how to split the TCMs between instructions and data before synthesizing the logic. At runtime, the Cortex-R4 can use any TCM for either purpose.

An important feature borrowed from the ARM1156T2-S is error detection for all memory structures associated with the processor core, including TCMs and caches. ARM recognizes that cosmic radiation is causing more soft errors in chips manufactured in smaller fabrication processes. The Cortex-R4 optionally supports parity checking and error checking and correction (ECC) for all caches, TCMs, and address tags.

With parity checking enabled, the Cortex-R4 generates a parity bit when writing each byte of data and the address tags. Later, the processor checks the parity when reading the data and tags. Parity can be odd or even and is pin-configurable at runtime. Errors can force the processor either to invalidate the offending cache line (triggering a write-through access to memory) or throw a precise exception, which allows the software to recover from the error. Adding ECC is a little more complicated, because it requires developers to implement their own external ECC logic.

Each TCM port can have its own ECC. A bit error on one TCM port won't affect I/O on the other ports. In sum, the Cortex-R4's configurable memory subsystems and error-correction features make it suitable for mission-critical embedded applications, such as the controllers in automotive braking systems.

Feature	Hard Drive	Automotive	Wireless & Imaging
MPU	—	12 regions	8 regions
Caches	—	Sometimes	Yes
Memory Parity	Sometimes	Yes	—
Breakpoints & Watchpoints	Minimum	Maximum	Maximum

Table 2. ARM suggests configuring the Cortex-R4 processor core in these ways for hard-disk controllers, chassis-level automotive systems, wireless modems, and imaging systems. Refer to Table 1 for the complete list of configuration options.

Memory Protection Is Robust, Too

Another important synthesis option is whether to include an MPU—and, if so, to specify the number of main-memory regions the MPU can protect. As the previously referenced Table 1 shows, the Cortex-R4's MPU can optionally protect 8 or 12 regions at a cost of 8,000 to 12,000 additional logic gates. The MPU allows a real-time operating system (RTOS) to shield system-level processes from user-level processes and to segregate multiple user-level processes from each other. User data and stacks can occupy their own memory regions, and some regions can overlap if necessary. Regions can be as small as 32 bytes. All these features are common to the ARMv6 and ARMv7 architectures.

Memory protection, like error correction, allows developers to build more-reliable embedded systems. However, ARM's MPU is not as capable as a full-fledged memory-management unit (MMU). It doesn't have a translation lookaside buffer (TLB) or other structures required for virtual-memory addressing. Without an MMU, the Cortex-R4 can't run virtual-memory operating systems such as Linux, but that's not a handicap for a deeply embedded controller core.

Even with an MPU, the Cortex-R4 lacks the TrustZone security technology that ARM introduced in 2003. TrustZone is an additional permission mode that supplements the usual privileged and user-level modes. (See *MPR 8/25/03-01*, "ARM Dons Armor.") ARM deems TrustZone unnecessary for the deeply embedded applications that the Cortex-R4 targets. Including TrustZone would have added 15,000 to 20,000 gates to the core.

Fab Process & Cell Library	ARM Cortex-R4 Configuration	Clock Freq (Worst-Case)	Core Logic	With Cache

TSMC 130G
Sage-HS
SRAM-SP-HS-HC
	8K caches + parity;
	12-region MPU;
	maximum debug
		290MHz*	1.99mm^2	3.35mm^2

TSMC 90G
Advantage Std.
Advantage RAM
	16K caches + parity;
	12-region MPU;
	maximum debug
		400MHz*	1.31mm^2	2.04mm^2

TSMC 90G
Sage-X
Advantage RAM
	8K caches, no parity;
	8-region MPU;
	minimum debug
		273MHz†	0.86mm^2	1.43mm^2

Table 3. Using the configuration options in Table 1, ARM synthesized three different versions of the Cortex-R4 with Artisan cell libraries, targeting TSMC's generic 130nm and 90nm fabrication processes. These statistics give developers some idea of what to expect when configuring the Cortex-R4 for different applications. (*Optimized for speed. †Optimized for area.)

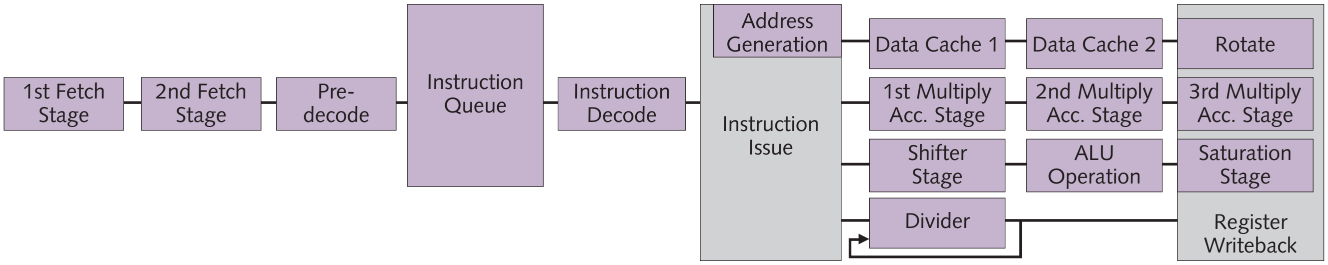

Figure 3. The Cortex-R4 pipeline is one stage shorter than the ARM1156T2-S pipeline, but ARM overhauled the pipeline instead of merely removing a stage. Given an ideal instruction sequence, the Cortex-R4 can issue, execute, and write back two instructions per clock cycle. Shortening the pipeline saves gates and power but reduces the Cortex-R4's maximum clock frequency compared with that of the ARM1156T2-S.

The remaining Cortex-R4 options listed in Table 1 are breakpoints and watchpoints. Developers can add as many as eight breakpoints and eight watchpoints to aid software debugging. Of course, the trade-off for these conveniences is the 850 gates that each breakpoint or watchpoint requires, which become useless baggage in the final design. Table 2 shows the different ways in which developers might configure the Cortex-R4 for different purposes.

Stripped down to a minimum configuration, the Cortex-R4 requires about 180,000 gates. That's about 50% larger than the older, slower, less capable ARM946E-S. Fully configured with all available options, the Cortex-R4 requires about 220,000 gates (excluding memories). That's about 30% smaller than the ARM1156T2-S, a processor sharing much in common with the Cortex-R4. Table 3 lists three different configurations of the Cortex-R4, along with statistics for maximum worst-case clock frequency and die area. This data shows how the configuration options can dramatically affect the clock speed and size of the processor.

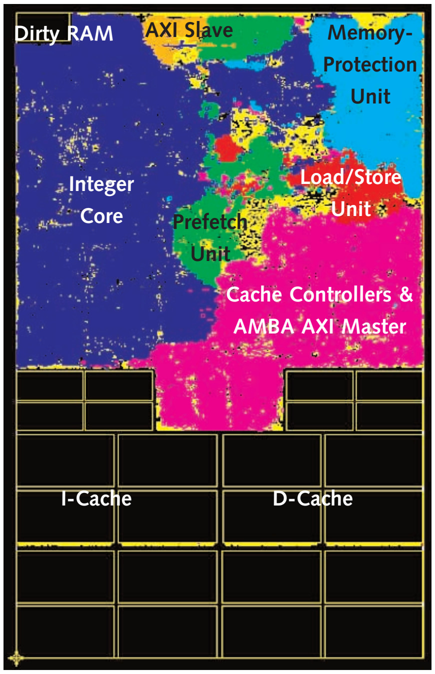

Figure 4. This synthesized layout of the Cortex-R4 includes some of the larger optional features, such as caches and an MPU. Fully configured, the processor core is about 220,000 gates (excluding memories). A minimal configuration is about 180,000 gates.

Although the Cortex-R4 inherits much from the ARM1156T2-S, ARM trimmed several features to save gates. Mainly, the Cortex-R4 has a slightly shorter pipeline than the ARM1156T2-S—eight stages instead of nine, which reduces the maximum clock frequency by about 25%. But ARM didn't perform a crude amputation. Instead, the entire pipeline is redesigned, starting with the smaller but more efficient prefetch unit, which is optimized for Thumb-2.

As Figure 3 shows, the pipeline is a conventional uniscalar design until the instruction-issue stage, when it splits into multiple execution pipes. The writeback stage is combined with the final execution stages of those pipelines. Under ideal conditions, the Cortex-R4 can issue, execute, and retire two instructions per clock cycle, much like the superscalar Cortex-A8. But because the Cortex-R4 isn't an end-to-end superscalar processor like the Cortex-A8, it doesn't spend as many gates on control structures.

ARM saved gates in various other ways, too. Cache controllers, if present, are slightly smaller than those in the ARM1156T2-S. The load/store unit is simpler (it doesn't support hit under miss), and the multiplier is smaller. There's only one AXI bus controller, but because it can handle transactions out of order, it's probably as efficient as dual in-order controllers. There's no FPU option for the Cortex-R4, either. However, the Cortex-R4 has the same hardware integer divider as the Cortex-M3, which can perform operations in 10–15 clock cycles that otherwise would take 40 cycles. And by generating memory addresses in the instruction-issue stage—one stage earlier than the ARM1156T2-S does—the Cortex-R4 reduces the load-use penalty to one clock cycle, the same as ARM9E processors. That's an improvement over the ARM1156T2-S, which levies a load-use penalty of two cycles.

Base configurations of processors from ARC and Tensilica are significantly smaller than the Cortex-R4, but their gate counts inflate rapidly as developers add features. It's not unusual for a 20,000-gate configurable processor to weigh in at 300,000 to 400,000 gates when fully loaded. All things considered, the Cortex-R4 is a little larger than other 32-bit synthesizable processors with similar performance and capabilities. Figure 4 shows a synthesized layout of the Cortex-R4.

Relaxed Timing Allows Slower SRAMs

Yet another welcome feature in the Cortex-R4 is an improved I/O interface for caches and TCMs. First seen in the ARM11 family, this interface has looser timing for data transfers, allowing developers to use slower compiled memory cells in the SRAM arrays.

Slower memory may not seem like a good thing, but it requires fewer transistors and consumes less power than faster memory. In some applications, slower memory is fast enough to get the job done, and it saves silicon and power. In addition, relaxed timing helps developers achieve timing closure on their designs, which shortens an often frustrating part of a project. If higher throughput is more important than lower power, faster memory remains an option.

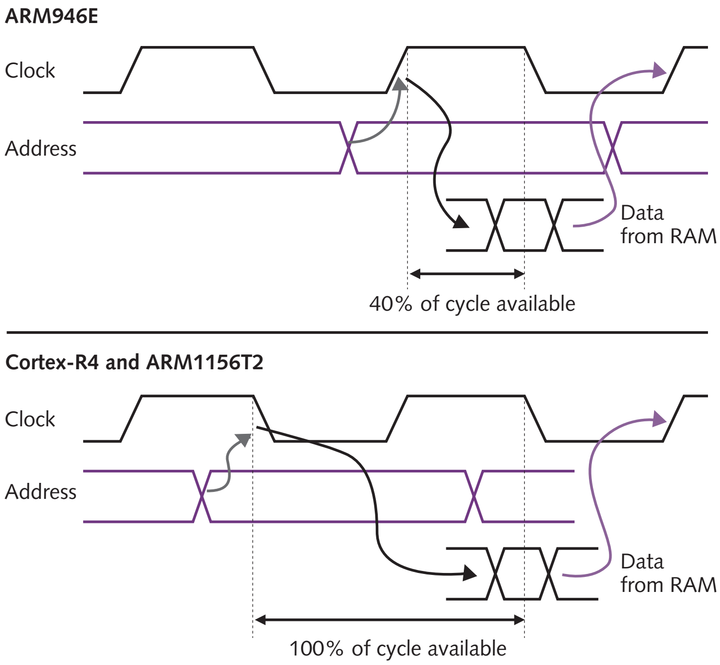

Figure 5. The Cortex-R4 inherits an improved interface for caches and TCMs from the ARM11 family. Looser timing for data transfers allows developers to use slower, lower-power SRAM arrays with the Cortex-R4. At 400MHz, the Cortex-R4's access time to local memory is 2.5ns, whereas a 300MHz ARM946E-S must access local memory in only 1.3ns.

Figure 5 illustrates the timing difference. Whereas the cache/TCM interface in the ARM946E-S leaves only 40% of a bus cycle available for transferring data ("data out"), the improved interface makes 100% of the bus cycle available for that purpose. Both interfaces make 10% of the bus cycle available for address setup. Thanks to the pipelined memory I/O and reduced load-use penalty described above, the Cortex-R4 relaxes the memory timing without sacrificing memory bandwidth or latency.

Table 4 shows the differences between two compiled-memory libraries from Artisan, a leading provider of physical intellectual property (IP) that's now a division of ARM. (See *MPR 9/7/04-01*, "ARM Extends Its Reach.") These are typical cell libraries that developers might use to compile SRAM arrays for caches or TCMs. Artisan's Metro library is optimized to reduce active power consumption and static leakage. To achieve those goals, it uses dynamic voltage scaling and proprietary leakage-reduction circuits. Artisan's Advantage library is optimized for higher performance while still maintaining good density. As Table 4 shows, the Metro library consumes only 46% as much dynamic power as the Advantage library, and it leaks only 20% as much power in standby mode.

To improve real-time performance, the Cortex-R4 has the same interrupt-handling features as ARM11-series processors. These features are a significant improvement over the ARM9E family. The Cortex-R4 supports nonmaskable interrupts (NMI) and can respond to an interrupt in only 20 clock cycles if the handler is waiting in a TCM. That compares to 118 cycles for the ARM946E-S and 54 cycles for the ARM966E-S.

In addition, Cortex-R4 exception handlers may consist entirely of Thumb-2 instructions, whereas ARM9E processors must exit Thumb mode to handle exceptions. With the help of NMIs, faster interrupt handling, and denser exception code, the Cortex-R4 is well suited for its role as a deeply embedded controller.

Competing Processors Slower, but Smaller

The Cortex-R4 is entering a vortex of competition. ARM's success—it's by far the leading processor-IP vendor—has inspired competing companies to introduce several new processor cores aimed directly at ARM's most popular products. ARC, MIPS, and Tensilica provide many alternatives to the ARM7, ARM9, ARM10, and ARM11 families, generally for lower licensing fees than ARM commands. And because ARM's competitors also offer greater configurability, developers can create a virtually unlimited number of alternatives.

However, the latest trend among configurable-processor vendors is to offer preconfigured versions of their cores. This seemingly contradictory strategy is an attempt to win more

Statistics For TSMC 90G	Artisan Metro	Artisan Advantage
Memory Die Area	0.031mm^2	0.048mm^2
Memory Access Time	2.1ns	1.23ns
Memory Power (Dynamic)	0.013mW/MHz	0.028mW/MHz
Memory Power (Leakage)	0.018mA	0.091mA

Table 4. This comparison of two memory libraries from Artisan demonstrates why looser timing on the Cortex-R4's cache and TCM interfaces is a boon for developers. By making it easier to use the Metro library, the Cortex-R4 readily adapts to applications that don't require faster memory. These statistics assume a 16KB cache (512 × 32 bits) compiled for TSMC's generic 90nm CMOS process.

designs from customers that may be intimidated by the idea of creating their own custom processor, even though the processor-configuration tools keep getting easier to use and more powerful. Over the past year, both ARC and Tensilica have introduced half a dozen preconfigured versions of their configurable cores. As Table 5 shows, some of those preconfigured cores will compete head-on with the Cortex-R4.

The most comparable competitors from ARC are the ARC 625D and ARC 750D. The 625D is based on the ARC 600 configurable-processor core, which is optimized for lower-power applications. The 750D is based on the ARC 700 configurable-processor core, which is optimized for higher throughput. Although ARC delivers both cores preconfigured, developers can use ARC's graphical design tools to customize them further, if necessary. Both cores are smaller than the Cortex-R4 and consume less power. The 625D is limited by a five-stage pipeline that prevents it from reaching the same clock frequencies as the Cortex-R4, but the 750D can match or exceed the Cortex-R4's clock speed. The 750D has a full-fledged MMU, so it can run virtual-memory operating systems such as Linux. (See *MPR 3/14/05-02*, "ARC's Preconfigured Cores.")

MIPS has a few processors similar to the Cortex-R4. A good example is the MIPS32 4KE, a derivative of the MIPS32 4K family introduced in 1999. (See *MPR 5/31/99-05*, "Jade Enriches MIPS Embedded Family.") Despite its age, the 4KE remains competitive. True, its five-stage pipeline will prevent the 4KE from reaching the same clock frequencies as the Cortex-R4, and the lack of branch prediction is another handicap. But the 4KE is appreciably smaller than the Cortex-R4 and consumes less power. Its greatest advantage is an MMU capable of running virtual-memory operating systems. In addition, MIPS offers the 4KE Pro, a configurable version of the 4KE that rivals the flexibility of ARC's and Tensilica's cores. (See *MPR 3/3/03-01*, "MIPS Embraces Configurable Technology.")

Tensilica's closest processors to the Cortex-R4 are two members of its new Diamond family of preconfigured cores: the 212GP and 570T. (See *MPR 3/20/06-01*, "Tensilica's Preconfigured Cores.") Tensilica describes the 212GP as a midrange controller core, which positions it squarely against the Cortex-R4 and some members of the ARM9 and ARM11 families. But it's significantly smaller: a mere 73,000 gates, less than half the size of a minimally configured Cortex-R4. In fact, the 212GP is about the same size as an ARM7TDMI-S. That will translate into significantly lower power consumption than the Cortex-R4, especially because Tensilica's cores have unusually extensive clock gating. What the 212GP sacrifices is clock frequency. Limited by a five-stage pipeline, it's about 15–20% slower than the Cortex-R4. Other drawbacks are a less sophisticated bus (AHB Lite instead of AMBA-3 AXI) and a much lower ceiling on the sizes of its optional caches and scratchpad memories.

Feature	ARM Cortex-R4	ARM ARM1156T2-S	ARM ARM946E-S	ARC ARC 625D	ARC ARC 750D	MIPS MIPS32 4KE	Tensilica Diamond 212GP	Tensilica Diamond 570T
Architecture	ARMv7	ARMv6T2	ARMv5TE	ARCompact	ARCompact	MIPS32	Xtensa LX	Xtensa
Core Freq	400MHz**	400MHz*	210MHz*	240MHz*	400MHz*	200–240MHz*	233–250MHz*	200–233MHz*
Pipeline Depth	8 stages	9 stages	5 stages	5 stages	7 stages	5 stages	5 stages	5 stages
Branch Predict	Dynamic	Dynamic	—	Static	Dynamic	—	—	—
Instr Length	32 bits	32 bits	32 bits	32 bits	32 bits	32 bits	24 bits	24 / 64 bits
Short Instr	16 bits Thumb-2	16 bits Thumb-2	16 bits Thumb-1	16 bits	16 bits	16 bits	16 bits	16 bits
DSP Instr	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Java Ext.	—	—	—	—	—	—	—	—
FPU	—	Optional 32/64 bits	Optional 32/64 bits	Optional 32 or 64 bits	Optional 32 or 64 bits	—	—	—
MMU/MPU	Optional MPU	Optional MPU	MPU	Optional MPU	MMU	MMU	MPU	MPU
Caches	0K–64K	0K–64K	0–1MB	0–32K	8K–64K	0–64K	8K	16K
TCM†	0–8MB	0–256K	0–1MB	0–512K	0–512K	0–1MB	0–128K	0–128K
RAM Parity	Optional	Optional	—	—	—	Yes	—	—
NMI‡	Yes	Yes	—	Yes	Yes	Yes	Yes	Yes
Bus Interface	AMBA-3 AXI	AMBA-3 AXI	AMBA AHB	AMBA AHB	AMBA AHB	MIPS BIU	AHB Lite, XLMI	AHB Lite, XLMI
Configurability	Medium	Low	Low	High	High	Medium	Low	Low
Size (Base)*	180–220k gates	240–300k gates	120k gates	0.71mm^2	1.12mm^2	0.04–1.9mm^2	73k gates	114k gates
Power/MHz	0.27–0.35mW**	0.45mW*	0.30mW*	0.08mW*	0.13mW*	0.12–0.37mW*	0.195mW*	0.275mW*
Introduction	2006	2004	2000	2005	2005	2001	2006	2006

Table 5. ARM's Cortex-R4 has plenty of competition. All the competing processors in this table are 32-bit synthesizable cores suitable for the same kinds of embedded applications as the Cortex-R4. In most cases, the Cortex-R4's eight-stage pipeline will allow it to reach slightly higher clock frequencies than processors with shorter pipelines. Another advantage for the Cortex-R4 is its AMBA-3 AXI bus interface, which isn't a standard feature from the competition. However, the Cortex-R4 tends to be larger and more power-hungry. Note that ARM's clock-speed and power estimates for the Cortex-R4 are the only ones in this table based on a 90nm fabrication process; all the others assume 0.13 micron. (*Generic 0.13-micron process, worst case. **Generic 90nm process, worst-case, 1.0V. †TCM: tightly coupled memory, also known as closely coupled memory or scratchpad RAM. ‡NMI: nonmaskable interrupts.)

A more formidable competitor from Tensilica is the Diamond 570T. It has a wider system interface than lower-end Diamond cores (64 bits vs. 32 bits) plus the same AHB Lite and XLMI buses as the 212GP. (XLMI is Tensilica's own Xtensa Local Memory Interface, which is 128 bits wide in the 212GP and 570T.) In addition, the 570T has special 32-bit I/O ports and queues. (For a detailed description, see *MPR 5/31/04-01*, "Tensilica Tackles Bottlenecks.") But the 570T's unique feature is its VLIW extensions, called Flexible-Length Instruction Xtensions (FLIX). FLIX allows the 570T to pack as many as three operations into a 64-bit instruction word. With three ALUs, two branch units, and dual multipliers, the 570T can execute as many as three FLIX operations per clock cycle, including one DSP operation. No other licensable 32-bit embedded-processor core has this capability.

Efficient and a Little More Configurable

The new Cortex-R4 is much like the two-year-old ARM1156T2-S but requires about 30% less silicon and is capable of averaging more instructions per clock cycle, albeit at a lower maximum clock frequency. Overall, the Cortex-R4 can deliver greater power efficiency at a lower cost—welcome attributes for a deeply embedded controller core. The lower maximum clock frequency isn't a serious drawback, because deeply embedded controllers are commonly found in systems that are sensitive to power consumption and radio-frequency emissions.

Another Cortex-R4 improvement that stands out is greater configurability. Without it, a typical configuration of the Cortex-R4 could logically belong to the ARM11 family. In a parallel universe without a Cortex strategy, the Cortex-R4 might be known as the ARM1146T2-S. Nevertheless, ARM still limits configurability to a few choices of core-level features and local memories, eschewing the anything-goes attitude of ARC, MIPS, and Tensilica, which allow their customers to modify the instruction-set architecture (ISA).

Price & Availability

ARM is licensing the Cortex-R4 processor core now. The synthesizable core is available in Verilog format. ARM doesn't publicly disclose upfront licensing fees or royalties. For more information, visit *www.arm.com/products/CPUs/ARM_Cortex-R4.html and www.arm.com/products/CPUs/families/CortexFamily.html*.

MPR believes ARM is resisting architectural configurability for two reasons. First, ARM fears that giving customers the unfettered ability to create their own instructions might fracture the ARM architecture. Those same fears led MIPS to impose some limits on customer extensions, although ARC and Tensilica don't seem to worry about it.

The second reason for ARM's resistance is that ARM has been spectacularly successful without configurable processors. In 16 years, ARM has risen from humble beginnings as a spinoff from Acorn Computer to become the world's leading provider of licensable processor cores. ARM processors are in virtually every cellphone and iPod, among a host of other popular products. Intel executives would gladly make sacrifices to the geek gods if doing so would make the x86 as widespread as the ARM architecture. And ARM did it without building a single fab.

Perhaps the Cortex-R4 is a step toward a middle level of configurability. Even if ARM never lets customers tinker with the ISA, the option to pick and choose from a menu of synthesizable features would make ARM's product line even more compelling to developers than it already is. And reducing the number of near-identical cores would also banish those alphabet-soup product names to the dustbin of history. ◇