Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

ARM's Midsize Multiprocessor

New Cortex-A5 Supports Four-Way Coherent Multiprocessing

By Tom R. Halfhill {10/26/09-01}

Multicore processors are becoming so commonplace that even basic cellphones, MP3 players, and other mobile embedded systems are embracing them. That's why ARM has announced its smallest Cortex A-series multiprocessor core. In a single-core configuration, it's small enough for workhorse microcontrollers, but a four-horse team of them can haul much bigger loads.

The new Cortex-A5, code-named Sparrow, was announced on October 21 at ARM's TechCon^3 conference in Silicon Valley. It's the third member of the Cortex-A family. Although it's smaller and slower than the Cortex-A8 or Cortex-A9 MPCore, it supports coherent multiprocessing with up to four cores, as well as uniprocessor configurations. As Figure 1 shows, ARM is positioning the 32-bit Cortex-A5 as a superior substitute for the five-year-old ARM1176JZ(F)-S and a major upgrade from the eight-year-old ARM926EJ-S.

Compared with its bigger brothers in the Cortex-A family, the Cortex-A5 sacrifices some clock-frequency headroom and throughput. In compensation, it uses less energy and silicon. It's binary-compatible with software compiled for the Cortex-A8 and Cortex-A9, and it makes only a few compromises on Cortex-A features.

Like other A-series processors, the Cortex-A5 has a full-fledged memory-management unit (MMU). With virtual memory addressing, it can run sophisticated embedded operating systems, such as Linux, Windows CE, Symbian, and Android. For security, the Cortex-A5 has ARM's TrustZone protected-execution technology. For greater code density, it supports the 16- and 32-bit instructions in the Thumb-2 instruction subset. To accelerate programs written in Java and Microsoft's .NET languages, it has ARM's Jazelle extensions. For multimedia

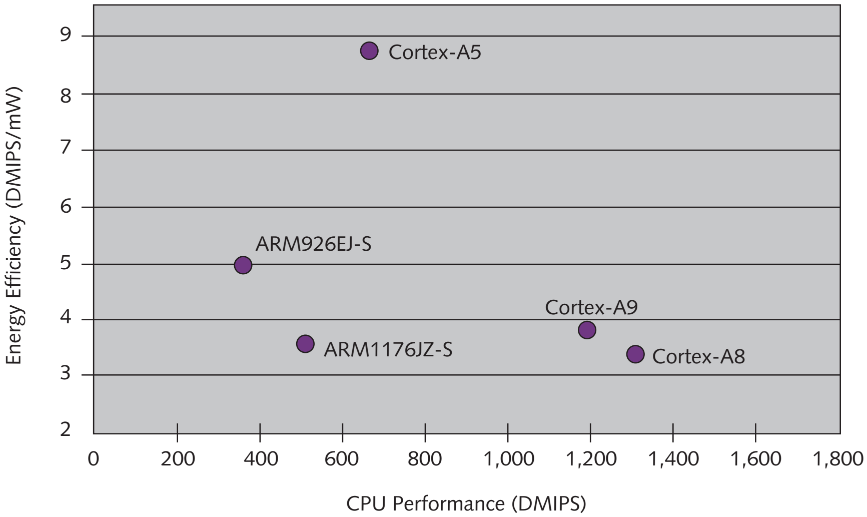

Figure 1. Even in a uniprocessor configuration, ARM's new Cortex-A5 is faster than similar members of the ARM9 and ARM11 families and is much more energy efficient. The Cortex-A8 and Cortex-A9 are faster than the Cortex-A5, but their higher power consumption will likely exclude them from applications targeted by the new processor. This figure uses ARM's data for speed-optimized cores synthesized with Advantage 10 libraries for 1.2V operation, with fabrication in TSMC's 65nm-LP process.

applications and light-duty signal processing, it offers the options of ARM's Neon extensions and a 32/64-bit FPU.

A few years ago, such a wealth of features—not to mention four-way coherent multiprocessing—would have put the Cortex-A5 atop the heap of 32-bit embedded processors. These days, the Cortex-A5 is merely a midsize model. It's a good solution for designs that don't need the higher throughput of a Cortex-A8 or Cortex-A9 but do need more horsepower and features than older cores can deliver.

Essentially, ARM has reengineered the ARM11 for the greater challenges of today's Internet-connected embedded systems. For ARM926 users, the Cortex-A5 is particularly attractive, because it significantly improves performance while holding the line on core size and power. The Cortex-A5 also will help ARM resist encroachments by Intel and the x86.

A Strong Family Resemblance

ARM defines Cortex-A cores as "applications processors" for running high-level software and sophisticated embedded operating systems. The midrange Cortex-R series is intended primarily for real-time systems, and the low-end Cortex-M series is intended for microcontrollers. (See *MPR 11/29/04-01*, "ARM Debuts Logical V7.")

Nevertheless, there's overlap. *Microprocessor Report* expects the first Cortex-A5 designs to be 32-bit microcontrollers—the kinds of devices now using ARM926 or ARM1176 cores. We expect the first controllers to be uniprocessor designs appearing late next year. They will be followed by multiprocessor SoCs for specialized applications.

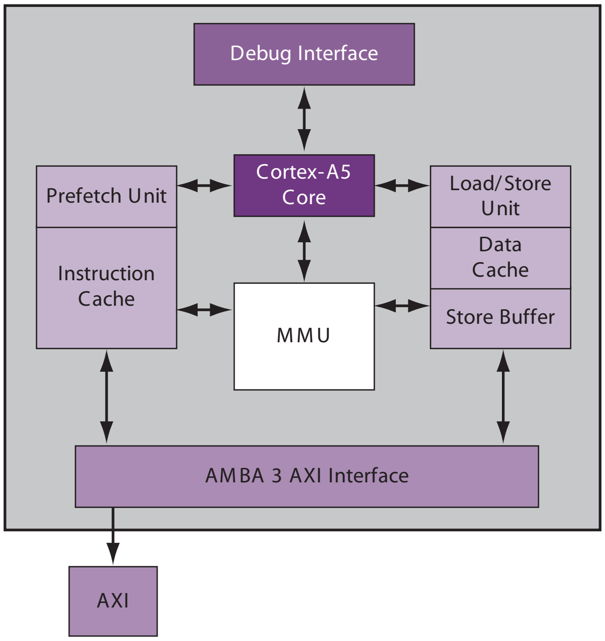

Figure 2. ARM Cortex-A5 block diagram. This processor bears a family resemblance to both the ARM1176JZ(F)-S and the newer ARM Cortex-A8. Featurewise, it's an upgrade over the ARM11 generation while maintaining binary software compatibility with the Cortex-A series.

Samsung has announced a Cortex-A5 license, and Atmel is reportedly another licensee. Neither company has publicly announced details of its future Cortex-A5 designs.

Like other processors in the Cortex-A family, the Cortex-A5 is based on the ARMv7-A instruction-set architecture (ISA). ARMv7-A includes the latest Thumb-2 instructions for better code density, plus the Thumb-2EE instructions associated with Jazelle. For backward compatibility, ARMv7-A supports the original Thumb instructions, too. (See *MPR 6/17/03-02*, "ARM Grows More Thumbs.")

Conceptually, the Cortex-A5 is a slower version of the Cortex-A8 or an enhanced version of the ARM1176JZ(F)-S. Figure 2 is a block diagram of the Cortex-A5. Feature for feature, it's almost identical to the Cortex-A8. Two differences stand out: the Cortex-A5 has a shorter integer pipeline, and it supports four-way symmetric multiprocessing.

Whereas the Cortex-A8 integer pipeline has 13 stages, the Cortex-A5 has eight, as Figure 3 shows. Shortening the pipeline reduces the maximum clock frequency but saves silicon and power. Still, an eight-stage pipeline is fairly deep for an embedded processor, so the Cortex-A5 is no laggard. When optimized for speed and fabricated in TSMC's 40nm-G process, it should reach 1.0GHz. In comparison, the deeper-pipelined Cortex-A8 can exceed 1.1GHz when optimized for speed and fabricated in TSMC's larger 65nm-G process.

Surpassing the ARM11

Although the Cortex-A8 is the Cortex-A5's next-of-kin, the ARM1176JZ(F)-S is a close cousin. ARM hopes the new core will lure developers away from the older one. Both have eight-stage pipelines, suggesting similar maximum clock speeds. (See *MPR 1/5/04-01*, "ARM Expands ARM11 Family.")

However, the Cortex-A5 is more efficient. It has dynamic branch prediction and can dual-issue a branch with another instruction. It executes 1.57 Dhrystone mips per megahertz, versus 1.2 Dhrystone mips per megahertz for the ARM1176JZ(F)-S. According to EEMBC's new CoreMark CPU benchmark, the Cortex-A5 is 12% faster than the ARM1176JZ(F)-S. (See *MPR 6/8/09-01*, "EEMBC's Dhrystone Killer.")

In ARM nomenclature, the "J" in ARM1176JZ(F)-S stands for Jazelle; the "Z" stands for TrustZone; the "(F)" indicates an optional FPU; and the "S" means "synthesizable." The Cortex-A5 dispenses with the alphabet soup but keeps the features.

Jazelle is ARM's technology for speeding up software written in programming languages that use run-time interpreters, just-in-time (JIT) compilers, or virtual machines. Java is the main target, but others are Perl, Python, and Microsoft's .NET languages. The original version of Jazelle, introduced in 2001, added instructions that accelerate Java bytecode execution. That part of Jazelle is now called Jazelle DBX (Direct Bytecode eXecution). It's the only version of

Jazelle in the ARM1176JZ(F)-S. (See *MPR 2/12/01-01*, "Java to Go: Part 1" through *MPR 6/4/01-01*, "Java to Go: Part 4.")

The Cortex-A5 has both Jazelle DBX and the newer Jazelle RCT (Run-time Compilation Target). Introduced in 2005, Jazelle RCT added a dozen new 16-bit instructions to the ARMv7 ISA and modified a few existing Thumb-2 instructions. Jazelle RCT also created a new execution state called ThumbEE for running these new Thumb-2EE instructions. (See *MPR 7/11/05-01*, "ARM Strengthens Java Compilers.") Bottom line: the Cortex-A5 is a slightly better engine for modern programming languages than the ARM1176JZ(F)-S or even the Cortex-A8, which lacks Jazelle DBX.

Both the Cortex-A5 and ARM1176JZ(F)-S support TrustZone, ARM's technology for protecting code that must remain secure—such as data encryption, authentication, or certificate management. It's a relatively simple solution. It adds one instruction, one configuration bit, one permission level, new control registers, and new security bits for the L1 cache. The new registers are secure versions of existing control registers (such as CP15). The new security bits mark each cache line as secure or nonsecure.

TrustZone supplements the existing user and privileged modes, allowing secure code to run as a monitored process alongside the operating system and application software. The Cortex-A5 implementation is unchanged from the ARM1176JZ(F)-S and is also identical to the Cortex-A8 and Cortex-A9 implementations. (See *MPR 8/25/03-01*, "ARM Dons Armor.")

Neon Lights and Floating Points

An FPU is optional for the ARM1176JZ(F)-S and remains optional for the Cortex-A5. However, the new core gets an upgrade. The Cortex-A5 works with ARM's latest FPU, VFPv3, instead of VFP11 for the older core. Both FPUs support single- and double-precision floating point, but VFPv3 has additional instructions.

Indeed, the Cortex-A5 implementation of VFPv3 adds a few more instructions, though it remains backward compatible with VFPv3 in other Cortex-A processors. One new instruction is a fused multiply-accumulate (FMAC). Additional changes bring VFPv3 into compliance with the latest IEEE 754-2008 standard. ARM hasn't decided yet whether the changes justify promoting the new VFPv3 to VFPv4.

The Cortex-A5 has another option unavailable for the ARM1176JZ(F)-S: ARM's Neon extensions. ARM introduced Neon in 2005 with the Cortex-A8. Neon's SIMD instructions and wide registers are useful for multimedia applications and light-duty signal processing. Neon is no substitute for a DSP in baseband processing, but it's sufficient for audio processing and similar tasks. (See *MPR 10/25/05-02* and *MPR 11/14/05-01*, "Cortex-A8: High Speed, Low Power.")

To save power and silicon, however, the Cortex-A5 uses a slightly different implementation of Neon than other Cortex-A processors. The SIMD registers and associated datapaths are only 64 bits wide, not 128 bits. Therefore, quadword operations must be double-pumped through the Neon unit, approximately halving their performance. Single-word and double-word operations should be unaffected.

Neon includes the VFPv3 FPU. Indeed, they share the same registers. SoC designs that don't need Neon can use the FPU alone, but Neon always includes the FPU. The trade-off for using Neon is that it enlarges the core logic by about 55% (excluding caches). Figure 4 shows a trial layout of the Cortex-A5 with Neon.

ARM says a Cortex-A5 with Neon and caches is actually smaller than an ARM1176JZ(F)-S with VFP11 when both processors are fabricated in the same process. And it uses less power than a similarly configured ARM926EJ-S. Specifically,

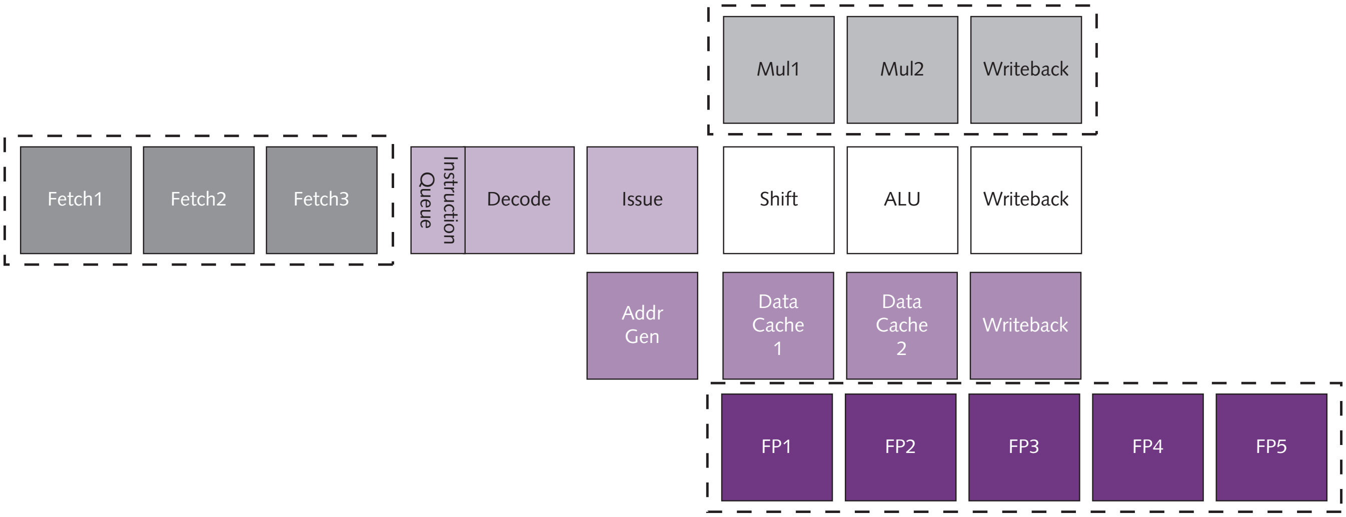

Figure 3. ARM Cortex-A5 pipelines. The basic integer pipeline is eight stages deep, including three prefetch stages that set up branch prediction. An instruction queue decouples the prefetch stages from the ALU section of the pipeline. A three-stage multiplier can execute 32- x 32-bit multiplies at a throughput rate of one instruction per cycle. Load/store instructions distribute the cache accesses over multiple stages to reduce critical timing paths. If the optional FPU or Neon extensions are present, a separate five-stage pipeline handles those operations.

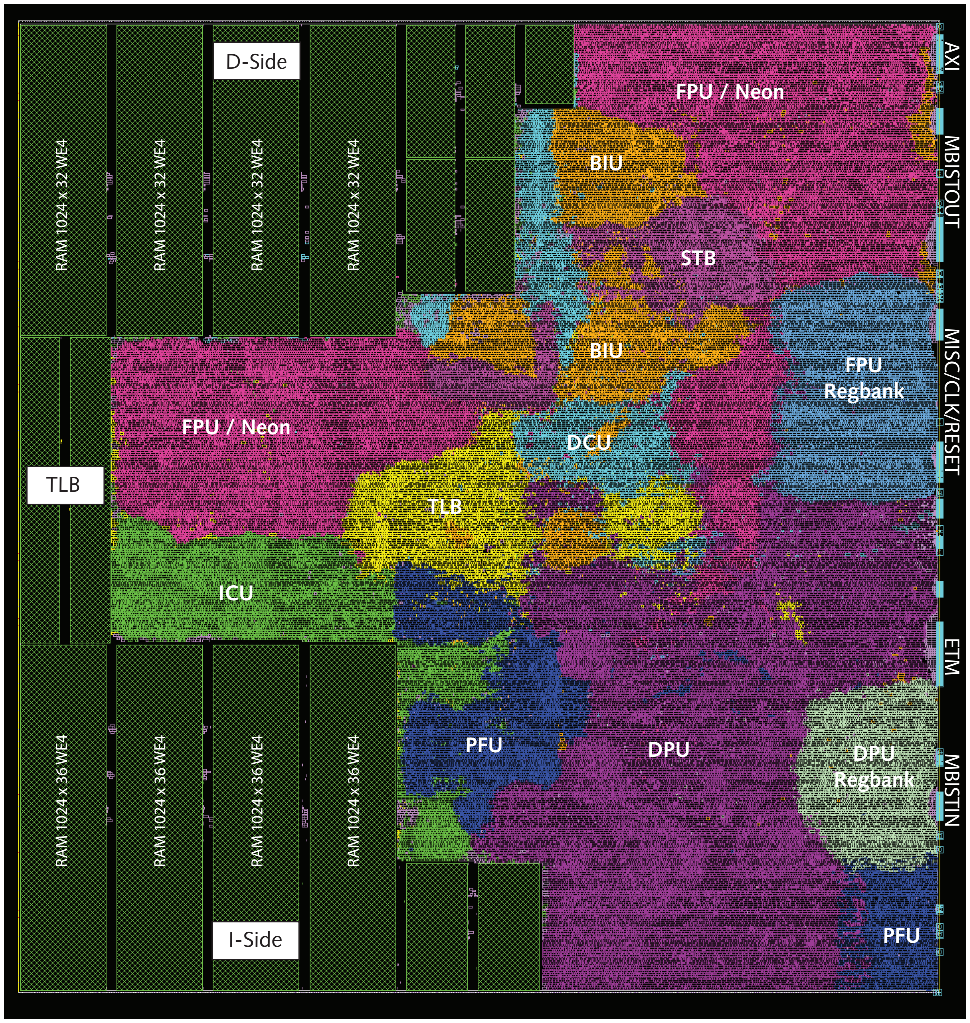

Figure 4. ARM Cortex-A5 trial layout. This configuration has 16KB instruction and data caches and Neon extensions. It targets TSMC's 40nm-LP process with ARM's Advantage 12T logic library, optimized for speed, with nominal V_t transistors (85% utilization). Even with modestly sized caches, the SRAM arrays occupy about one-third of the layout. Neon occupies about 22%.

the Cortex-A5 trial layout in Figure 4 occupies 0.68mm^2 in TSMC's 40nm-LP process. Without caches, it would occupy 0.42mm^2. Without caches or Neon, the core alone would occupy 0.27mm^2. ARM estimates the maximum clock frequency at 453MHz. Under simulation, power consumption when running Dhrystone 2.1 is 0.13mW per megahertz, or 59mW at the maximum clock speed. Energy efficiency is 12Dmips per milliwatt. Table 1 summarizes the differences among the Cortex-A5, Cortex-A8, ARM1176JZ(F)-S, and ARM926EJ-S.

It's remarkable that ARM was able to improve the features and throughput of the Cortex-A5 while reducing its size and power compared with two previous generations of processors. If only software programmers could do this.

AXI Bus Departs From ARM11

Another difference between the Cortex-A5 and ARM1176JZ(F)-S is the system I/O interface. The new core has a 64-bit AMBA-3 AXI bus interface with parallel read/write channels. Gone is the split AXI bus in the older core, which accessed instructions and data in separate memory regions—a classic Harvard architecture.

Instead, the Cortex-A5 adopts a "modified Harvard" bus architecture. Like a von Neumann architecture, it accesses instructions and data in the same memory space, but instructions and data have their own internal pathways and L1 caches. The core can still fetch instructions and load data at the same time, like a traditional Harvard machine. Other Cortex-A processors have modified Harvard architectures, too.

The ARM1176JZ(F)-S was the first core to implement AXI, which superseded the AMBA High-speed Bus (AHB). Since then, ARM has improved AXI to suit the requirements of faster memory and newer applications. ARM says the Cortex-A5 implementation of AXI delivers three times as much real throughput as the older implementation, as measured by a memcpy() operation running at the same bus frequency on both processors.

Of course, bus frequencies will vary, depending on the chip design. If a Cortex-A5 microcontroller drives the AXI bus at 166MHz, then DDR2 DRAM would effectively double the data rate to 333MHz. With a 64-bit AXI bus, the maximum theoretical bandwidth would be 2.6GB/s. (To reduce the cost of external memory, a microcontroller would probably use a 16- or 32-bit interface.)

The Cortex-A5 implementation of AXI is an even bigger improvement when compared with AHB, which is found on older processors like the ARM926EJ-S. Among other things, AXI can reorder outstanding reads and writes for greater efficiency. It can collect multiple memory accesses together and issue them in rapid-fire fashion, minimizing the latency of the first access in a burst.

Moving a chip design from an ARM9 to the Cortex-A5 will require some work on the bus interface, because peripherals must migrate from AHB to AXI. One compromise is to migrate only the fastest peripherals to AXI while retaining AHB for slower ones. Moving a design from an ARM11 to the Cortex-A5 also involves some work, even

Feature	ARM Cortex-A5	ARM ARM1176JZ(F)-S	ARM ARM926EJ-S	ARM Cortex-A8	ARM Cortex-M3
ARM ISA	ARMv7-A	ARMv6Z	ARMv5TEJ	ARMv7-A	ARMv7-M
Architecture Width	32 bits	32 bits	32 bits	32 bits	32 bits
Instruction Lengths	16 / 32 bits	16 / 32 bits	16 / 32 bits	16 / 32 bits	16 bits
Thumb Instructions	Thumb Thumb-2	Thumb	Thumb	Thumb Thumb-2	Thumb Thumb-2
Jazelle Extensions	DBX, RCT	DBX	DBX	RCT	—
TrustZone	Yes	Yes	—	Yes	—
Coherent Multiprocessing	Optional 1–4 cores	—	—	—	—
Integer Pipeline	8 stages	8 stages	5 stages	13 stages	3 stages
FPU, DSP	Optional Neon or VFPv3	Optional VFP11	Optional VFP9	Optional Neon or VFPv3	—
Memory Architecture	Modified Harvard	Harvard	Harvard	Modified Harvard	Harvard
L1 Cache	4K–64K (code) 4K–64K (data)	4K–64K (code) 4K–64K (data)	4K–128K (code) 4K–128K (data)	16K or 32K (code) 16K or 32K (data)	—
L2 Cache (External)	Optional 16K–8MB	—	—	Optional 128K–1MB	—
Tightly Coupled Memory (TCM)	—	1 or 2, 0–1MB (code) 1 or 2, 0–1MB (data)	0–1MB (code) 0–1MB (data)	—	0–1MB (code) 0–1MB (data)
Memory Management	MMU	MMU	MMU	MMU	Optional MPU
System Interface	AMBA-3 AXI 1 x 64 bits Opt. 2 x 64 bits MP	AMBA-3 AXI 2 x 64 bits	AMBA AHB 2 x 32 bits	AMBA-3 AXI 1 x 64 or 128 bits	AHB-Lite 2 x 32 bits
Real-Time Trace	CoreSight ETM	CoreSight ETM11	ETM9	CoreSight ETM	Optional ETM
Core Freq. (max) (IC Process)	480MHz–1.0GHz (40nm-LP, 40nm-G)	620MHz (90nm-G)	470MHz (90nm-G)	>1.1GHz (65nm-GP)	~270MHz (90nm-G)
Core Size (no cache) (IC Process)	0.27mm^{2} (40nm-LP)	1.95mm^{2} (90nm-G)	1.01mm^{2} (90nm-G)	< 3.0mm^{2} (65nm-GP)	33k–60k gates
Dhrystone 2.1	1.57Dmips / MHz	1.2Dmips / MHz	1.1Dmips / MHz	2.0Dmips / MHz	1.25Dmips / MHz
Power (IC Process)	0.12mW / MHz (40nm-LP)	0.8mW / MHz (90nm-G)	0.2mW / MHz (90nm-G)	< 0.45mW / MHz (65nm-GP)	0.047mW / MHz (90nm-G)
Introduction	2009	2004	2001	2005	2004

Table 1. Feature comparison of the Cortex-A5, Cortex-A8, ARM1176JZ(F)-S, and ARM926EJ-S. The Cortex-M3 isn’t really in this class, but we include it to show the difference in features between the Cortex-M and Cortex-A series, despite their power/performance similarities. It’s apparent from this comparison that the Cortex-A8 is a direct ancestor of the Cortex-A5. Another close relative is the Cortex-A9 MPCore (not shown), which uses the same multiprocessor extensions and ARM PL310 external L2 cache controller. Unfortunately, the differences in target fabrication processes make power/performance comparisons among these cores difficult. Even variations of the same process can make a big difference. ARM estimates 0.08mW per megahertz for a Cortex-A5 fabricated in TSMC’s 40nm-G process, and 0.12mW per megahertz in TSMC’s 40nm-LP process.

though both use AXI. Developers will have to adapt the ARM11’s classic Harvard bus to the Cortex-A5’s modified Harvard bus. That adaptation should be easier than migrating from AHB.

Multiple Cores Boost Performance

In a single-core configuration, the Cortex-A5 is a worthwhile feature upgrade from the ARM1176 and a big performance upgrade from the ARM926. When fabricated in the same process, the Cortex-A5 will run at about the same clock frequency as the ARM1176JZ(F)-S and about 50% faster than the ARM926EJ-S. The core will be about the same size as the ARM926EJ-S and smaller than the ARM1176JZ(F)-S, and it will use less power.

To make the Cortex-A5 even more attractive, ARM offers the same coherent multiprocessing extensions introduced with the Cortex-A9 MPCore in 2008. Indeed, the extensions are fully compatible, so multicore software already running on the Cortex-A9 can run unmodified on the Cortex-A5. The two processors are so much alike that ARM booted a Windows CE binary on the Cortex-A5 by hacking the CPU-ID to masquerade as a Cortex-A9.

Developers can integrate two, three, or four Cortex-A5 processors in a multicore design. Figure 5 is a block diagram of a quad-core configuration. The multiprocessing extensions are thorough. Snoop logic maintains data coherency among the L1 caches of all the cores. An interrupt controller distributes interrupts to the appropriate core, which can respond individually, without interrupting other cores. A special slave bus-interface unit provides coherent access to external memory or to an optional L2 cache controller.

The MP bus interface supports an optional second AXI port—a likely necessity to avoid I/O bottlenecks in a multicore design. Like the first AXI port, it's 64 bits wide and has parallel read/write channels. Developers can use these interfaces in different ways. Both AXI ports can connect to the external L2 cache controller, or one port could connect to the memory system while the other connects to on-chip peripherals.

ARM has synthesized a trial layout of the Cortex-A5 in a dual-core configuration. This configuration omits Neon, the FPU, and a second AXI bus. It includes 16KB instruction and data caches for each core, plus 32 interrupt lines. It was optimized for speed using ARM's Advantage 12T logic library (using nominal V_t transistors throughout), and it targets TSMC's 40nm-LP process. The maximum estimated clock frequency is 433MHz. Total core area is 1.31mm^2, of which 0.75mm^2 is logic.

Although the Cortex-A5 is compatible with the Cortex-A9's multiprocessing scheme, it differs from the ARM11 MPCore that ARM introduced in 2004. The ARM11 MPCore is based on the ARM1176JZ(F)-S and supports up to four cores, but its multiprocessing extensions are less thorough. For instance, it doesn't support coherent I/O control for its twin AXI buses. Migrating a design from the ARM11 MPCore to the Cortex-A5 will require more work than migrating from the Cortex-A9.

MIPS Competes With Cortex-A9

At first glance, the Cortex-A5 invites comparisons with the MIPS32 1004K, another licensable processor core designed for coherent multiprocessing with up to four cores. However, MIPS Technologies is aiming for higher performance. ARM's Cortex-A9 MPCore is a better opponent for the powerful MIPS 1004K. The new Cortex-A5 is a smaller, lower-power processor, more suitable for mobile systems.

MIPS introduced the 1004K last year. (See *MPR 4/28/08-01*, "Multicore Multithreading With MIPS.") Target applications focus on mains-powered consumer electronics, a market in which MIPS is particularly strong, whereas ARM is stronger in mobile electronics. MIPS designed the 1004K for HDTVs; TV set-top boxes with HD and video-recorder functions; home networking appliances; and multifunction printers. The 1004K may also be suitable for some mobile computing devices with wireless Internet access, but the Cortex-A5 will have the advantage when top performance isn't needed.

What sets the 1004K apart is hardware multithreading. It's still the only licensable processor core to combine multithreading with coherent multiprocessing. Each core can manage two simultaneous threads, so a quad-core design is the virtual equivalent of eight cores. Instructions from two or more software processes can share the same pipeline at the same time, switching contexts in one clock cycle. Each context may be a lightweight program thread or a heavyweight task, such as an operating system or application program. (Intel refers to this technology as Hyper-Threading.) The 1004K inherits multithreading from its parent, the MIPS32 34K processor. (See *MPR 2/27/06-01*, "MIPS Threads the Needle.")

Although the Cortex-A9 MPCore lacks multithreading, it's capable of superscalar execution. It can decode two instructions per clock cycle and issue four instructions per cycle. Additionally, the Cortex-A9 MPCore can execute instructions out of order, with speculative execution. In contrast, the MIPS 1004K has a simpler uniscalar pipeline—though it's one stage deeper—and in-order execution, without speculation. Both processors

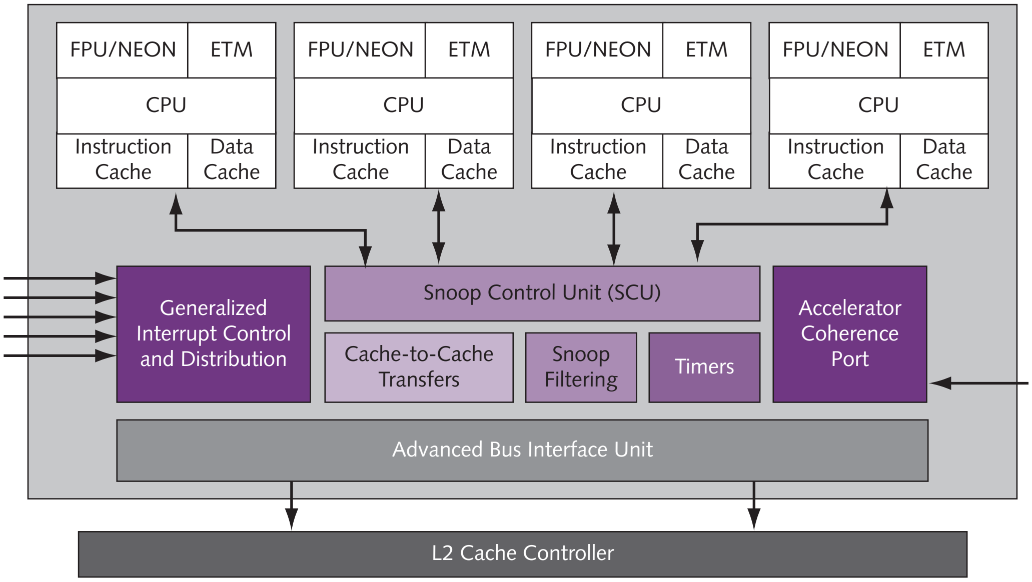

Figure 5. Cortex-A5 quad-core block diagram. The Cortex-A5 works with the same coherent multiprocessing extensions as the Cortex-A9 MPCore and is binary-compatible with Cortex-A9 software. Multicore SoCs can integrate two, three, or four processor cores. Optional features include a second 64-bit AXI bus and a shared external L2 cache.

Feature	ARM Cortex-A5	MIPS 1004Kc and 1004Kf	ARM ARM11 MPCore	ARM Cortex-A9 MPCore
CPU Architecture	ARMv7-A	MIPS32 R2	ARMv6	ARMv7-A
Architecture Width	32 bits	32 bits	32 bits	32 bits
Related CPU Core	Cortex-A8	MIPS32 34K	ARM1176JZ(F)-S	ARM11 MPCore
Symmetric Multiprocessing	2–4 cores (ARM MPCore)	2–4 cores (MIPS CPS)	2–4 cores (ARM11 MPCore)	2–4 cores (ARM MPCore)
Coherent On-Chip Bus	AMBA-3 AXI 1 or 2 x 64 bits	OCP 1 x 64 bits	AMBA-3 AXI 2 x 64 bits	AMBA-3 AXI 1 or 2 x 64 bits
Intercore Coherency	Yes	Yes	Yes	Yes
Coherent Cache Snoop	L1 + L2	L1 + L2	L1	L1 + L2
Global Int. Control	Yes	Yes	Yes	Yes
Coherent I/O Control	Yes	Yes	—	Yes
Coherent Debug	ARM CoreSight	MIPS PDtrace	ARM CoreSight	ARM CoreSight
Threads Per Core	1	1 ορ 2	1	1
Pipeline Depth (Integer)	8 stages	9 stages	8 stages	8 stages (9–11 clocks)
Superscalar Execution	Limited (Branch + ALU)	—	—	2-way decode, 4-way issue
Out-of-Order Execution	—	—	—	Yes, with speculation
Branch Prediction	Dynamic	Dynamic	Dynamic	Dynamic
L1 Cache (I / D)	4K–64K per core	0–64K per core	16–64K per core	16–64K per core
L2 Cache (Cache Controller)	Optional (ARM PL310)	Optional (MIPS SOC-it)	Optional (ARM L220 or PL310)	Optional (ARM PL310)
MMU	Yes	Yes	Yes	Yes
FPU	Optional, SP / DP (ARM VFPv3)	Optional, SP / DP (MIPS32 1004Kf)	Optional, SP / DP (ARM VFP11)	Optional, SP / DP (VFPv3)
16-Bit Instructions	Thumb-2	MIPS16e	Thumb	Thumb-2
DSP / SIMD Extensions	ARMv6 SIMD, optional ARM Neon	MIPS DSP ASE	ARMv6 SIMD	ARMv6 SIMD, optional ARM Neon
Java Extensions	Jazelle DBX + RCT	—	Optional Jazelle DBX	Jazelle RCT, optional DBX
Custom Extensions	—	MIPS CorExtend	—	—
Secure Execution Mode	ARM TrustZone	—	—	ARM TrustZone
Core Frequency (Max, Worst Case)	480MHz–1.0GHz (40nm-LP, 40nm-G)	800MHz (65nm-GP)	610MHz (90nm-G)	1.0GHz (65nm-GP)
Dhrystone 2.1 (per core)	1.57Dmips / MHz	1.56Dmips / MHz	1.25Dmips / MHz	2.5Dmips / MHz
Power (Typical)	0.12mW / MHz (40nm-LP)	~0.5mW / MHz (65nm-GP)	0.27mW / MHz (90nm-G)	n/a
Final RTL Availability	1Q10	June 2008	2004	April 2008

Table 2. Feature comparison of the ARM Cortex-A5, Cortex-A9 MPCore, ARM11 MPCore, and MIPS32 1004K processors. All are 32-bit licensable embedded-processor cores supporting coherent multiprocessing with up to four cores. The MIPS 1004K is the only one that combines hardware multithreading with multiprocessing. The Cortex-A9 has other advantages (superscalar execution, instruction reordering, speculation) that contribute to higher single-thread performance, as reflected in its greater Dhrystone mips per megahertz. The ARM11 MPCore is the "legacy" processor in this comparison, though still no slouch. Note: Mind the differences in process technologies when comparing these vendor-provided estimates of performance and power. (n/a: data not available.)

have dynamic branch prediction. The obvious question is whether the 1004K's multithreading offsets the Cortex-A9's instruction-juggling skills. In the absence of public benchmark results, we can only surmise that the answer depends on the workload. The Cortex-A9 probably has better single-thread performance, whereas the 1004K is probably better at multitasking.

In any case, this article is about the Cortex-A5. ARM envisions the new processor going into some of the same consumer-electronics products that MIPS targets, but

Price & Availability

The ARM Cortex-A5 is a licensable 32-bit embedded-processor core announced at ARM's TechCon^3 conference on October 21. General release for the uniprocessor RTL is scheduled for 4Q09, with multiprocessor RTL following in 1Q10. ARM doesn't publicly disclose licensing fees or royalties. For more information: *www.arm.com/products/CPUs/ARM-Cortex-A5.html*

probably as a coprocessor. For example, the Cortex-A5 might be the audio coprocessor in an HDTV or set-top box, while a more powerful processor like the MIPS 1004K handles the heavy-duty digital-video processing. Of course, ARM would like to see the Cortex-A9 MPCore challenging the 1004K and other MIPS processors for the center-stage role, but MIPS tends to top the bill in those applications.

Table 2 compares the ARM Cortex-A5, Cortex-A9 MPCore, and ARM11 MPCore with the MIPS32 1004K. Unfortunately, performance specifications are as hard to mine as unobtainium, especially when trying to make fair comparisons that minimize the variables of synthesis parameters, physical libraries, and process technologies. The important takeaway here is that SoC developers have an unprecedented choice of licensable embedded-processor cores for symmetric multiprocessing.

Not many years ago, a table with specifications like this would have been comparing high-end microprocessors for servers and workstations, not licensable embedded-processor cores. Yet all these cores are suitable, to a greater or lesser degree, for battery-powered mobile systems that fit into the palms of our hands. Clearly, the growing demands of digital media, mobile telephony, and wireless Internet connectivity are driving the evolution of embedded processors to new heights.

Conclusions

The Cortex-A5 is a useful addition to ARM's ever-growing product line. It's a lower-power gateway to the Cortex-A series, opening the door for compact mobile designs that must run sophisticated embedded operating systems and application programs. In a multicore configuration, the Cortex-A5 multiplies throughput while remaining economical. Early interest from Atmel and Samsung in the Cortex-A5 for microcontrollers indicates that a single-core configuration is attractive for low-cost, low-power devices.

Software compatibility with the Cortex-A8 and Cortex-A9 MPCore will be appreciated by developers eager to migrate their designs into lower-power realms. They will sacrifice some throughput, but not features. Existing designs based on the ARM1176JZ(F)-S and ARM11 MPCore will require a little more effort to port—on both the hardware and software sides—but the new core has much in common with the older cores. Developers looking to upgrade from the ARM926EJ-S will gain major features and performance while surrendering little or nothing in silicon area and power.