Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

MULTICORE MULTITHREADING WITH MIPS

New MIPS32 1004K Coherent Processing System Has Four-Way SMP

By Tom R. Halfhill {4/28/08-01}

Four-bangers are the low-end motors of the automobile world, but quad-core microprocessors are currently the hot rods of computing. On April 1, MIPS Technologies made it easier for chip designers to create quad-core SoCs by introducing the industry's first licensable processor core supporting four-way symmetric multiprocessing (SMP) and chip multithreading. A full implementation with four dual-threaded cores offers the virtual equivalent of eight-way SMP.

Of course, chip developers have been building larger multicore SoCs for years, using embedded-processor cores from MIPS, ARM, ARC International, Tensilica, and others. And in 2004, ARM introduced the ARM11 MPCore, an ARM1176-based processor adapted for two-, three-, or four-way SMP. What sets the new MIPS 1004K Coherent Processing System (CPS) apart from the crowd is its unique combination of coherent SMP and hardware multithreading, licensed as synthesizable intellectual property (IP).

The MIPS 1004K CPS is a package of IP that includes the new MIPS32 1004K ("ten-oh-four kay") processor core plus the memory-management and I/O components needed to design a coherent two-, three-, or four-way SMP subsystem. (Single-core designs are also possible.) It has numerous configuration options, such as single- or dual-threading per core, I/O coherency, global interrupt controls, and coherent debug units. The MIPS 1004K CPS is portable to any digital-IC fabrication process, and its Open Core Protocol (OCP) bus enables easy integration with other embedded IP.

Son of the MIPS 34K Processor

Essentially, the base configuration of the 1004K core is a MIPS32 34K processor core adapted for SMP. The 34K, introduced in 2006, was the first licensable processor core with hardware multithreading. (See *MPR 2/27/06-01*, "MIPS Threads the Needle.") This feature allows instructions from two or more software processes to share the same CPU pipeline at the same time, with single-cycle context switching. A "process," in this sense, may be a lightweight thread within a program or a heavyweight task, such as an operating system or application program. The CPU maintains duplicate registers, stacks, and flags to preserve the state of each thread during context switches. (Intel refers to its version of the technology as Hyper-Threading.)

Although the latest version of the 34K processor supports up to nine threads per core, MIPS limits each 1004K core to two threads, at least for now. This limit reduces the opportunity to perform useful processing when a thread stalls on a memory access, because the second thread may stall, too. However, MIPS says that larger-scale multithreading in an SMP design would needlessly bloat the cores while delivering sharply diminishing returns in throughput. (Even in its largest PC and server microprocessors, Intel has never ventured beyond two threads per core, either.) MIPS will revisit this decision as fabrication technology keeps improving and demand for higher performance keeps rising.

The 1004K CPS is intended for building application processors in high-performance embedded systems, especially consumer electronics—a MIPS stronghold. Among the targets are HDTVs; next-generation TV set-top boxes with HD and video-recorder functions; home networking appliances; multifunction printers; and mobile computing devices with wireless Internet access.

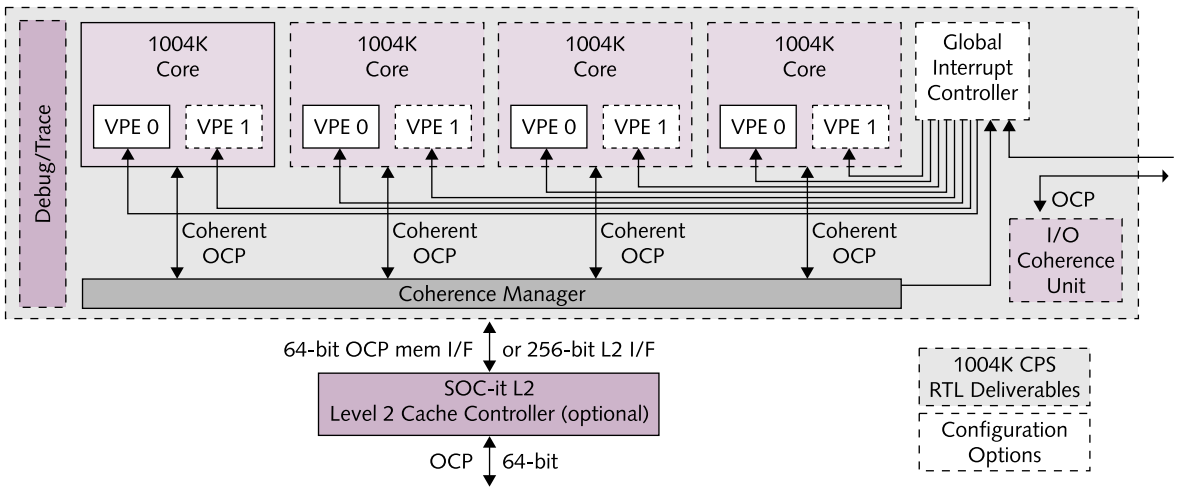

Figure 1. Four-way SMP with the MIPS Coherent Processing System (CPS). In this full-blown configuration, four 1004K processor cores are each capable of dual multithreading, providing the virtual equivalent of an eight-way SMP system. Smaller designs are possible, using fewer cores and/or one thread per core. The coherence manager is a hardware block that maintains coherency among the shared memories, transparent to software. The global interrupt controller does likewise for interrupts. The optional I/O coherence unit preserves coherency for input/output operations. The optional debug/trace unit is similarly coherent and can access all the processors and other blocks.

Many of these products must run powerful operating systems, such as Linux or Windows CE. They may also require a Java virtual machine and consumer-friendly graphical user interface. Reaching for higher performance by simply scaling the clock frequency might bust the power budget. MIPS says a four-way SMP configuration of the 1004K CPS with multithreading can hold typical power consumption to about 1.6W at 800MHz, assuming fabrication in a 65nm CMOS process.

Although the 1004K CPS launches MIPS into a higher realm of embedded-processor performance, it still doesn't compete directly with high-performance standard-part microprocessors. For instance, MIPS licensee RMI sells multicore chips with as many as eight 64-bit processor cores, each with four-way multithreading. As a MIPS architectural licensee, RMI designs its own MIPS-compatible cores. RMI's XLR and XLS chips are designed for communications equipment and other high-end applications. (See *MPR 5/17/05-01*, "A New MIPS Powerhouse Arrives.") The MIPS 1004K CPS isn't quite so muscular, but it allows any MIPS licensee to design an SMP chip that's suited for lower-power embedded systems.

Some MIPS customers may find their 1004K-based designs competing with a standard-part newbie to the market—Intel's x86-compatible Atom microprocessors. One of Intel's target applications for Atom is mobile Internet-access devices that are larger than a cellphone but smaller than a notebook computer. MIPS believes the 1004K CPS is ideal for the same products. An important difference between the 1004K CPS and Atom is that the former is licensable IP, whereas the latter is an off-the-shelf solution. Developers can build highly optimized SoCs around the 1004K CPS, or they can save a year or two by purchasing standard parts. (See *MPR 4/7/08-01*, "Intel's Tiny Atom.")

Ready for Replication

Because the 1004K core is almost identical to the 34K core, this article won't dwell on their core-level features, already described in our previously referenced 34K article. It's enough to say that both processors are based on the MIPS32 Release 2 architecture and have nine-stage uniscalar pipelines, in-order execution, memory-management units (MMU), optional FPUs, and configurable instruction/data caches (0–64KB each). Both support the subset of MIPS16e 16-bit instructions for greater code density, as well as the MIPS DSP Application-Specific Extensions (ASE) for digital-signal processing. (See *MPR 5/31/05-01*, "The MIPS32 24KE Core Family.") The most significant difference, noted above, is that the 1004K CPS supports only two threads per core instead of nine.

Both the 34K and 1004K processors support MIPS CorExtend technology, which lets developers extend the instruction-set architecture. (See *MPR 3/3/03-01*, "MIPS Embraces Configurable Technology.") Because of operating-system limitations, all 1004K processors in a multicore SMP subsystem must have the same core configuration—developers can't customize one core differently from the others. However, this restriction doesn't preclude additional cores outside the SMP subsystem from having different configurations. Indeed, those cores may be completely different processors. For example, an SoC could have a two-, three-, or four-way 1004K subsystem for application processing, plus a MIPS32 24KE processor or DSP core for signal processing.

What truly sets the 1004K CPS apart from the 34K processor is the additional hardware for coherent SMP.

("Hardware" is a somewhat misleading term, because all the IP is synthesizable, but the end result is hardware.) In addition to supporting as many as four cores, the CPS has several new components: a coherence manager, a global interrupt controller, an I/O coherence unit, a coherent debug/trace unit, and a coherent OCP bus that ties everything together. These components allow a 1004K-based system to perform SMP while maintaining full coherency among the processors, caches, memory, and peripherals. The 1004K CPS supports the popular modified-exclusive-shared-invalid (MESI) coherency protocol.

With an SMP embedded operating system (so far, Linux is the only choice for the 1004K CPS), these gory details are hidden from programmers. No code is needed to manage coherency in software, and the processors needn't bear the burden of running such code. Consequently, a hardware-managed SMP system can deliver better throughput (or the same throughput at lower power levels) than a software-managed SMP system can. Although these principles are well known to server architects, they're worth repeating for those new to SMP on a chip.

Figure 1 is a block diagram of a four-way SMP design using the MIPS 1004K CPS. It's virtually an eight-way design, because each 1004K core has two virtual processing elements (VPE). Essentially, a VPE is the hardware needed to support an independent thread context. Each VPE can manage one thread and has its own interface to the global interrupt controller. In addition, each processor has a 64-bit OCP interface to the coherence manager. In turn, the coherence manager has either a 64-bit OCP interface to main memory or a 256-bit interface to the optional L2 cache controller. (The L2 controller is part of the MIPS SOC-it IP library.) The coherence manager can transfer 32 bytes of data per clock cycle to or from the cache controller.

Feature	MIPS 34Kc	MIPS 1004Kc	MIPS 1004Kc
Architecture	MIPS32 R2	MIPS32 R2	MIPS32 R2
Core Configuration	1	2	4
CPU Threads	2	4	8
Pipeline Depth	9 stages	9 stages	9 stages
L1 Cache (I / D)	32K / 32K (Config 0–64K)	32K / 32K (Config 0–64K)	32K / 32K (Config 0–64K)
MMU	Yes	Yes	Yes
FPU	—	—	—
MIPS16e	Yes	Yes	Yes
DSP ASE	Yes	Yes	Yes
Coherence Mgr	—	Yes	Yes
Global Interrupt Ctrl	—	Yes	Yes
I/O Coherence Unit	—	—	Yes
Debug Unit	—	—	Yes
Core Frequency	800MHz	800MHz	800MHz
IC Process	TSMC 65nm GP Speed optimized	TSMC 65nm GP Speed optimized	TSMC 65nm GP Speed optimized
Die Area	1.7mm^2	3.8mm^2	7.5mm^2
Dhrystone 2.1	>1,300Dmips	2,500Dmips	5,000Dmips
Power (typical)	0.4W	~0.8W	~1.6W

Table 1. Comparison of dual- and quad-core MIPS 1004Kc CPS configurations with the single-core MIPS 34Kc. (The "c" suffix indicates that these cores lack their optional FPUs; the same cores with FPUs are designated the 1004Kf and 34Kf.) MIPS has estimated the die areas, clock frequencies, and typical power consumption, assuming fabrication in TSMC's 65nm GP process with nine-track low-threshold-voltage standard cells and Dolphin standard memory cells. These are post-layout estimates with clock trees, ready for tapeout. The estimates assume worst-case conditions and capacitance but don't include on-chip variations and PLL clock jitter.

Relatively Little Core Bloat

MIPS says the fully configured subsystem in Figure 1, including a 256KB L2 cache, would occupy less than 10mm^2 of silicon, assuming fabrication in 65nm CMOS. The L2 cache would account for about 2.5mm^2 of that area. All told, it's admirably small for a four-way SMP system (eight-way SMP, with multithreading). Not many years ago, such a system would have filled a sizable box. And the 1004K core delivers much more throughput than the early MIPS microprocessors that once powered some of the industry's fastest workstations and servers. Architecturally, the 1004K core is roughly similar to the MIPS R3000 microprocessor of 1990, but it runs 32 times faster (800MHz vs. 25MHz) while using only 10% as much power (about 0.4W vs. 4.0W).

Table 1 shows more details of dual- and quad-core 1004K CPS configurations and compares them with a MIPS 34K processor. All these configurations assume dual-threaded cores. Notice that a dual-core 1004K CPS design is only 2.2 times larger than a single-core 34K design, and that a quad-core 1004K design is only 4.4 times larger. SMP adds surprisingly little overhead in die area per core.

Part of the SMP overhead in each core is a new "intervention port" for cache snooping. This port to the on-chip OCP bus has 64-bit datapaths with 32-bit memory addressing, and it enables coherency among multiple cores in an SMP configuration. In the background, invisible to application software, the processors keep duplicate tags for their L1 data caches and compare tags to see if any processor has modified its cached data. If one of these snoops finds a discrepancy, the processors ask the coherence manager to synchronize their caches and update the tags. Note that the intervention port is *not* the coprocessor port (COP) typically found on MIPS processors. The COP remains available for other purposes.

Figure 2 is a block diagram of the coherence manager that supervises cache snooping and other SMP operations. In accordance with the MESI protocol, the coherence manager automatically synchronizes the L1 caches by copying data from one cache to another. In addition, the coherence manager supervises all accesses to the optional L2 cache or directly to main memory, if there's no L2.

An optional feature of the coherence manager is speculative memory access—it can start reading data from

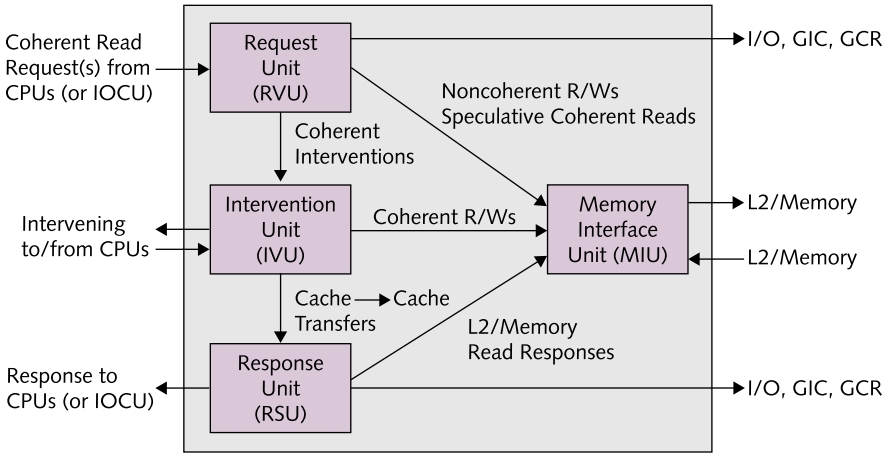

Figure 2. Coherence-manager block diagram. This key component maintains coherency among multiple processors by synchronizing their L1 data caches and supervising all accesses to memory. The coherence manager talks to the processors over an OCP bus connected to dedicated intervention ports at each end. If the optional I/O coherence unit is present, the coherence manager shares connections with it as well. The memory interface unit may have a 64-bit OCP port directly to main memory or a 256-bit interface to the optional L2 cache. Another port connects to the global interrupt controller.

memory while looking for the same data in the other cores' caches. If the coherence manager can't find the data in the caches, the prefetched data from memory may already be available. In return for all its services, the coherence manager occupies less than 0.4mm^2 of silicon at 65nm. It runs at the same clock frequency as the processors.

Additional Configuration Options

The global interrupt controller (GIC) handles external system-level interrupts as well as interprocessor interrupts—and thread-level interrupts, if multithreading is enabled. At design time, developers can configure the GIC to recognize as many as 256 different interrupts.

Another configuration option is the I/O coherence unit (IOCU). As Figure 3 shows, it sits on the OCP bus between the coherence manager and peripheral I/O devices, where it maintains coherency among those peripherals and their memory transactions. Although the IOCU is part of the 1004K CPS, developers can choose to omit it from their designs. MIPS decided to make the IOCU a separate component, apart from the primary coherence manager, because some existing system designs may already be managing I/O coherency in software. Developers unwilling to rewrite this software when updating a design can leave out the IOCU, which is a little more than half the size of the primary coherence manager (~0.25mm^2 vs. <0.4mm^2). Other developers may choose to include the IOCU, trading a little silicon and power for hardware-managed I/O coherency.

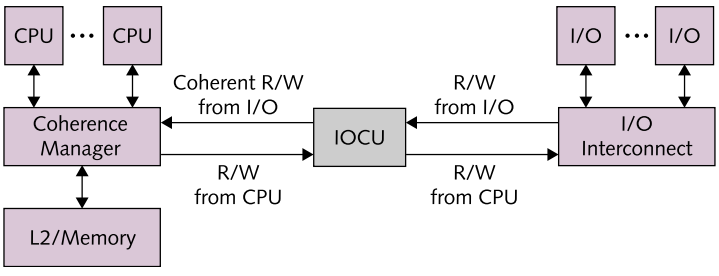

Figure 3. The optional I/O coherence unit (IOCU) connects I/O peripheral devices to the SMP subsystem. Together with the primary coherence manager, the IOCU maintains coherency among peripheral I/O devices, caches, and main memory, relieving the CPU of managing coherency in software.

The IOCU has several capabilities, some of which are configurable at design time or programmable at run time. It can snoop the L1 caches in all the processors and snoop their shared L2 cache, or it can snoop only the L2. It can write data directly into the L2 cache, bypassing the processors altogether. It can repackage bursts of DMA traffic from the I/O bus to conform with the cache lines and word lengths in the coherent memories. And it can rearrange memory requests to give peripheral I/O transactions priority over CPU transactions.

In addition, the IOCU can perform noncoherent data transfers. This feature seems odd for a coherence unit, but it's useful when some DMA traffic requires coherency while other data does not. For example, a TCP/IP router must examine the packet headers for routing information, but it may be uninterested in the packet payloads. The IOCU can maintain coherency for the header data while handling the payload data noncoherently. Payloads are usually larger than headers, so this arrangement reduces the amount of data the system must keep coherent.

Another design-time configuration option for the MIPS 1004K CPS is an SMP-aware debug/trace unit. It supports Ethernet and USB interfaces and the MIPS PDtrace extensions to EJTAG. PDtrace has an Eclipse-based debugging tool that displays a color-coded list of running processes. Developers can halt and single-step individual processor cores—even individual threads—while other processors and threads keep running. In most respects, PDtrace works the same on the 1004K CPS as it does on the 34K processor, except that it's aware of coherency in an SMP system. It keeps track of shared memory and caches, and it can trace execution through the coherence manager and IOCU. MIPS hasn't publicly disclosed the size of this unit.

Many Approaches to High Performance

On the surface, the MIPS 1004K CPS is straightforward. It gives developers a self-contained package of synthesizable IP for building a four-way SMP subsystem on an SoC. Basically, MIPS is shrinking what used to be called a server onto a single chip. Perhaps the most difficult challenge for developers is determining the optimum configuration for a particular design problem.

In the past, project managers had enough trouble deciding among several competing embedded-processor architectures, then choosing a processor core representing that architecture. Now, as multicore SoCs become common, developers must not only find the optimum processor but also determine the optimum number of processors, the optimum multicore configuration for those processors, and (with the 1004K core) the optimum number of hardware threads for those processors.

Even if a developer settles on MIPS instead of a rival CPU architecture, many choices remain. One path to high throughput is the MIPS32 74K processor, which uses two-way superscalar pipelining, out-of-order execution, and fast clock speeds to wring superlative performance from a single core. (See *MPR 5/29/07-01*, "MIPS 74K Goes Superscalar," and *MPR 6/4/07-01*, "MIPS 74K Performance Update.") Another choice is the MIPS 34K processor, which allows up to nine threads per core. Of course, it's possible to design multicore SoCs using these cores and other MIPS processors instead of the new 1004K CPS. Indeed, most MIPS-based designs already have multiple cores.

MIPS suggests that the 74K processor is preferable to the 1004K CPS when developers must use an operating system that doesn't yet support SMP, or when the application software is mostly single threaded, or when clock speeds up to 1.0GHz can deliver the necessary performance without exceeding the power budget. The 1004K CPS looks better if multiple tasks must run concurrently, or if a single processor running at 1.0GHz is insufficient or impractical. If a system frequently uses DMA to shovel data from one place to another, the 1004K's IOCU can lift the burden of DMA cache-coherency management from the CPU.

After narrowing down all these choices to the 1004K CPS, developers face more decisions. What is the optimum configuration of 1004K processors and threads? Are two single-thread cores better than one dual-thread core? Are two dual-thread cores as good as four single-thread cores? Using MIPS CorExtend technology, would a few application-specific instructions make a bigger difference than adding more cores, more threads, or more clock cycles? Which alternative offers the best balance of throughput and power consumption? When does a particular configuration reach its point of prohibitively diminishing returns?

These kinds of questions will keep multiplying in step with the number of processor cores that fit on a chip and with the growing capabilities of those processors. There are other considerations as well, far beyond the scope of this article. For instance, the best solution may not be a shared-memory SMP system at all, but perhaps a parallel-memory system that doesn't need coherency, or an asymmetric design with heterogeneous processors. Embedded-system developers are now facing design decisions that were once the exclusive concern of server architects and large-system analysts.

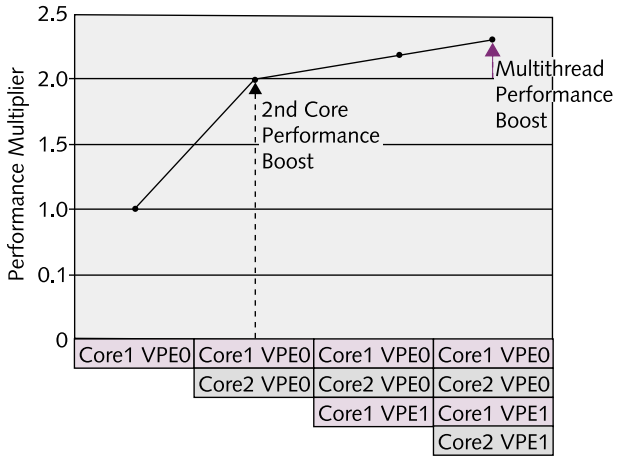

Figure 4. JPEG decompression on four different configurations of the MIPS 1004K CPS. Adding a second single-thread core gives the biggest boost in throughput over one single-thread core. It doubles performance, but it also doubles the die area and power consumption of the processors. The third configuration adds dual threading to one processor in a dual-core configuration. This enhancement boosts performance by a more modest 7% over the second configuration while adding less than 10% more die area to one of the cores. The fourth configuration enables dual threading in both cores, improving throughput by about 14% over the second configuration while adding less than 10% more die area to both cores. (Data source: MIPS.)

Evaluating Multiple Cores and Threads

The only sure way to answer these questions is to model different designs in simulation, then test representative examples of application code. Of course, the larger the design, the slower it runs in simulation. This trial-and-error process is time consuming and could easily lengthen a design project by several months. Yet the answers are crucial, unless time-to-market considerations outweigh other factors, such as optimum performance, power consumption, and die cost.

Figure 4 shows the results of one such exploration. MIPS ran EEMBC's JPEG decompression test on four simulated configurations of the 1004K CPS. The simplest configuration used only one single-thread core. The most complex configuration had two dual-thread cores—only half as complex as a full quad-core, dual-thread design. For this exercise, MIPS didn't simulate a triple- or quad-core configuration. The main purpose of this comparison (and others, not shown here) was to highlight the performance boost from multithreading, not the boost from multiple cores,

Feature	MIPS 1004Kc and 1004Kf	ARM ARM11 MPCore	ARM Cortex-A9 MPCore
CPU Architecture	MIPS32 R2	ARMv6	ARMv7 Cortex-A
Architecture Width	32 bits	32 bits	32 bits
Ancestor CPU Core	MIPS32 34K	ARM1176JZ(F)-S	ARM11 MPCore
Symmetric Multiprocessing	2–4 cores (MIPS CPS)	2–4 cores (ARM11 MPCore)	2–4 cores (ARM MPCore)
Coherent On-Chip Bus	64-bit OCP	Dual 64-bit AMBA-3 AXI	Dual 64-bit AMBA-3 AXI
Intercore Coherency	Yes	Yes	Yes
Coherent Cache Snoop	L1 and (optional) L2	L1	L1 and (optional) L2
Global Interrupt Control	Yes, optional use	Yes	Yes
Coherent I/O Control	Yes, optional use	—	Yes
Coherent Debug Unit	Yes, optional use (MIPS PDtrace)	Yes (ARM CoreSight)	Yes (ARM CoreSight)
Threads Per Core	1 or 2	1	1
Pipeline Depth	9 stages	8 stages	8 stages (9–11 clocks)
Superscalar Execution	—	—	2-way decode, 4-way issue
Out-of-Order Execution	—	—	Yes, with speculation
Branch Prediction	Dynamic	Dynamic	Dynamic
L1 Cache (I / D)	0–64K per core	16–64K per core	16–64K per core
L2 Cache	Optional (MIPS SOC-it L2 controller)	Optional (ARM L220 or PL310 L2 controller)	Optional (ARM PL310 L2 controller)
MMU	Yes	Yes	Yes
FPU	Optional, SP / DP (MIPS32 1004Kf)	Optional, SP / DP (ARM VFP11)	Optional, SP / DP (New Cortex-A9 FPU)
16-Bit Instruction Subset	Yes (MIPS16e)	Yes (Thumb-1)	Yes (Thumb-2)
DSP / SIMD Extensions	Yes (MIPS DSP ASE)	Yes (ARM DSP and SIMD)	Optional (ARM Neon)
Java Extensions	—	Optional (Jazelle DBX)	Optional (Jazelle DBX or Jazelle RCT)
Custom Extensions	Yes (MIPS CorExtend)	—	—
Secure Execution Mode	—	—	Yes (ARM TrustZone)
Core Frequency (Maximum, worst case)	800MHz (65nm GP, speed optimized)	610MHz (90nm G, speed optimized)	1.0GHz (65nm GP, speed optimized)
Dhrystone 2.1 (per core)	1.56Dmips / MHz	1.25Dmips / MHz	2.5Dmips / MHz
Power (Typical)	~0.5mW / MHz	0.27mW / MHz	n/a
Final RTL Availability	June 2008	2004	April 2008

Table 2. Feature comparison of the MIPS 1004K CPS, ARM11 MPCore, and ARM Cortex-A9 MPCore. The most important characteristic bringing these processor cores into direct contention is their capability for coherent SMP. All three processors support dual-, triple-, and quad-core SMP configurations with coherent shared memory. At the core level, the Cortex-A9 has the advantage of two-way superscalar execution and out-of-order processing, but the 1004K processor offers dual multithreading. Differences between their optional extensions may also make or break a deal. Consider the Dhrystone scores a rough starting point for comparing throughputs of these complex processors. Likewise for the clock-frequency and power-consumption estimates, which depend greatly on physical-IP libraries and other factors. ARM still hasn't released power-consumption estimates for the Cortex-A9. (n/a: data not available.)

because multithreading is a distinguishing feature of the MIPS 1004K CPS.

MIPS has been testing similar 1004K CPS configurations with various EEMBC benchmarks, including networking tests, cryptography tests, video tests, and office-automation tests—all relevant to real-world applications. Unfortunately, MIPS hasn't yet published certified EEMBC scores for these tests. However, MIPS says these preliminary tests are yielding similar results. Adding cores can double, triple, or quadruple performance over a single core—assuming there's enough data parallelism in the code to exploit the additional resources, and assuming there's enough I/O bandwidth to keep the cores fed. Naturally, adding cores also increases the die area. Enabling dual threads in one or more cores improves performance by a lesser degree (up to 15–30%) but has a negligible effect on die area.

The effect of various SMP configurations on power consumption is more difficult to estimate. EEMBC's EnergyBench requires actual silicon for testing, and the MIPS 1004K CPS is at least a year away from that point. Adding more cores can be expected to increase power consumption at a fairly linear rate. Multithreading improves pipeline utilization and overall efficiency, which should translate into expending less energy to perform tasks. However, multithreading may also require larger caches to prevent thrashing when two or more contexts compete for cache space. Much depends on the nature of the application and the holistic system design. Unfortunately, these behaviors are

even more difficult to model than application-level throughput.

MIPS 1004K CPS vs. ARM MPCores

The MIPS 1004K CPS competes head-to-head with two of ARM's SMP-capable processors: the four-year-old ARM11 MPCore and the new Cortex-A9 MPCore. All are 32-bit synthesizable embedded-processor cores supporting coherent SMP in dual-, triple-, or quad-core configurations. Their relatively deep pipelines permit clock speeds near 1.0GHz when chips are fabricated in speed-optimized 65nm CMOS. All three cores are for high-performance application processors in consumer-electronics products. (See *MPR 5/24/04-01*, "ARM Opens Up to SMP.")

They also have interesting differences. MIPS derived the 1004K core from the multithreaded 34K core, retaining the same basic in-order uniscalar pipeline. ARM derived the Cortex-A9 from the ARM11 MPCore, which in turn was an enhancement of the ARM1176JZF-S processor. Both ARM11 cores have in-order uniscalar pipelines. (See *MPR 1/5/04-01*, "ARM Expands ARM11 Family.") For the Cortex-A9, ARM added two-way superscalar pipelines, out-of-order instruction processing, speculative execution, and a faster FPU—much like the features of the Cortex-A8. (See *MPR 10/25/05-02* and *MPR 11/14/05-01*, "Cortex-A8: High Speed, Low Power.")

Theoretically, the Cortex-A9 can execute at least twice as many instructions per clock cycle as the ARM11 MPCore or MIPS 1004K processors. In reality, a superscalar processor with out-of-order execution is lucky to average 1.5 times more instructions per cycle than a uniscalar in-order processor, except under ideal conditions. Although the ancient Dhrystone benchmark is a poor way to measure this performance, the Dhrystone scores do grossly reflect the differences among these microarchitectures. MIPS says the 1004K processor executes 1,250Dmips at 800MHz, or 1.56Dmips per megahertz. ARM says the ARM11 MPCore executes 1.25Dmips per megahertz and the Cortex-A9 executes 2.5Dmips per megahertz. (When announcing the Cortex-A9 last year, ARM estimated the performance at 2.0Dmips per megahertz. ARM recently raised that score, claiming the original estimate was conservative. *MPR* doesn't hold Dhrystone benchmarks in high regard, but they are the only common metrics available for these processors at this time.)

MIPS could have derived the 1004K core from the superscalar out-of-order 74K processor instead of the simpler 34K processor, essentially matching the Cortex-A9's instruction-juggling talents. Instead, MIPS opted for less complexity, even to the point of trimming the maximum number of hardware threads from nine to two. Note that multithreading, which is absent from the Cortex-A9, wins back some execution bandwidth that MIPS sacrificed by not going superscalar and out of order. This compensation is poorly measured by the Dhrystone benchmark, whose single-threaded programs fit entirely into the caches of modern processors. Depending on the nature of the application code, a multithreaded 1004K processor may not be surrendering much throughput to the fancier Cortex-A9, and the 1004K core's relative simplicity conserves silicon and power. (ARM still hasn't released power-consumption estimates for the Cortex-A9, despite shipping production-ready RTL in April.)

Other features distinguish these processors. The Cortex-A9 has a new Accelerator Coherence Port, not found in the ARM11 MPCore. This port enables hardware-managed coherence between the MMU and caches—a feature duplicated in the MIPS 1004K CPS. We judge ARM's optional Neon extensions for the Cortex-A9 to be superior to the 1004K's thinner DSP ASE. In addition, ARM offers optional Java-acceleration extensions, and the Cortex-A9 supports ARM's TrustZone secure execution mode. (See *MPR 7/11/05-01*, "ARM Strengthens Java Compilers," and *MPR 8/25/03-01*, "ARM Dons Armor.") On the other hand, MIPS CorExtend technology lets developers add custom extensions, an important feature that ARM disdains. Table 2 summarizes these competing cores.

Unfortunately, at this point, we cannot evaluate a critical characteristic of these processors: their ease of use for developers. Although the MIPS 1004K CPS, ARM11 MPCore, and ARM Cortex-A9 MPCore are closely matched in features, it's not clear which package of SMP-ready IP and tools is easier to work with. ARM may have gained a slight advantage by deriving the MPCore features of the Cortex-A9 from the existing ARM11 MPCore. Since its debut in 2004, the ARM11 MPCore has accumulated 14 licensees. The new Cortex-A9 has already won seven licensees. These customers are unlikely to switch to MIPS. Other developers may also favor ARM's more popular CPU architecture and established SMP technology.

Although the MIPS 1004K CPS is the first package of IP from MIPS that supports coherent SMP out of the box, several MIPS customers have created their own SMP

Price & Availability

Production-ready RTL for the MIPS 1004K Coherent Processing System (CPS) is scheduled to ship in June. Some customers are already working with prerelease versions. Like most processor-IP vendors, MIPS Technologies doesn't publicly disclose license fees, but MIPS says the 1004K CPS will be priced in line with other MIPS high-performance embedded-processor cores, such as the MIPS32 34K and MIPS32 74K cores. There are two basic versions of the MIPS32 1004K processor: the 1004Kc (without FPU) and 1004Kf (with FPU). Other options include MIPS CorExtend technology and an L2 cache controller (available separately as part of the MIPS SOC-it IP library). For more information, visit *www.mips.com/products/cores/32-bit-cores/mips32-1004k/*.

designs using other MIPS processor cores. These customers are likely candidates to adopt the new 1004K CPS. Additionally, almost all MIPS 34K customers are running an SMP operating system, implementing SMP in multiple threads instead of on multiple processors. These customers could probably move their application software onto the 1004K CPS with little effort.

MIPS 1004K CPS Expands Choices

Overall, the MIPS 1004K CPS is a well-integrated SMP subsystem with lots of configurable flexibility. Developers already using the MIPS architecture—and there are plenty in the consumer-electronics industry—will find the 1004K an attractive new avenue to higher performance. In a little over two years, MIPS has impressively expanded its line of high-performance 32-bit processors by introducing the 34K, 74K, and 1004K cores.

More important, MIPS offers high performance in distinctive ways. The 34K processor can execute as many as nine threads in hardware, well suited for multitasking applications. The 74K processor has deep pipelining, two-way superscalar execution, and out-of-order processing, for superb single-core throughput at high clock speeds. The new 1004K CPS brings dual threading and coherent SMP to multicore designs with up to four processors. This diversity is unprecedented in a product line from one vendor. Furthermore, the configurability of these processors provides many opportunities to fine-tune their performance. Developers willing and able to explore these options are almost sure to find a good solution for their design problems.