Mitsubishi Mixes Microprocessor, Memory

M32R/D Combines 32-Bit RISC Core with 2 Mbytes of On-Chip DRAM

by Jim Turley

Lending new meaning to the term "embedded microprocessor," Mitsubishi Electronics has developed a hybrid CPU/DRAM chip that places a 32-bit RISC core in the middle of a 16-Mbit DRAM. The combined CPU/DRAM makes an ideal controller for embedded applications that need to minimize PC-board real estate and power consumption.
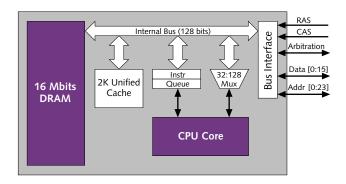
Although not the first chip to integrate a CPU and DRAM, Mitsubishi's M32R/D has by far the largest on-chip memory capacity of any microprocessor. It is truly both a microprocessor and a memory: its bus is bidirectional, initiating cycles to external peripherals and responding to memory requests from other processors or DMA controllers.

The chip, which is sampling now but won't enter production for another year, should appear first in new cellular "featurephones" and wireless organizers, where its space and power savings will be most valuable and where software compatibility is not a big issue. At 66 MHz, the CPU turns in a respectable 52 MIPS, based on Dhrystone 2.1; as a synchronous DRAM, it offers lackadaisical 120–180-ns access times.

Mitsubishi says its M32R/D is just the first of a planned assortment of CPU/DRAM hybrids. The company sees this mixture as a natural step toward reducing cost, power consumption, and package size as the market for battery-powered organizers, PDAs, and wireless communications products grows. While other vendors integrate peripherals with their CPU cores, Mitsubishi is addressing a much more basic demand for memory.

Integrated DRAM Cuts Memory Overhead

Mixing a CPU and a DRAM offers several advantages. If the application's memory requirements can be satisfied by the 2 Mbytes of on-chip RAM, the processor never has to go

Figure 1. Mitsubishi's unique M32R/D device combines a 16-Mbit synchronous DRAM with a 32-bit microprocessor core via an internal 128-bit bus, all running at 66 MHz.

off-chip for memory, greatly reducing power consumption. The combination also reduces both PCB area and pin count, important attributes for small-form-factor and high-volume products. Performance is enhanced by the low-latency access to on-chip RAM. And electromagnetic emissions are reduced because all signal transitions are kept within the device.

Power consumption is indeed quite low. In a worst-case scenario with all internal functions running and the external bus active, the M32R/D consumes nearly 700 mW from its 3.3-V supply. In standby mode (with the CPU stopped), the chip needs only 2 mW to refresh itself.

Typical consumption when executing from on-chip RAM and cache is only 270 mW; the typical power of a comparable 3.3-V CPU and discrete DRAM would be nearly one watt. Thus, the M32R/D yields a very impressive power/performance ratio, especially considering 16 million of the chip's 17 million transistors are devoted to DRAM and contribute only indirectly to its performance.

Another advantage of having the DRAM on chip is that it can be made very fast. From outside the package, the M32R/D appears like a 16.7-MHz synchronous DRAM, but internally, the memory array has a 128-bit path to the CPU. The core can access this memory in a single five-clock transaction, or 75 ns.

Any other microprocessor would be hard pressed to read from a commercial DRAM in that amount of time, regardless of bus speed. For example, using standard 70-ns page-mode DRAMs and allowing for logic overhead, setup time, and hold time, it would take 14 cycles (5-3-3-3) to complete the same four-word burst transaction at 66 MHz. Synchronous DRAMs could cut the total to 8 cycles (5-1-1-1) —still 60% slower than the M32R/D.

Cache Lessens Reliance on Internal DRAM

Although the M32R/D core is swimming in a sea of DRAM, the chip also has a cache. The 2K of direct-mapped unified instruction/data cache shares the 128-bit-wide bus with the CPU core, DRAM logic, and external bus interface, as shown in Figure 1. The CPU maintains an instruction queue with two 128-bit entries (typically 8–16 instructions); when one of the entries is empty, another 128-bit line is transferred from the cache in a single 66-MHz cycle. Cache misses filled from the on-chip DRAM take five cycles (75 ns) to load.

To access peripherals or additional memory, the chip relies on its 16-bit external data bus. Operating at only 16.7 MHz, the bus has a fairly long latency and low bandwidth. Clearly, fetching code across the external bus would seriously impact performance, but with 2M of RAM close at hand, many applications should rarely need to.

When the external bus is not busy, the DRAM is available to other masters in the system. Because the DRAM is shared, other potential masters must first arbitrate for the M32R/D's external bus. After arbitration, the access time for the first read is 2–3 clock cycles, or 120–180 ns. An internal buffer allows burst reads of up to 128 bits (eight transfers), with succeeding words available on successive clocks. Write cycles require only a single clock.

Basic Instruction Size Is 16 Bits

The M32R/D marks the debut of a new 32-bit architecture developed by Mitsubishi. The core sports many now familiar RISC-like attributes, such as an orthogonal 32-bit register set, single-cycle execution, and a five-stage pipeline.

The instruction set uses a mixture of 16- and 32-bit instructions. Most instructions are encoded in 16 bits, with long-displacement branches and large constants using a second 16-bit word. A two-operand addressing style with 16 registers keeps instruction words short. The CPU employs a basic load/store programming model with instructions operating primarily on 32-bit data. The chip's logical address space encompasses 32 bits as well, although only the lower 24 are available on the package.

The 86 integer instructions consist of the usual assortment of arithmetic, logical, and flow-control operations, including add, subtract, multiply, and stepped divide. Logical shifts can be to the right or left; arithmetic shifts are to the right. Loads and stores can use postincrement addressing, but only stores can predecrement. All instructions execute in a single cycle except loads, stores, and branches, which take three clocks, and division, which takes 32. Both PC-relative and absolute branches and jumps are implemented, with or without linking for subroutine calls.

Multiply-Accumulator Is Distinctive

The most distinctive part of the M32R/D's instruction set is its complement of four multiply-accumulate (MAC) instructions. Multiply and multiply-accumulate instructions work on either 16-bit or 32-bit values in the registers. For 16-bit operations, either the upper or lower half of the registers can be specified, as shown in Figure 2, through the MACHI and MACLO instructions, respectively.

In both cases, a 40-bit value is added to the product, with the 40-bit result deposited in a dedicated 64-bit accumulator. The entire $16 \times 16 + 40 \rightarrow 40$ operation completes in one clock. The chip can also do a $32 \times 16 + 56 \rightarrow 56$ MAC in a single cycle. Overflows are flagged and can be tested by a conditional-branch instruction afterward.

The chip can perform a straightforward 32×32-bit multiply, but this instruction takes two passes through the chip's 16×32 multiplier and needs three clocks to complete.

The CPU also includes a pair of interesting single-cycle saturate-and-round instructions for dealing with extended-precision calculations. After any multiply or multiply-accumulate instruction, the result in the accumulator can be rounded to a 16- or 32-bit signed value. When rounding to 32 bits, the most significant 16 bits of the accumulator are replaced with the sign bit while the least significant 16 bits are rounded off, leaving the rounded, saturated result in the middle of the accumulator.

By supporting 40 or 56 bits of internal precision for intermediate calculations until the result is rounded off, the M32R/D permits considerably more precise calculations than most 32-bit chips. Programmers can produce pseudo-FP functions simply by multiplying operands by several orders of magnitude and then dividing and rounding the results.

The single-cycle MAC unit takes up a good chunk of the M32R/D's die size, accounting for 4.1 mm^2 (about 40%) of the CPU core. Mitsubishi felt the bulky unit was worth the cost, given its goal of powering communications-oriented devices that need simple signal-processing capabilities.

DRAM Process Pressed into Service

Mitsubishi builds the M32R/D on its 0.45-micron two-layer-metal fab line in Japan, the same one it uses for conventional 16-Mbit DRAMs. While the process is obviously ideal for fabricating DRAM cells, as other vendors have discovered, it can be challenging for logic designs.

Although the transistor size is just 0.45 microns, the two metal layers are wider than those of logic-oriented half-micron processes. Most modern logic processes also have three or four metal layers, which helps increase gate density and reduce propagation delays. The Mitsubishi DRAM process is optimized to reduce leakage current, not to promote fast gate switching. The company readily admits that in a different 0.5-micron process, the M32R/D's processor core could reach 100 MHz.

The core measures 5.7 mm^2, not counting the hardware MAC unit, cache, tags, or control logic. That dimension is
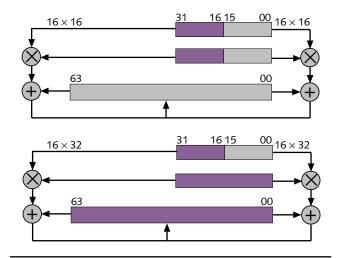

Figure 2. The M32R/D can execute four kinds of multiply-accumulate (MAC) instructions with its 16×32 MAC unit, taking either half of a 32-bit register as the multiplicand. All MAC instructions execute in a single cycle at 66 MHz.
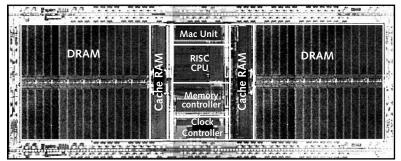
substantially (though perhaps not significantly) larger than an ARM7 core, which takes up 3.9 mm^2 in a 0.7-micron two-metal process, or Hitachi's SH-3 core at just 3.1 mm^2 in its 0.5-micron process. Adding the cache control, MAC, and bus interface bulks up the logic portion of the chip further. Overall, nearly 20% of the M32R/D's 154-mm^2 die is devoted to logic rather than memory, as Figure 3 shows.

The MDR Cost Model produces an estimated manufacturing cost of $24 for the M32R/D, considerably more expensive than other microprocessors in its performance range—which is hardly surprising, considering what the M32R/D includes. Compared with the cost of a conventional 16-Mbit DRAM, the chip's build cost is quite reasonable.

The question for designers, of course, is whether the M32R/D's price, at $80, is competitive at the system level. At current prices, 16-Mbit SDRAMs list for about $32–$36, depending on clock rates (which, at 66 MHz and up, are considerably faster than Mitsubishi's 16-MHz hybrid). An equivalent 16-MHz SDRAM might sell for about $30 in volume—making the processor component of the M32R/D worth about $50.

At $50 for a low-power processor that delivers 50 MIPS, the M32R/D is competitive with some of the newer RISC-like designs, neither a great bargain nor a big extravagance. By 2Q97, both DRAM and CPU prices will fall, making Mitsubishi's $80 price less competitive.

If price is removed from the equation, the M32R/D offers big advantages in both PCB space and power consumption. That combination will make the chip appealing to makers of pocket organizers, portable telephones, digital cameras, and network terminals. In all these cases, the

M32R/D's total lack of an installed software base should not be a major drawback to its adoption.

A Different Kind of Integration

Mitsubishi hopes the M32R/D and follow-on products like it will appeal to new embedded developers who want to mix performance with long battery life and convenient packaging. Other microprocessor vendors are eyeing these same applications, but with chips that integrate LCD controllers, serial ports, and touch-screen interfaces instead of memory.

Mitsubishi's chip heralds the first move of the DRAM vendors into higher-margin business. After years of reaping fabulous profits during the PC boom, DRAM makers are now looking for something new to pad their fab lines and their bottom lines. Mixing memory and microprocessors is a natural, and it's much easier for an entrenched DRAM manufacturer to add a little logic than it is for a CPU vendor to develop a competitive DRAM process.

The manufacturing challenges are still formidable, which is why it's taken so long for a major vendor to take the plunge. But as long as the bulk of the die is still DRAM, embedded applications with relatively small code requirements and moderate performance needs will be well served by this combination.

This trend spells good news for vendors like NEC, Toshiba, and Hitachi that already hold all the pieces they need. It's also an opportunity for small startups with a new idea to partner with the growing number of foundry services. It's less welcome news to traditional logic vendors like Intel, AMD, Motorola, National, and others that are constrained to sell their microprocessors "naked."

DRAM is certainly a necessity in most systems, and most embedded designers treat it as a commodity, so they often don't care much whose DRAM they use. But the choice of CPU architecture is something else entirely. On that front, Mitsubishi will have a tough time promoting its new architecture and instruction set. The company plans to roll out an assembler, C compiler, TRON operating system, and emulator by midyear under its own name. A high-level Verilog description of the core will be available in 3Q96, an obvious nod to potential ASIC clients.

There's an opportunity here for some enterprising company to merge a large DRAM array with a better-established microprocessor core like MIPS, ARM, or SPARC. The combination would provide all the tangible advantages of Mitsubishi's unique chip with the development-tool infrastructure that many developers want.

More hybrid devices with bigger and faster memory arrays are certainly in the offing, following each new DRAM generation by 12 months or so. By that reckoning, 32-bit CPUs with 64-Mbit DRAMs should appear in 1997 and provide true single-chip systems. These chips will, in turn, enable another wave of small, portable surprises. ◩

Figure 3. Mitsubishi's unusual M32R/D embeds a 32-bit CPU core in the middle of a 16-Mbit DRAM device. The die measures 154 mm^2 overall in Mitsubishi's 0.45-micron two-layer-metal process. The CPU core accounts for less than 6 mm^2 of that total.