ARM10 Points to Set-Tops, Handhelds

New Core to Hit 300 MHz, Wrestles With Intel's StrongArm

by Linley Gwennap

EMBEDDED PROCESSOR FORUM

Pushing the ARM instruction set to new performance levels, the ARM10 will improve ARM's position in areas such as Windows CE computers and set-top boxes. At 300 MHz and 420 Dhrystone MIPS, the new core will deliver more than twice the performance of ARM9-based processors. The core can also be paired with a powerful new floating-point unit. At last month's Embedded Processor Forum, ARM10 designer Dave Jaggar described the new CPU, due to reach volume production in 2000.

The ARM10 will be the first design from Advanced RISC Machines that outperforms the original StrongArm (see MPR 11/13/95, p. 16). For most ARM licensees, the single-sourced StrongArm is a concept car they will never be able to drive. The ARM10 should finally provide an accessible alternative. By the time the ARM10 ships, Intel plans to deploy faster StrongArms, using its forthcoming SA2 core (see MPR 8/3/98, p. 11). But for the vast majority of ARM customers, particularly those who design their own ASICs, the ARM10 will be the new high end in the ARM lineup.

MICHAEL MUSTACCHI

ARM architect Dave Jaggar tells how ARM10 will enable new high-performance ARM applications.

Designed for High Clock Speeds

Taking a page from StrongArm, the ARM10 designers' first goal was to increase the clock speed as much as possible, as clock speed is the best lever for increasing performance. Although the ARM10 retains the same basic five-stage pipeline used in the ARM9 (see MPR 12/8/97, p. 10), nearly the entire pipeline has been reoptimized for faster operation. The claims are impressive: 300-MHz operation in a typical 0.25-micron process, 50% faster than the ARM9 in the same process. Since the ARM10 will not tape out for at least six months, we must wait to see if the actual parts meet these aggressive goals.

Taking advantage of the relatively roomy transistor budget of a 0.25-micron ASIC, Jaggar's team threw transistors at the problem of increasing clock speed. For example, the execute stage must perform a complete shift-and-add operation and then check if the result is zero, all in a single clock cycle. The ARM9 does these operations sequentially. The ARM10 checks if the result is zero even as the result is computed, by comparing the first operand with the two's complement of the second operand. Putting the zero check in parallel with the shift-and-add reduces cycle time.

Of course, all pipeline stages must be sped up to achieve the higher clock speed. The fetch and memory stages are the most difficult, as they are limited by the access time of the cache arrays. The ARM10 solves this problem with new dedicated adders that calculate the address in the previous stage, while the main adder is busy with other operations. This change spreads the cache access across one and a half cycles, enabling the faster clock speed. These design techniques are often used in high-power processors; the trick for ARM is to squeeze them into a tiny core with a 1-W power budget.

The Not-Quite-Superscalar Core

Performance is further improved with a few minor changes that reduce pipeline stalls. Whereas the ARM9 assumes that all branches are not taken, incurring a three-cycle penalty for taken branches, the ARM10 implements a simple static predictor (backward taken, forward not taken) that achieves an accuracy of about 65% in many cases. As before, the penalty for mispredicting a branch is three cycles, but the ARM10 will take this penalty less often than its predecessor does.

The ARM10 allows its multiple function units to operate in parallel, although it can still issue only one instruction per cycle. For example, instead of stalling during the two cycles it takes to perform an integer multiply, the ARM10 can execute a second instruction during this period (so long as it is not another integer multiply). The ARM10 is the first ARM processor with a nonblocking cache, allowing the CPU to continue executing instructions during a cache miss as long as the pending data is not immediately needed.

By operating its function units in parallel, the ARM10 can process several operations at once. For example, an ALU operation, an integer multiply, a cache miss, a cache hit, and a floating-point operation could all be executing in the same cycle. Previous ARM designs require each operation to complete before the next begins. Thus, the ARM10 gains some of the benefits of a superscalar design without the complexity of issuing multiple instructions per cycle.

Until now, the folks at ARM, Ltd. have avoided techniques such as branch prediction and parallel execution, which are popular in general-purpose processors, in their quest for moderate performance with the simplest possible design. To meet its performance goals, however, the ARM10 must move to a more sophisticated design. Because the new

core will appear only in processes of 0.25 micron or smaller, the extra logic needed for these functions has become vanishingly small.

New Instructions Create ARM v5

While making these changes to the ARM9 core, the design team thought to add a few new instructions as well, creating the ARM v5 architecture. The only new computational instruction is Count Leading Zeroes (CLZ), which locates the first 1 in a 32-bit value. This operation speeds normalization in soft-coded division routines, as ARM still refuses to add an integer division instruction. The CLZ instruction is also useful in some encryption and graphics algorithms.

As in the ARM9, the new core handles Thumb instructions (see MPR 3/27/95, p. 1) natively instead of with an add-on decoder. The ARM10 includes new instructions to allow ARM routines to call compressed Thumb-encoded subroutines (and vice versa) with a single instruction. In other ARM/Thumb chips, the only mode-switch instruction is a simple branch that does not save a return address. The company says customers are mixing ARM and Thumb code more finely than it originally expected; the new instructions simplify these finer mixtures.

The new instructions and, more important, the extra circuitry to improve clock speed, require more die area. The ARM10 integer core will be about 60% bigger than the ARM9, or about 4 mm^2 in a 0.25-micron four-layer-metal process. In the same process, the old ARM7 core consumes less than 2 mm^2.

Vector Unit Adds Floating-Point Capability

To complement the new core, ARM is developing a floating-point coprocessor called the VFP10. ARM's only previous FPU, in the 7500FE (see MPR 6/17/96, p. 16) offered limited performance and was rarely used; the new coprocessor is much more powerful. It has a large register file that can be addressed as 32 single-precision (SP) registers or 16 double-precision (DP) registers.

As Table 1 shows, the VFP10 provides a full range of floating-point operations. By setting a control field, the programmer can designate these instructions to operate on either scalar values (one register) or vectors consisting of multiple registers. Load Multiple (FLDM) and Store Multiple (FSTM) instructions can transfer similarly large quantities of data to or from memory.

The VFP10 can process only one operation per cycle, however, so the vector mode will not provide the same large performance boost as in the PowerPC G4 (see MPR 11/16/98, p. 17) and Intel's Katmai. By launching up to 16 operations with a single instruction, however, the single-issue core is freed to issue and execute integer and memory operations while the vector calculation completes.

The FLDM and FSTM instructions offer a similar advantage while improving bandwidth by using the chip's 64-bit data-cache bus to load two SP values (or one DP value) per cycle. They are also useful for saving and restoring state during a procedure call; a single FSTM can save a block of registers even as other instructions execute in parallel.

The floating-point unit can issue a single-precision MAC (multiply accumulate) every cycle, achieving a peak rate of 600 MFLOPS at 300 MHz. These instructions have a four-cycle latency. Double-precision values, however, require two passes through the multiplier, so DP MACs and multiplies issue every two cycles and have a five-cycle latency. All other DP calculations can be issued every cycle. Because the processor can load either two SP operands or one DP operand per cycle, in either case it has enough load bandwidth to match the issue rate of the multiplier.

The VFP10 uses ARM's coprocessor interface and is thus transparent to the programmer, who merely sees a new set of registers and new instructions that can be freely intermixed with standard integer ARM instructions. Thus, the VFP10 will be much easier to program than the logically separate and ultimately unpopular Piccolo DSP (see MPR 11/18/96, p. 17). Piccolo is also an integer DSP, whereas the VFP10 will handle more-demanding signal-processing needs.

The new floating-point unit is well suited to 3D graphics. A VFP10-enhanced ARM processor could be the heart of a set-top box, providing a 3D user interface and 3D games. Most ARM10 applications, however, don't need a floating-point unit and will dispense with the VFP10, which takes up as much die space as the entire integer core and also increases its power consumption.

ARM1020 Enables Powerful ASICs

Following its usual business model, ARM will provide the basic ARM10 core, with or without the VFP10 coprocessor, to licensees interested in embedding the core within their own ASIC designs. To assist in these designs, ARM will

Instr	Definition	Scalar	Vector	Instr	Definition	Scalar	Vector
FMAC	Multiply-Accumulate	▼	▲	FCMP	Compare	▼	
FMSC	Multiply-Subtract	▼	▲	FCMPZ	Compare With Zero	▼	
FNMAC	Negated Mul-Acc	▼	▲	FCVTDS	Convert Double to Single	▼	
FNMSC	Negated Mul-Sub	▼	▲	FCVTSD	Convert Single to Double	▼	
FMUL	Multiply	▼	▲	FITOF	Convert Integer to Float	▼	
FMNUL	Negated Multiply	▼	▲	FFTOI	Convert Float to Integer	▼	
FADD	Add	▼	▲	FMOV	Move to/from ARM Register	▼	
FSUB	Subtract	▼	▲		from/to VFP Register		
FDIV	Divide	▼	▲	FLDR	Load Single Value	▼	
FSQRT	Square Root	▼	▲	FSTR	Store Single Value	▼	
FCPY	Copy (Move)	▼	▲	FLDM	Load Multiple Values		▲
FABS	Absolute Value	▼	▲	FSTM	Store Multiple Values		▲
FNEG	Negate	▼	▲				

Table 1. ARM's VFP10 adds a new set of floating-point instructions. Many support a vector mode that operates on multiple sets of operands.

For More Information

ARM does not sell processors, instead licensing its designs to others. For more information on the ARM10, check the Web at *www.arm.com*.

also provide a larger macro that it calls the ARM1020T. As Figure 1 shows, the 1020T will combine the basic ARM10 integer core with a 32K instruction cache, a 32K data cache, a pair of MMUs, debug logic, and a bus interface.

At 300 MHz, the new processor's thirst for instructions and data will be far greater than an external memory can quench. The large on-die caches, eight times the size of the caches included in the ARM940, are needed to keep the pipeline flowing efficiently. Furthermore, the ARM10's target applications, including Windows CE, require large caches. Thus, ARM10 implementations will need plenty of die space. For this reason, system designers focused on low cost are likely to stay with the ARM9 or even the ARM7.

Both caches have 64-bit interfaces to the CPU core, providing high instruction and data bandwidth. To improve the hit rate and reduce thrashing, the caches are 64-way set- associative. The data cache can operate in write-back or write-through mode and, as mentioned previously, is nonblocking. Critical blocks of code or data can be locked down in the caches to enable deterministic performance.

The 1020T MMUs are, of course, fully compatible with Windows CE, enabling the ARM10 to be used in future derivatives of today's Palm PCs and Jupiter-class portables.

The external interface is a high-bandwidth version of ARM's AMBA bus. The new version doubles the data width to 64 bits. Taking a page from high-end processors such as Pentium II, the bus will support multiple masters and split transactions and operates at speeds of up to 150 MHz, resulting in a sustainable bandwidth of more than 1 Gbyte/s. To avoid CPU stalls, the 1020T includes an eight-entry write buffer for pending bus traffic.

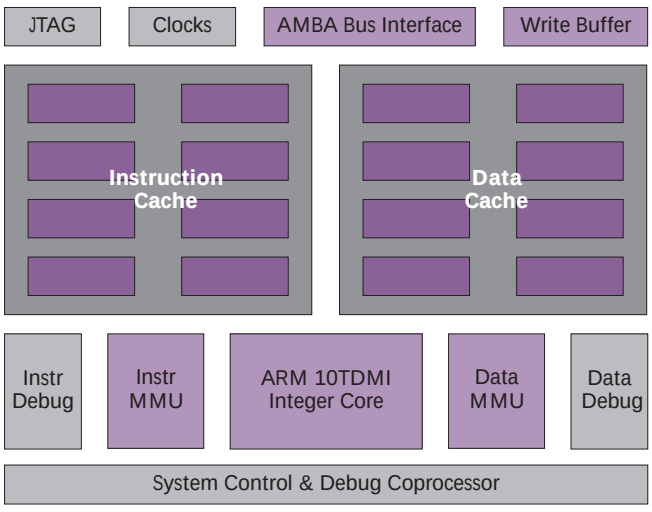

Figure 1. Floor plan of the ARM1020T shows that the dual 32K caches dominate the integer core. In a 0.25-micron four-layer-metal process, the 1020T is expected to consume 50 mm^2.

Typically, the AMBA bus will be used to connect only to other on-chip devices, simplifying the high-speed design. ASIC designers can use a simpler, slower bus to connect to off-chip memory and peripherals, if desired. This external bus can be 64 bits or less, but narrower buses may not provide enough bandwidth to keep the 300-MHz core fed.

ARM estimates the 1020T macro will consume a maximum of 1 W at 300 MHz, using a 2.0-V supply. To achieve a lower power rating, necessary for smaller portable devices, chip vendors could reduce the clock speed below 300 MHz, although this would reduce performance as well. Other vendors will quickly move the ARM10 to 0.18-micron processes, allowing clock speeds in excess of 300 MHz while dropping power dissipation to around 600 mW.

Jaggar expects the ARM10 will tape out in 2Q99, with early samples available in 3Q99. Allowing time for debugging these samples, we don't expect production parts to appear until early 2000. Custom ASICs using the ARM10 core probably won't reach production until 2H00. Given that tapeout is still several months away, ARM would not confirm which of its partners will be producing ARM10-based products, nor would it discuss any initial customers for these products.

StrongArm-2, SH-4 Will Challenge ARM10

The ARM10 represents a large step up for ARM and a powerful force in the industry. At 300 MHz and 420 MIPS, it would outrun any embedded processor shipping today. But the ARM10 isn't shipping today, and by 2000 many other vendors will have improved their performance.

In fact, Hitachi's SH-4 core (see MPR 10/28/96, p. 32), shipping in the SH7750 today, already comes close to the ARM10 in performance. At 200 MHz, the superscalar SH-4 is rated at 360 Dhrystone MIPS, although Hitachi uses the Dhrystone 1.1 metric while ARM adopts the more stringent Dhrystone 2.1 test. The SH-4 also includes a vector floating-point unit that, unlike ARM's, processes vectors in SIMD fashion, handling four single-precision FP MACs per cycle. This parallelism gives the Hitachi chip more than twice the FP performance of a 300-MHz VFP10.

The SH7750 will be used in Sega's Dreamcast videogame platform, due to appear later this year. Thus, while the ARM10 could be used for 3D gaming, its graphics are unlikely to be competitive with those of next-generation game platforms without extensive external assistance. The ARM chip is more likely to be used in set-top boxes that require some 3D graphics but not leading-edge game performance.

Another key market for the ARM10 will be WinCE-based computers. Today, these fall into two main categories: handheld Palm PCs and larger Jupiter-class subnotebooks that sport their own keyboards. The handhelds require CPUs that burn less than 500 mW of power, while the larger units can handle up to 1 W or so. Both types of devices are dominated by MIPS and SH processors today, although HP recently introduced the Jornada subnotebook with a StrongArm-1100 processor.

The 2.0-V version of the ARM10 should be faster than the current StrongArm and fit within the 1-W limit, allowing it to fit into larger WinCE boxes. To put the ARM10 into a handheld unit, designers must either crank down the clock speed or wait until the 0.18-micron version is available. The ARM10 could displace StrongArm in these portable computers, allowing other ARM chip makers, not just Intel, to play in this game.

Intel plans to make it difficult for ARM to gain the high ground by deploying its second-generation StrongArm, the SA2, by 2000. But Intel has been forced to deploy a new team on the SA2 (see MPR 10/26/98, p. 3), raising doubt about its ability to deliver a strong product on schedule.

ARM Seeks New Markets

ARM has its own credibility problems. When the company announced the ARM9 last year, it optimistically hoped that production would begin in 1Q98. Three quarters later, the first ARM9 parts are just entering production, and none has been announced for general availability. Most licensees continue to rely on the ARM7.

This low-cost multivendor core has made the ARM architecture nearly ubiquitous in cellular telephones, pagers, and similar equipment. But without faster parts, ARM has been thwarted in emerging high-performance markets such as handheld computers and set-top boxes. The fast but low-power StrongArm chips helped open doors, but these chips remain single sourced, and their transition from Digital to Intel has been rocky. ARM needs faster cores to avoid losing its substantial momentum.

The ARM9 and the ARM10 should fill this role. The ARM9 provides a large boost over the aging ARM7, and the ARM10 should exceed even StrongArm in performance. Although the ARM10 may not be the fastest embedded processor when it ships, it should be in the top tier, giving ARM access to a new range of high-end design wins. The ARM legions await the fulfillment of ARM's promises. M