Jade Enriches MIPS Embedded Family

First Synthesizable Cores From MIPS Implement New 32-Bit Architecture

by Tom R. Halfhill

Having completed its spin-off from SGI, MIPS Technologies is now bringing its microprocessor architectures into line with its business strategy: the single-minded pursuit of the embedded market.

At the recent Embedded Processor Forum, Engineering Director David Courtright of MIPS Technologies unveiled two new architectures that will carry the Rx000 family toward the future of high-performance embedded cores and system-on-a-chip devices. The new architectures, known as MIPS32 and MIPS64, are 32- and 64-bit derivatives of existing MIPS architectures. Courtright also described the first two cores based on MIPS32: the 4Kc and the 4Kp, popularly known as Jade and Jade Lite.

EMBEDDED PROCESSOR FORUM

Essentially, MIPS is overhauling its technology to reflect the company's change of focus since parting ways with SGI last year. No longer is MIPS a purveyor of high-end RISC processors for state-of-the-art graphics workstations and servers. Although a residual team of engineers at SGI continues to design high-end CPUs (mainly by stretching the R10000 core), MIPS is concentrating all its efforts on the embedded potential of its processors.

This strategy should not be construed as a fallback position. In 1998, the only 32-bit embedded architecture that outshipped MIPS was the undying Motorola 68K family. By fine-tuning its technology for the demands of new embedded applications, MIPS is attempting to capitalize on that success.

MIPS32 and MIPS64 define two broad families of embedded processor cores. They still emphasize high performance, but they do so within the boundaries of lower power, lower cost, and greater configurability. They are the first CPU cores that MIPS will deliver as synthesizable soft cores as well as the traditional hard macros. All this will give ASIC designers, embedded-system developers, and MIPS's semiconductor partners more freedom to customize the cores and integrate them with custom logic, memory, and on-chip peripherals.

The new architectures should also strengthen the company's position against competitors such as ARC Cores (see MPR 5/31/99, p. 16) and Tensilica (see MPR 3/8/99, p. 12), which already offer configurable soft cores. Although the Jade cores are much less configurable, they're an important step in the right direction for MIPS. They also provide additional evidence that the industry is pushing chip design and integration further down the chain toward embedded developers.

Mnemonic	Description	Notes
CLO	Count leading ones	New in MIPS32
CLZ	Count leading zeroes	New in MIPS32
MADD	Multiply-add (MAC)	New in MIPS32
MADDU	Multiply-add unsigned	New in MIPS32
MSUB	Multiply-subtract	New in MIPS32
MSUBU	Multiply-subtract unsigned	New in MIPS32
MUL	Multiply with register write	New in MIPS32
WAIT	Wait for interrupts	New in MIPS32
CACHE	Cache operation	Kernel mode
ERET	Return from exception	Kernel mode
TLBWI	Write indexed TLB entry	Kernel mode
TLBWR	Write random TLB entry	Kernel mode
TLBP	Probe TLB for matching entry	Kernel mode
TLBR	Read index for TLB entry	Kernel mode
MOVN	Move conditional on not zero	MIPS-IV
MOVZ	Move conditional on zero	MIPS-IV
PREF	Prefetch	MIPS-IV
DERET	Return from debug exception	EJTAG operation
SDBBP	Software debug breakpoint	EJTAG operation

Table 1. MIPS32 includes the entire MIPS-II instruction set plus these instructions. Some instructions are from the MIPS-IV instruction set, while others are completely new to MIPS32.

What's New in MIPS32

MIPS32 and MIPS64 combine elements of the MIPS-I, MIPS-II, MIPS-III, and MIPS-IV instruction sets with extensions developed by MIPS' partners and some new features tailored for embedded applications.

MIPS32 starts with the complete MIPS-II instruction set and, as Table 1 shows, adds 19 new instructions. These include cache prefetches, conditional moves, multiply-accumulate, multiply-subtract, targeted multiply (MUL), a wait instruction for power management, and instructions that count leading zeroes and ones (CLZ and CLO).

Some of these instructions were developed by MIPS' partners and have now become official parts of the MIPS32 architecture. They're largely self-explanatory, except perhaps for CLZ, CLO, and MUL. Instructions that count leading zeroes and ones are useful for normalization functions, floating-point emulation, and cryptographic algorithms, among other things. MUL is useful when a program doesn't need the higher half of the result of a 32-bit or smaller multiply operation. It stores the lower half of the result directly in a general-purpose register (GPR), thus saving an instruction to move the result from the special multiply LO register and allowing fully pipelined multiplies.

MIPS32 also has better exception handling, an MMU interface, a standardized kernel mode, new configuration registers, timer registers for real-time interrupts, and an enhanced JTAG debug interface.

Exception handling in MIPS32 resembles the method used by the R4000 and R5000, and it's a good example of

how MIPS tweaked the architecture for better embedded performance. Earlier MIPS processors had special entry points in the exception-handling logic for resets and TLB misses, but all other types of exceptions had to pass through a general exception vector (essentially, a long switch statement) that figured out whether the exception was an interrupt or not. This wasted a few cycles when the CPU handled an interrupt—generally not a problem for workstations and servers, but it is an issue for embedded real-time operating systems. In MIPS32, interrupts have their own vector into the exception-handling logic, shortening interrupt latencies.

Another OS-friendly feature in MIPS32 is a standardized kernel (privileged) operating mode. Previously, the kernel-mode interface varied from one MIPS generation to the next, so OS vendors had to keep revising their system software. Only the user mode for application software was upwardly compatible across the MIPS-I through -IV generations. MIPS32 and MIPS64 standardize the kernel-mode interface, so operating systems will be compatible with future cores.

Likewise, MIPS32 makes it easier for operating systems and other kernel-mode processes to recognize different core implementations. Currently, a program can check a special register to identify a chip, but it still has to match the ID against a table of known CPUs to discover the chip's specific features, such as the TLB size or cache addressing. MIPS32 defines new configuration registers that readily expose this information to kernel-mode processes—so, for example, an OS can immediately understand how to initialize the TLB or clear the cache tags.

Again, this is a seemingly small feature that wasn't too important when MIPS was producing a limited range of CPUs for workstations and servers. It's much more important for synthesizable cores that allow embedded designers to create a more varied range of chips.

Two Shades of Jade

MIPS has announced two Jade cores: the higher-end MIPS32 4Kc and the lower-cost MIPS32 4Kp (Jade Lite). Both are compatible with application software compiled for the R3000 and for the R4000 in 32-bit mode. Both cores can run most embedded operating systems, but only the 4Kc can run Windows CE, because the 4Kp lacks a TLB, as Figure 1 shows.

According to MIPS, Jade cores will run at clock frequencies of 100 to 150 MHz (worst case) or 150 to 225 MHz (typical) in a 0.25-micron process. The fully static cores occupy less than 3 mm^2 (4Kc) or 2 mm^2 (4Kp) of die area, not including caches. With 8K instruction and data caches, MIPS says the die size will be less than 10 mm^2.

Projected power consumption for the 4Kc at 2.5 V is 1 mW/MHz, or about 300 mW with 16K of cache when clocked at 150 MHz. Projected performance at that frequency is 180 Dhrystone 2.1 MIPS (600 MIPS/W). Naturally, these numbers will vary depending on the designer's choice of macro libraries, SRAMs, and so forth.

MIPS has the 4Kc core running in an FPGA now and expects TI (a MIPS partner) to tape out the hard core in June. The 4Kp core is scheduled for tapeout in 3Q99.

Less information is available about MIPS64 implementations. Future 64-bit cores are known as the MIPS64 5K (Opal) and the MIPS64 20K (Ruby). MIPS will offer the Opal cores in hard and soft versions but will deliver Ruby only as a hard macro. Ruby's performance target of at least 1,000 MIPS requires finer tuning than soft-core synthesis tools allow.

Both 64-bit cores will have additional features from the MIPS-IV architecture that was introduced with the R10000 in 1994. They could also include the MDMX (Mad Max) multimedia extensions (see MPR 11/18/96, p. 24). The first MIPS64 cores are scheduled to debut in 2000 and 2001.

It's also likely that the MIPS64 cores will implement floating-point instructions, including some of those defined in the latest MIPS-V instruction set. The Jade cores, in contrast, are strictly fixed-point devices. That's not a limitation of the MIPS32 architecture, however. In fact, MIPS32 has better support for floating-point operations than the MIPS-II architecture on which it's based. MIPS32 defines 32 floating-point registers and eight related condition codes, versus only 16 registers and one condition code in MIPS-II.

One oddity of the Jade cores is their lack of support for the MIPS16 compressed-code extensions. They can't execute 16-bit instructions at all. This is a strange omission, because the whole point of MIPS16 was to improve code density for embedded applications. According to MIPS, the new architectures don't exclude MIPS16 on technical grounds, so it's possible that future cores could support the shorter instructions. Until then, MIPS32 cores won't match the code density of CISC chips, 16-bit RISC competitors like the Hitachi SuperH, or compressed-code alternatives like the PowerPC's CodePack.

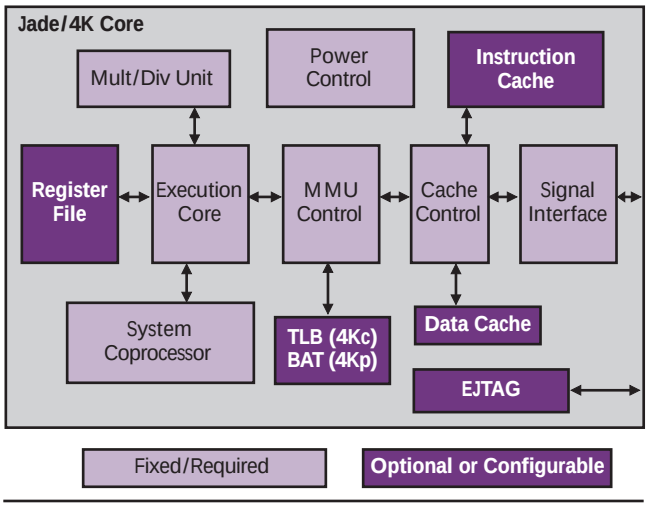

Figure 1. The Jade/4K soft core lets designers customize the register file, resize the caches, and eliminate the enhanced JTAG debugger unit. The 4Kc has a TLB, while the 4Kp has only a BAT.

Configuring the Core

As Figure 1 shows, the basic Jade core consists of several blocks that designers can modify or, in some cases, eliminate—including the caches, the MMU support, the register file, and the debug interface.

For most applications, MIPS recommends an 8K instruction cache and an 8K data cache. But numerous variations are possible. Designers can choose from direct mapping or two-, three-, or four-way set-associativity. Sets may consist of 1K, 2K, or 4K blocks of 16-byte lines. (A "0K" option omits the cache altogether.) The caches use a write-through policy, and each cache line is lockable. Cache misses stall the pipeline only until the first critical word becomes available.

A four-way set-associative cache with 4K blocks would yield the maximum cache size of 16K. The minimum cache size is 1K, direct-mapped. The recommended 8K cache could be two- or four-way set-associative. Yet another alternative is to replace one set of a cache with on-chip scratchpad RAM or instruction ROM. Designers can modify the cache without affecting the cache-control block or execution units.

To manage memory, the 4Kc core has a 32-entry TLB on the MMU, while the 4Kp uses a simpler block-address translation (BAT) scheme. With the TLB, memory pages can range in size from 4K to 4M. The BAT requires fewer gates than a TLB and accounts in part for the 33% smaller die size of the 4Kp. (A more significant factor is the smaller multiplier unit, which works iteratively instead of pumping the multiply instruction through the execute stage in a single cycle.)

The enhanced JTAG interface is optional, and designers can also exercise some control over the implementation of the register file. The size of the file is fixed: 32 GPRs, all 32 bits wide. But the registers may consist of flip-flops or latches, and if they consist of flip-flops, designers can choose whether they want muxes or gated clocks in front of the flip-flops. Muxes are easier to implement with synthesis tools, but they'll consume a little more silicon area and burn a little more power. More sophisticated engineers might prefer flip-flops with gated clocks, while those who live on the edge will settle for nothing but latches.

Internally, the Jade cores are bus-agnostic. MIPS says none of the existing on-chip buses qualifies as an industry standard, so rather than choose sides, MIPS designed a generic signal interface that should work with virtually any bus. The interface offers numerous options, such as dynamically controlled wait states and bi-endian addressing. It has dual unidirectional data channels that support pipelined addressing, burst transactions, and single-cycle reads and writes.

MICHAEL MUSTACCHI

David Courtright of MIPS describes the Jade cores at the Embedded Processor Forum.

MIPS could no doubt influence on-chip bus standards by throwing its weight behind one of the likely contenders, such as the VSI Alliance, the Advanced Microcontroller Bus Architecture (AMBA), or the Virtual Component eXchange (VCX). But it appears that MIPS would rather dodge the issue for now—an approach that also preserves more options for its customers.

Redesigning the Pipeline

Perhaps the most difficult challenge of overhauling the Rx000 architecture was redesigning the pipeline to make it more suitable for soft cores. The five-stage scalar pipeline of the R3000 was a logical starting point for a MIPS32 core, but the R3000 architects had used extremely tight timings to squeeze out the last drop of performance.

As Figure 2 shows, the R3000 triggers several critical operations—such as reading the instruction cache, reading the register file, and looking up entries in the data TLB—on falling edges of the pipeline clock. It triggers reads from the data cache on a rising-edge clock signal, but the timing is so tight that a read-align operation has to finish in the same pipe stage. Finally, the R3000 splits its instruction-address calculations into two half-cycle operations that surround reads from the instruction cache.

Those kinds of tweaks are praiseworthy for a high-performance RISC processor aimed at a specific fabrication process. They're a nightmare for a soft core that must accommodate the inefficiencies of synthesis tools and the vagaries of generic processes at Fabs R Us. Without major changes, an

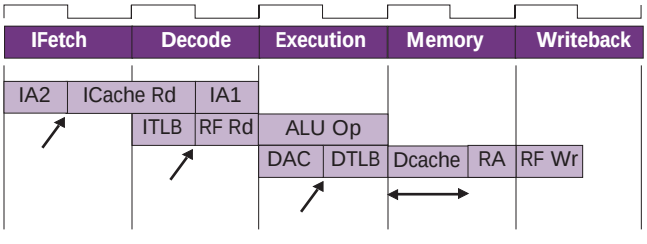

Figure 2. The original R3000 pipeline triggers too many critical operations on falling-clock edges and times some operations too tightly for synthesis tools. MIPS had to alter the pipeline for Jade.

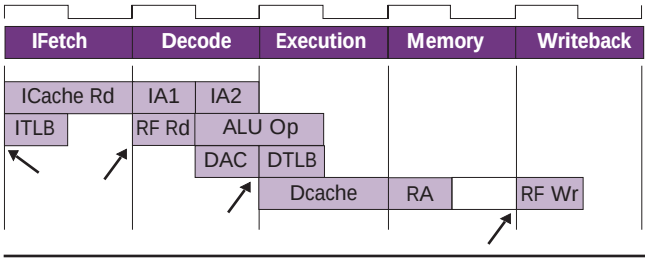

Figure 3. The Jade pipeline synchronizes critical operations on rising clock edges, unites instruction-address calculations in a single stage, and loosens the timing of data-cache accesses.

R3000-style pipeline would have been troublesome to synthesize, and it would have been difficult to use generator-based SRAMs for the caches.

To begin solving these problems, MIPS architects shifted all of the operations forward by one phase to synchronize them with the rising edges of the clock. They also moved accesses from the instruction TLB and the data cache to rising edges, which allows the read-align operation to finish in a separate stage from the cache access. Finally, they joined the instruction-address calculations into a single-stage operation that follows reads from the instruction cache.

These modifications eased most of the timing problems but inadvertently converted the five-stage pipeline into four stages. (As one engineer wryly observed, "Marketing was not amused.") Also, writes to the register file now triggered on a falling edge of the clock in the same stage as the read aligner. So the architects pushed the register writes backward by one phase, which synchronized them with a rising clock signal and restored the "lost" pipe stage.

As Figure 3 shows, the new pipeline has three more advantages. Cache accesses get most of a clock cycle to complete, which is better for slower, generator-based SRAMs. Because all of the critical operations now trigger on rising clock edges, it's easier to replace those synthesized blocks with custom logic, especially self-timed or dynamic logic. And register bypasses now trigger in the middle of clock cycles; this should give synthesis tools more leeway in balancing stage-logic delays.

In a final touch, MIPS created an entirely new multiplier that shares the front end of this five-stage pipeline with other instructions but has its own back-end stages, including a sixth stage for register writes. This unit can issue a 32×16 MUL or MADD instruction on every cycle and complete the instruction in the same amount of time as a load; it can issue a 32×32 MUL or MADD on every other cycle, with one extra cycle to complete. Only the 4Kc has this multiplier, however; the 4Kp uses the iterative multiplier instead.

Customers Need Customization

The spin-off from SGI and change of strategy are not the only reasons why MIPS is overhauling its architecture. Evidently, MIPS is also paying attention to market trends.

An amazing variety of embedded computing devices has appeared on the scene in recent years, and the pace is accelerating. Digital cameras are finally gaining a beachhead in the photography market; mobile phones are growing more sophisticated even as they shrink in size; handheld computers are succeeding where Apple's Newton failed; and faddish gizmos like Tamagotchi virtual pets, MP3 players, and obnoxious Furbys are catching on with kids.

To serve those markets, embedded designers need more components to choose from, or components that are easier to customize. They need hardware that's more malleable, like software. It's no surprise that MIPS is rethinking its architecture and is starting to license its intellectual property in the form of configurable soft cores.

Price & Availability

MIPS will offer the Jade/4K cores to licensees as hard macros or soft cores. Both include five models: a stand-alone instruction-level model, a bus-functional model, a system integration model (actually a combination of the first two models), a hardware/software Cosim model, and a cycle-accurate simulation model. MIPS has the 4Kc core running in an FPGA now and expects Texas Instruments to tape out the hard core in June. The 4Kp core is scheduled for tapeout in 3Q99. Prices vary, according to the licensing arrangement.

For more information, point your browser to *www.mips.com/products.*

Jade and future synthesizable cores based on MIPS32 and MIPS64 should give MIPS an advantage against ARM, which has been rapidly gaining market share in the past two years. ARM licensees, with the sole exception of Intel, are not allowed to modify the ARM microarchitecture.

But although the Jade cores are a significant first step, they don't go far enough. Soft cores that are almost fully compatible with MIPS processors are available from Lexra (see MPR 5/10/99, p. 5), and they have hooks that allow designers to add new instructions or custom coprocessors. Likewise, ARC Cores and Tensilica have soft cores that are much more configurable than Jade. ARC has a "configuration wizard" for Windows that guides designers through the steps of creating new instructions, adding function units, and modifying the register file. Tensilica has a similar tool that works over the Web, and its XTensa package includes an assembler that automatically recognizes new instructions. Another vendor, Triscend, is integrating field-programmable logic with an ARM7TDMI core in yet another approach to configurability. If this trend continues, pretty soon CompUSA will have to find shelf space next to Visual Basic for something like Visual Verilog.

Meanwhile, Nintendo's choice of a PowerPC processor instead of an Rx000 chip for its next-generation game console (see MPR 5/31/99, p. 5) is robbing MIPS of its biggest customer. The huge revenue generated by Nintendo has been like alimony during MIPS's divorce from SGI; in a few years, as Nintendo 64 sales decline, those payments will taper off. The game is not over for MIPS—two of the chips in Sony's next-generation game machine are based on MIPS cores (see MPR 4/19/99, p. 1). But it's a clear signal that MIPS needs to diversify its customer base, and the obvious way is to diversify its products.

Fortunately, MIPS is blessed with good technology, engineering talent, and a willingness to adapt. MIPS should regard Jade as the starting point for even more flexible cores in the future—and watch out for clever competitors who are quickly changing the rules of the game. M