M•Core M300 Gains Poise, Performance

Second-Generation Motorola Core Enhances Math Ability, Branch Handling

by Jim Turley

EMBEDDED PROCESSOR FORUM

Floating-point support and better branch handling are the two features that separate the latest generation of M•Core from its predecessor. At the recent Embedded Processor Forum, Motorola design manager Brian Branson pulled the wraps off the M300, the second generation of M•Core processors. Quite apart from its technical merits, M•Core appears to be on Motorola's fast track toward becoming the midrange embedded processor of choice, complementing PowerPC in the 32-bit segment of Motorola's vast array of processor architectures.

The M300 differs from today's M200-derived parts by having dual instruction fetch, optional support for single-precision floating-point arithmetic, and a simple but effective tweak that trims execution time for small inner loops. The first M300-based chips will enter design next month; Motorola won't say when the chips are due.

MICHAEL MUSTACCHI

At Embedded Processor Forum, Motorola's Brian Branson discusses the M•Core M300.

Branch Handling Helps Loops

First announced at Microprocessor Forum one year ago (see MPR 10/27/97, p. 12), M•Core is yet another 32-bit design from a company awash in CPU instruction sets. M•Core's position relative to the 68K, ColdFire, and PowerPC (among others) has been unclear. M•Core began at the low end of Motorola's high end, but its progress has been swift. At Embedded Processor Forum, Branson described a scalar M•Core running at 100 MHz with two new features: branch prediction and a simple FPU.

Like the ARM10 (see MPR 11/16/98, p. 14), the new M300 core assumes that forward branches are not taken, but that backward branches are taken. Given that most backward branches appear at the bottom of iterative loops, this is a safe bet most of the time. ARM cites 65% accuracy for this simple prediction, a statistic Motorola does not dispute.

Moreover, Branson said that about 13% of all instructions in the dynamic mix of a typical embedded program are taken branches. Of those, about 75% are short branches, with the target no more than 32 instructions away.

Armed with these statistics, the M•Core M300's designers implemented a simple but effective optimization for short backward branches. In microarchitectural simulations, the M300 trims about 7.5% off total execution time, compared with an otherwise identical M•Core implementation.

The M300's optimization goes a bit beyond just assuming backward branches will be taken. The M300 includes an address latch and an instruction latch, plus multiplexers on the address and instruction paths. The latches hold the address of the top of the loop and the first instruction of the fall-though path, respectively.

Using branch folding and the address latch, the M300 repeats loops with zero overhead. When the loop condition proves false, the instruction latch supplies the first instruction of the fall-through path, hiding some of the latency involved in prefetching the new instruction stream and following the correct flow of control.

When backward branches are taken, the M300 saves two cycles per iteration that would be lost to a pipeline bubble. Overall, this technique saves almost two cycles per loop iteration; Motorola quantifies the savings as $2m - 4$ cycles, where m is the number of iterations.

Motorola's implementation was quick and easy, but it is restricted to small loops. The branch target must be no more than 32 instructions (64 bytes) away, because the M300 compares only the lower six bits of the target address. It would have been trivial for Motorola's engineers to extend the scope of the loop optimization by designing a larger address comparator, but the team felt that the 6-bit comparator gave the best tradeoff of performance versus complexity. Motorola's simulations indicate that about 75% of loops fit within this size bracket. Since the branch folding saves only two cycles per iteration, optimizing larger loops that take hundreds of cycles to execute delivers diminishing returns.

Floating Point Comes to M•Core

During his presentation, Branson also unveiled M•Core's other big enhancement: floating-point arithmetic. Officially dubbed the M330, the new core is an M300 with a single-precision FPU, a first for the new chip family.

The M330's FPU implements a dozen new instructions, listed in Table 1. There are no special FP registers; single-precision operands are stored in the general-purpose registers. As Table 1 shows, the M330 provides only the most rudimentary FP operations: add, subtract, multiply, and divide. No geometric or transcendental functions (sine, tangent, etc.) are supported, and any floating-point exceptions cause faults that must be trapped and handled in software for IEEE compliance.

In spite of these limitations (or perhaps, because of them), the M330 executes FP instructions faster than the 68060. Floating-point multiplication, for example, finishes in about half the number of cycles needed by the '060. Clearly, the state of the art in FPU design has improved considerably at Motorola.

Programmers can choose to run the M330 in "real-time default" mode, if they prefer. In this mode, all floating-point underflows are truncated to zero and overflows set to infinity. It's not IEEE compliant, but it is faster and avoids exception processing.

The M330 improves integer performance as well. A new hardware multiplier helps 16 × 16-bit integer multiplication finish in one cycle; 32 × 32-bit multiplies require 1–2 cycles, because they may pass through the multiplier twice.

M•Core Poised to Overtake ColdFire

Motorola intends to use M•Core as the basis for many, if not most, future application-specific chips for industrial, automotive, and portable wireless devices. M•Core will supplant the 68K—and possibly ColdFire—as the CPU core of choice when PowerPC performance is not required.

The few integrated M•Core processors now in existence, such as the MMC2001 (see MPR 3/30/98, p. 13) and MMC2080 (see MPR 9/14/98, p. 14), have peripherals either designed from scratch or borrowed from previous designs and fitted with a "bus gasket" to adapt them for M•Core's internal bus. M•Core chips can also use a bus bridge for wholesale conversion from one internal bus to another. Motorola's minions are busily shifting many existing peripherals over to the M•Core bus, using both methods.

Branson didn't give a schedule for M300-based chips. He did say they would run at 100 MHz in a 0.25-micron, 2-V process, such as HiP-3 (see MPR 9/14/98, p. 1). Allowing for Motorola's characteristic conservatism, we estimate M300 chips may reach 150 MHz or better by the end of 1999. Newer processes will, of course, push that speed even higher.

This leaves ColdFire's position somewhat in doubt. Clearly the successor to 68K designs, and with some appeal for those with a need for partial software compatibility, ColdFire has rapidly (and unsurprisingly) overtaken the 68K in price/performance. ColdFire appeared to be the logical middleman between the 68K at the low end and PowerPC at the high end. It now appears that M•Core may be the heir apparent to Motorola's midrange.

Mnemonic	Description	Latency (cycles)	
		M330	68060
fsadd	FP Add	2	3–5
fssub	FP Subtract	2	3–5
fsmul	FP Multiply	2	3–5
fsmulr1	FP Multiply Into R1	2	n/a
fsdiv	FP Divide	17	37–39
fsmac	FP Multiply-Add	3	n/a
fsmsub	FP Multiply-Subtract	3	n/a
fsflt	FP Convert to Float	2	3–4
fsint	FP Convert to Integer	2	3–4
fsneg	FP Negate	1	1–3
fscmplt	FP Compare, Less Than	1	1–3
fscmpne	FP Compare, Not Equal	1	1–3

Table 1. The M300 M•Core processor core adds 12 new single-precision floating-point instructions. n/a = not applicable.

Price & Availability

Designs based on the M•Core M300 core will begin in December. No schedule for delivery of M300- or M330-based chips has been announced.

For more information, contact Motorola (Austin) at 800.521.6274 or visit *www.motorola.com/mcore.*

Motorola executives describe M•Core as "scalable" and capable of reaching speeds of 350–500 MHz "in the near future," while discussions of ColdFire are salted with terms like "legacy" and "compatible."

Both families have roadmaps that point to superscalar and "superpipelined" execution units in future generations, though the dates are fuzzy. M•Core has beaten ColdFire to the punch by adding floating point, which ColdFire may never get. Likewise, M•Core has spaces in its opcode map specifically for pseudo-DSP and other application-specific extensions to the base instruction set, although the company has yet to announce plans to fill them.

In the most recent of Motorola's reorganizations, the company spawned four groups focused on the automotive, wireless, networking, and entertainment markets. M•Core belongs to the automotive group; the other groups share joint custody of Motorola's other CPU and DSP families.

Automotive and wireless systems are better served by M•Core than by ColdFire. Home electronics are cost sensitive but not usually power sensitive, so ColdFire works well in that space, and PowerPC's place in Motorola's networking roadmap is well documented (see MPR 9/14/98, p. 12).

At this point, M•Core's youth handicaps it against ColdFire's legacy software support. But apart from habituating a sentimental attachment to 68K-derived source code, M•Core promises the developer some tangible technical advantages and, perhaps, performance. Motorola touts M•Core's superior power efficiency (a major reason behind its development) and high code density, and M•Core now appears to be on the fast track for future speed and architectural enhancements.

Symbian (*www.symbian.com*) is porting its operating system, putting M•Core second only to ARM in support for EPOC32. This makes M•Core a shoo-in for wireless systems (at least those from Motorola), and media or DSP extensions could make it a winner for consumer products as well. We expect a sizable portion of 68K users to gradually migrate to M•Core, confining ColdFire and PowerPC to legacy and performance applications, respectively. M