

# M I C R O P R O C E S S O R

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

## WHITE PAPER

# THE MIPS32 24KE CORE FAMILY

*High-Performance RISC Cores With DSP Enhancements*

*By Chinh Tran, Engineering Manager, MIPS Technologies, Inc.  
Technical Contributors: Chijioke Anyanwu, Sanjai Balakrishnan,  
Anshul Bhargava, James Jiang, Radhika Thekkath {5/31/05-01}*

*(Editor's note: This is an edited version of the white paper that MIPS Technologies submitted with its presentation at Spring Processor Forum. *Microprocessor Report* has added a sidebar with some analysis of the new MIPS32 24KE processor family.)*

The MIPS32 24KE core family is the latest family of high-performance synthesizable microprocessor cores from MIPS Technologies, Inc. These cores feature DSP enhancements at a negligible cost. The 24KE core family fills an important gap in the convergence of RISC processors and digital-signal processors (DSP). This convergence is enabled by emerging market trends, and it offers cost advantages and technical challenges. The 24KE design meets these challenges quite handily, as this paper shows.

The 24KE cores were built by adding the new DSP enhancements to the existing pipeline in the 24K cores. The performance benefit for DSP applications from these enhancements can be as much as two times compared with a 24K design. And yet the 24Kec core has minimal frequency degradation and increase in die area compared with the 24Kc core. This paper discusses some of the novel implementation techniques used to achieve this result.

This paper is organized as follows. First, we describe the market trends promoting the convergence of RISC processors and DSPs. Then, we provide an overview of the 24K core family, since this is the foundation upon which the 24KE core family is built. This overview is followed by a brief description of the MIPS32 DSP Application-Specific Extension (ASE). This description leads to a discussion of

the design challenges, in which we focus on the base core of each family. The design challenges are described in the context of adding the DSP enhancements to the 24Kc core to create the 24KEc core. The paper concludes with a discussion of the achieved results and a summary.

### The Convergence of RISC and DSP

Embedded consumer devices require both general-purpose processing, such as running the operating system, and digital-signal processing, such as decoding music streams. Examples of such devices include set-top boxes, DVD players and recorders, voice-over-IP (VoIP) phones, MP3 players, mobile phones, and personal digital assistants (PDA). Add to this need for combined processing the fact that RISC processors can already execute some DSP operations, and the trend seems natural.

According to DSP analyst Will Strauss at Forward Concepts, "At least 30% of RISC processors are suitable for 'real' DSP applications."<sup>1</sup> Sample applications include audio codecs, packet-protocol processing, and JPEG processing. Furthermore, as RISC processors continue to increase their speeds, they are subsuming even more types of DSP applications. Therefore, this trend should continue, if not accelerate.

There are advantages and challenges to combining general-purpose processing and digital-signal processing on the same core. One big advantage is a more efficient hardware architecture.<sup>2</sup> The additional die area to implement DSP enhancements in the core can be kept small compared with the area for a programmable DSP. One reason is that the DSP enhancements can share much of the hardware already in the core. In addition, a combined architecture eliminates the external data bus between the processor core and the DSP. This can also reduce power consumption, because data is no longer driven back and forth through such an external bus.<sup>2</sup>

Another advantage is that software developers need to use only a single tool chain for writing both general-purpose code and DSP code. Eliminating the need to deal with multiple tool chains should lower development costs and accelerate development.

The advantages discussed above are significant, but three key challenges must be overcome to permit combining general-purpose and digital-signal processing on a single core. First, the core must deliver enough performance for real-time execution of DSP tasks. Second, the DSP enhancements must be added to a high-performance RISC core without adversely affecting the clock frequency, which would impact both general-purpose performance as well as DSP performance. And, third, the die-size increase of the enhanced core must be kept small. These challenges are the focus of this paper.

RISC processors have many benefits that enable them to achieve high performance. For example, caches and translation look-aside buffers (TLB) help to improve software performance. RISC instruction sets and pipelines are usually designed for relatively high clock frequencies. And, of course, RISC instruction sets lend themselves to good compilation.<sup>3</sup> A distinguishing feature among RISC architectures is the ability to run several different operating systems ported to that architecture. MIPS processors, such as those in the 24K core family, offer the advantage of being supported by operating systems commonly used in embedded systems.

DSPs usually offer special features, such as fractional arithmetic, saturation, and single-instruction multiple-data (SIMD) operations. They also often implement special

complex instructions to boost their digital processing performance. These instructions do not follow the RISC paradigm, because they perform expensive functions, such as operating directly on memory operands. Consequently, they create many critical paths that limit the clock frequencies of DSPs.<sup>2</sup>

A RISC core with DSP enhancements should inherit all the benefits of the RISC architecture. If it incorporates DSP support in a RISC-like manner, it will avoid the increased cycle time required to implement complex instructions. This will enable it to achieve a higher clock frequency and better code compilation than a DSP achieves. When compared with having a separate RISC processor and DSP, the combined core should have a smaller die area dedicated to DSP operations, due to the hardware reuse and elimination of the external bus discussed earlier.

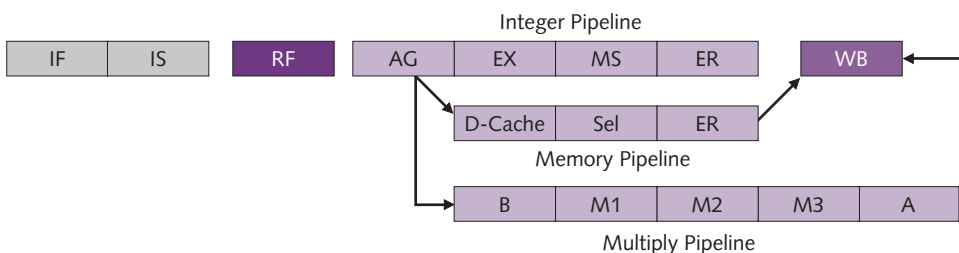
The rest of this paper focuses on the creation of such a RISC core with DSP enhancements. The 24Kc core is the base RISC core for this design. At Fall Processor Forum in 2004, MIPS Technologies introduced our DSP ASE, an architectural extension to enhance DSP performance with low implementation costs. Implementing this ASE in the base design results in the MIPS32 24KEc core.

### Basic Features of the MIPS32 24K Core Family

The MIPS32 24K core family includes the highest-performance 32-bit synthesizable cores available for the embedded market. The base core in this family, the 24Kc core, can achieve 400MHz to 625MHz in a 130nm CMOS process and can execute 576 to 900 Dhrystone mips. The 24K core family implements the MIPS32 Release 2 architecture, which includes vectored interrupts and shadow register sets to minimize and bound hardware-interrupt response times. These features support real-time applications.

An optional floating-point unit is available for the 24Kf core. An optional CorExtend interface is available on the Pro Series versions of the cores. The 24K cores also feature a single-issue pipeline, a decoupled 32×32-bit integer multiplier, and an aggressive memory subsystem. Figure 1 shows a diagram of the 24K's instruction pipeline.

The pipeline of the 24K core family is eight stages deep, logically divided into four sections. The fetch unit operates



**Figure 1.** The 24K core family has an eight-stage integer pipeline divided into four sections. Multiply instructions are separately pipelined, so they are nonblocking, and multiply-accumulate (MAC) instructions can repeat every clock cycle.

autonomously from the rest of the machine, decoupled by an eight-entry instruction buffer. The processor reads two instructions from the I-cache (instruction cache) each cycle, allowing fetches to proceed ahead of the execution pipeline. Speculation accuracy is improved with a branch-history table, holding 512 bimodal entries, and a four-entry return-prediction stack. A full cycle is allocated for the

I-cache RAM access, with the cache hit/miss determination and way selection occurring in the following stage.

One cycle is allocated to read operands from the register file and to collect other bypass sources. Separate execution pipelines handle integer and load/store instructions. For memory operations, address generation occurs in the AG stage. Next, the processor reads the data cache. Like the instruction cache, the D-cache (data cache) is four-way set-associative, may range in size from 16KB to 64KB, allows line locking, and supports optional parity protection. The MS stage performs hit calculation, way selection, and load alignment. The processor can accommodate up to four nonblocking load misses, allowing hits under misses.

Normal ALU operations pass through the AG stage and do their real work in the EX stage. This skewed ALU preserves the two-clock load-to-use relationship common to many other MIPS cores. The exception-recovery stage prioritizes any exceptions. Finally, the writeback stage updates the register file and other instruction destinations with new results.

Multiply instructions are nonblocking in the pipeline of the 24Kc core; subsequent instructions that do not depend on the result of the multiply can proceed without delay. Depending on the multiply instruction, its result is written into either the HI and LO accumulator or a general-purpose register (GPR).

Multiply instructions execute in a separate five-stage pipeline. Execution starts in the Booth recoding (B) stage,<sup>4,5</sup> which corresponds to the EX stage of the normal integer pipeline. Then, the operation propagates through the multiplier array during the M1, M2, and part of the M3 stage. The multiplier array produces a number in carry-save format.<sup>5,6</sup> A final addition to convert this number to two's-complement format is performed in the latter part of the M3 stage. The result is available at the beginning of the A stage, which is used to select between the multiply datapath and other sources for the accumulator value. The processor writes the new result into the accumulator at the clock edge between the A stage and the WB stage.

The multiply/divide unit (MDU) achieves a repeat rate of one multiply-accumulate instruction (MAC) every cycle. More details about multiplication follow later in this paper.

### MIPS32 DSP Extensions

The MIPS32 DSP ASE consists mainly of new instructions that execute in the integer and multiply pipes of the processor. The ASE includes all the typical DSP features, such as register-based SIMD instructions that can add, subtract, shift, or multiply. The SIMD instructions operate on as many as four operands simultaneously and support operands of 8, 16, and 32 bits.

The ASE also provides fractional arithmetic with saturation and rounding. There are several flavors of MACs, including dot-product-accumulate. Precision expansions and reductions enable scaling operations. Absolute, bit-reverse, and other instructions enable common DSP operations to perform efficiently. The ASE adds three new accumulators to the architecture, for a total of four.

In addition to all the commonly found operations, the ASE also adds some advanced features that attain extra performance without requiring a complex implementation. An example is a variable bit-extract method that efficiently extracts bits from an incoming stream. Another feature efficiently processes complex numbers. The ASE also includes a novel and efficient way to support virtual circular buffers. To minimize implementation cost, new state elements are limited to a DSP control register and the aforementioned three new accumulators.

### Design Challenges of Adding DSP to the 24K

Implementing the DSP ASE with negligible cost is the goal of the 24KE core family, which includes the 24KEc, 24KEf, 24KEc Pro, and 24KEf Pro cores. A customer can choose the appropriate core, depending on whether an optional FPU and/or an optional CorExtend interface is needed. To describe the design challenges for this core family, we will focus on the 24KEc core.

Specifically, the goal is to implement the DSP enhancements without significantly affecting the speed or die area of the core. There are two major challenges. First, the single-cycle ALU execution path already has critical timing. Inserting any additional logic in the datapath would increase the cycle time. Second, many additional features are required in the multiply datapath. Support for these features must be added without increasing the cycle time in any of the multiply pipeline stages. It's best to do this without adding another stage to the multiply pipeline. Also, with the wider datapath in the MDU, it's even more important to minimize the effect on the core's size.

The single-cycle execution of the ALU makes the timing of its datapath critical. The most critical paths are through the 32-bit adder and shifter. The diagram in Figure 2 shows the existing logic in the 24Kc core and the new DSP logic (shaded in purple). The DSP enhancements require saturating the output of the adder and rounding the output of the shifter. Furthermore, the results from these and other DSP instructions require selecting from new sources, so we need to add more mux levels. However, the additional saturation and rounding logic, plus the additional mux levels, would significantly increase the cycle time.

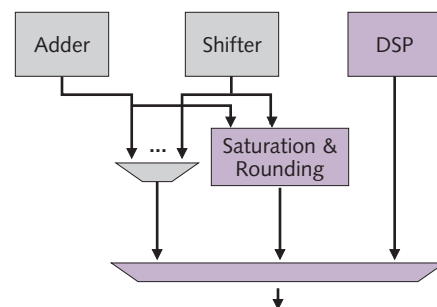
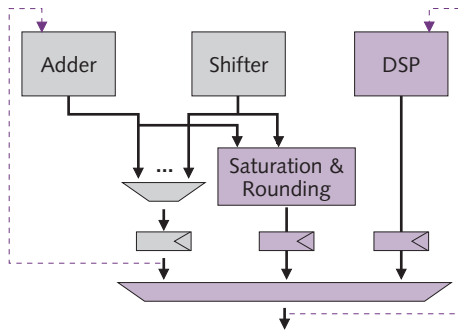


Figure 2. The execution stage datapath in the 24KEc core has critical timing.

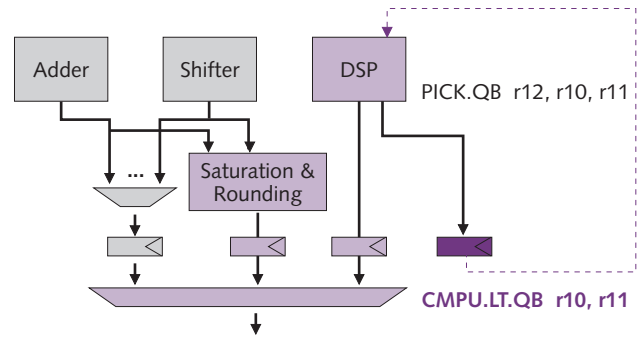


**Figure 3.** Forwarding results of DSP instructions from GPRs requires a second clock cycle.

To avoid significantly increasing the cycle time, we added a second cycle to select between the existing ALU results and new DSP sources. Note that existing ALU operations are not affected; their results are still forwarded to a dependent instruction for execution the next cycle. Therefore, if the first instruction of a back-to-back sequence is not a DSP instruction, result forwarding can still take place.

If the first instruction of a sequence is a DSP instruction, an additional cycle is required before forwarding the result, as Figure 3 shows. A good compiler or programmer can usually mask this one-cycle delay, because of the vectorized nature of DSP code. Another instruction, such as a data load, can often fill this delay slot between the two dependent instructions.

However, there are some DSP instruction sequences in which it would be useful to forward the result to the next instruction. A compare followed immediately by a pick is such an example. In Figure 4, the first instruction in such a sequence compares the corresponding bytes in register 10 and register 11. Based on the result of the comparison, it sets the condition-code bits in the DSP control register. In the next cycle, the second instruction advances to the execution stage. The instruction shown in Figure 4 picks each byte from GPR 10 or GPR 11, based on the corresponding condition-code bit. The result is four bytes, some of which may be from GPR 10 and some from GPR 11.



**Figure 4.** A DSP compare instruction sets the condition-code bits. Result forwarding to the next instruction without a delay is possible for useful DSP sequences.

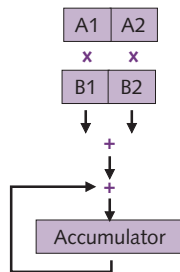
The pick instruction depends only on the condition-code bits, which can be forwarded from the previous cycle. Unlike DSP results in a GPR, the condition-code bits need not pass through the mux levels during the extra cycle. By using dedicated bits, the processor can execute useful sequences, such as compare and pick back-to-back, without delay.

By using such techniques, we were able to maintain a high clock frequency for the 24KEc core. We also minimized additional die area by reusing the ALU's 32-bit adder and shifter. We did so without affecting the existing ALU instructions while enabling result forwarding for useful back-to-back DSP instruction sequences.

Another design challenge was implementing the dot-product-accumulate instruction in the MDU. For this instruction, each half of a source GPR contains one Q15 operand. (Q15 is a 16-bit representation of a fractional number.) This instruction performs two simultaneous multiplications of the corresponding halves from the two source registers. The processor adds the two products to complete the dot-product operation. Then, the processor adds that result to the contents of the specified accumulator. As illustrated in Figure 5, this instruction performs two multiplications and two additions.

Moreover, the dot-product-accumulate instruction and other multiplies in the DSP ASE must support multiple data types and perform saturation. For optimal performance, it's desirable to maximize the repeat rate for as many types of MACs as possible.

$$\text{Acc} = \text{Acc} + (\text{A1} \times \text{B1}) + (\text{A2} \times \text{B2})$$



**Figure 5.** The dot-product-accumulate instruction performs two multiplications and two additions, storing the results in the specified accumulator.

### Modifying the Multiplier Array

The multiplier array and adder make up most of the multiply datapath. The multiplier array consists of rows of carry-save adders (CSA). These rows of CSA blocks add the partial products of the multiplication. Each CSA adds three bits to produce two bits, a carry and a save. In Figure 6, each CSA block represents enough individual CSAs to operate on its input partial products. The final row produces two carry-save numbers, which must be combined using a 64-bit adder to convert them into the final two's-complement result.<sup>5,6</sup> (The figure doesn't show all the staging registers.)

The final two carry-save numbers are forwarded through staging registers to the next instruction, so that MACs can have a repeat rate of one per cycle. The datapath forwards the accumulated value in carry-save format before it propagates through the 64-bit adder.

To support dual-multiply operations in the DSP ASE, the multiplier array is configured as two halves. For a single 32-bit multiplication, both halves function as part of a common array. For dual-multiply operations, the left and right subarrays each yield a product in carry-save format. The dot-product-accumulate instruction requires these products to be added, then accumulated. However, other dual-multiply instructions in the ASE simply require these products as results. For those instructions, the products are available at the point indicated by the ovals in Figure 6. Not shown are the two 16-bit adders required to convert these carry-save values into two's-complement numbers. These adders are small relative to the rest of the multiply datapath and have no timing impact. They are omitted from the figure because they are not part of the multiplier array.

A row of muxes added to the left subarray supports the dot-product-accumulate instruction described earlier. These muxes can shift the bits from the left subarray so they have the same significance as the bits in the right subarray. This action effectively aligns the two products so they can be added. The elegance of this approach is that we extensively reuse the existing logic. We use the left subarray for the second multiply, since it is not needed for 16-bit multiplication. Also, later stages of the multiplier array are reused to perform the two additions for the dot-product-accumulate instruction. Therefore, the additional cost for this second addition is limited to a single level of muxes.

Saturation for the special case when both multiplier and multiplicand are  $-1$  is performed in parallel with the multiplier array. Another level of muxes is required to substitute the products with the maximum fractional value. Additional logic after the 64-bit adder handles saturation after accumulation. Multiply instructions that saturate after accumulation have a repeat rate of one every two cycles, because final saturation occurs after the accumulated value would have been forwarded to the next instruction. All other MACs have the best repeat rate possible—one per cycle.

Note our high reuse of existing hardware, as indicated by the large ratio of existing logic to new logic (purple) in Figure 7. We avoided adding any additional CSA levels, and the added logic was small and introduced little delay. In the dot-product-accumulate instruction example, performing a dual-multiply, aligning the products, and adding them together required only one additional level of muxes in the multiplier array.

Our approach to redesigning the MDU allowed us to maintain a high clock frequency with the 24KEC core. We minimized additional die area by reusing the

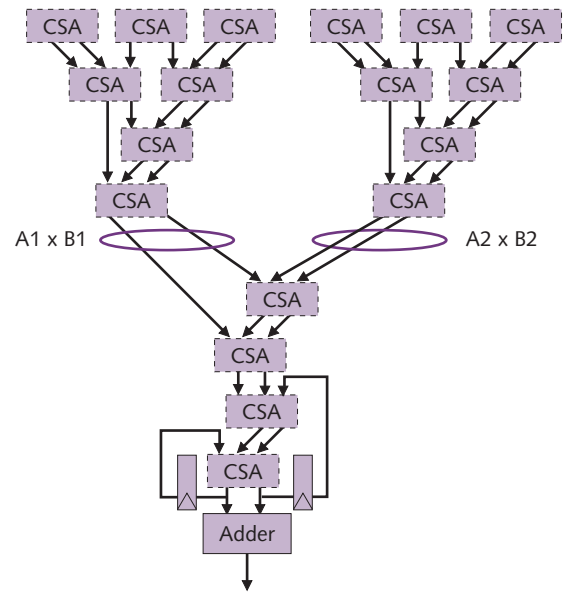


Figure 6. The multiplier datapath supports dual-multiply operations and a repeat rate of one MAC per cycle.

multiply datapath to implement most of the DSP multiply logic, and we did so without affecting existing multiply instructions. Furthermore, the implementation includes DSP MACs, which can be executed back-to-back for an optimal repeat rate.

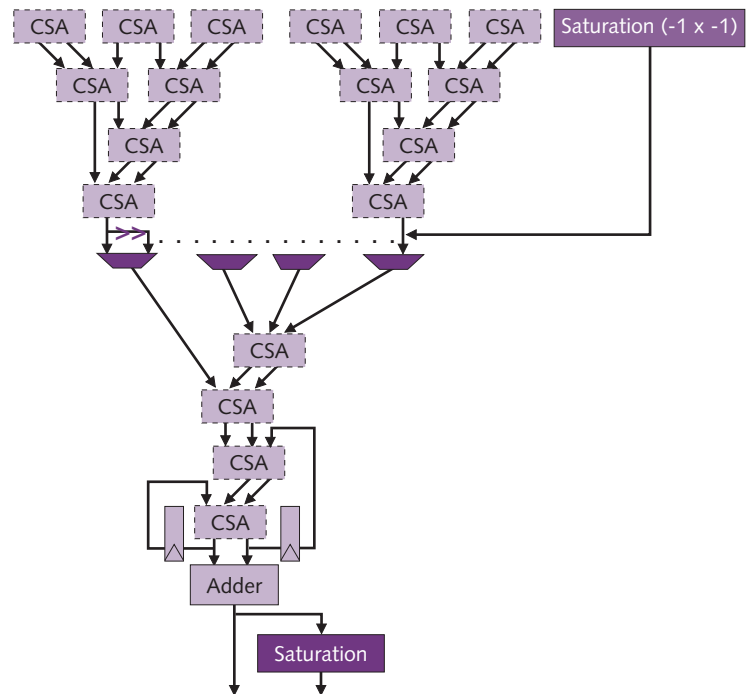


Figure 7. Minimal additional logic was added to support DSP multiply instructions.



## References

1. W. Strauss, from personal correspondence with the author.
2. W.P. Hays, "DSPs: Back to the Future," *ACM Queue*, pp. 42–51, March 2004.
3. J.L. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufman, San Francisco, California, 1996.
4. A.D. Booth, "A Signed Binary Multiplication Technique," *Quarterly Journal of Mechanics & Applied Mathematics*, Oxford University Press, Vol. 4, Part 2, pp. 236–240, 1951.
5. V.C. Hamacher, Z.G. Vranesic, and S.G. Zaky, *Computer Organization*, McGraw-Hill Book Company, San Francisco, California, 1984.
6. C.G. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronics Computers*, Vol. EC-13, February 1964, pp. 14–17.

## Measuring the Results

Table 1 compares the key characteristics of the 24KEc core, when it is fabricated in TSMC's 0.13-micron G process, with the 24Kc core. Both cores are capable of 400MHz, worst case, using a high-speed library. The 24Kc core can achieve a slightly higher speed, but we target our synthesis runs at 400MHz. The additional DSP logic is about 9% of the core logic. This corresponds to an increase of only 2.7% of the total core-die area with 32KB caches. Therefore, the DSP logic has negligible effects on processor speed and total area.

At 400MHz, the 24KEc core can reach 800 million multiply-accumulate operations per second (MMACS) with 16-bit data. Figure 8 shows a wide range of DSP performance measurements using the new DSP extensions. These measurements are mostly for inner loops of functions, so the overall application speedup will depend on the percentage of

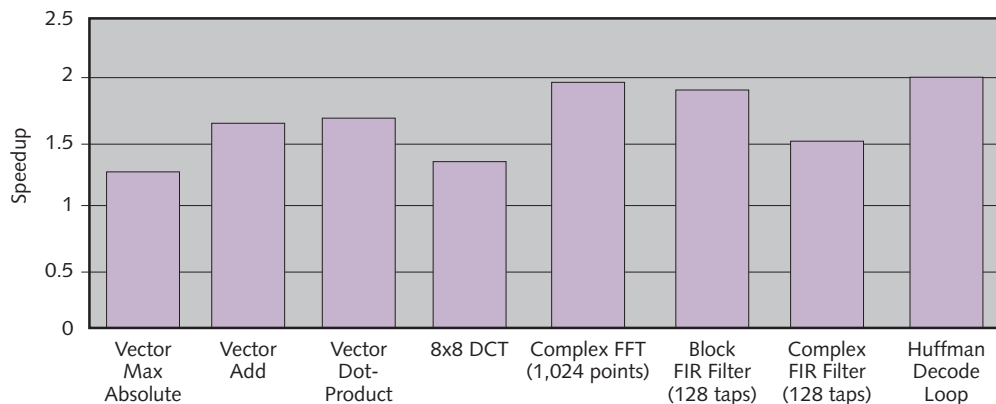


Figure 8. The 24KEc core executes DSP functions at up to twice the speed of the 24Kc core.

Feature	MIPS 24KEc	MIPS 24Kc	Difference
Core Freq	400MHz	400+MHz	1%
Logic Gates	377K	347K	9%
Die Area	7.5mm <sup>2</sup>	7.3mm <sup>2</sup>	2.7%

Table 1. Adding DSP extensions to the 24KEc had a negligible effect on the processor's clock frequency and die area. These die-area estimates are based on implementing the processors in TSMC's 0.13-micron G process after configuring the cores with 32KB instruction and data caches. The base size of the 24KEc core without caches is 3.0mm<sup>2</sup> in this process.

time these loops account for in the application. Note that the speedup numbers compare hand-optimized MIPS32 assembly code on a 24Kc core (without DSP instructions) with hand-optimized assembly code on the 24KEc core (using DSP instructions). We ran these experiments on 24Kc and 24KEc core simulators.

## Summary

The 24KE core family delivers accelerated DSP performance at a negligible cost. Simulation results show that the 24KEc core significantly improves DSP performance over the 24Kc core. The 24KEc core demonstrates that such meaningful DSP enhancements can be made to a RISC processor core with negligible effects on clock speed and die area. The success in overcoming the design challenges of implementing the 24KEc core further confirms the viability of moving digital-signal processing onto the main processor core.

Merging digital-signal processing onto the main processor also offers other benefits. The combined processing on a single processor creates a more efficient system architecture and results in smaller die area and lower power. The common tool chain for general-purpose and DSP code lowers software-development costs and shortens software-development time. All these factors make the 24KE core family ideal for embedded applications requiring a high-performance, low-cost synthesizable microprocessor with accelerated DSP performance.

The 24KE core family is already available for licensing to early adopters and will become generally available later this summer. Processor cores from MIPS Technologies Inc. have traditionally been used as the host processor in embedded devices. Now MIPS cores, specifically the 24KE core family, can run digital-signal processing applications efficiently and hence are well suited to meet the convergence of RISC processors and DSPs to yield system-on-chip integration benefits.

# MIPS 24KE: BETTER LATE THAN NEVER

By Tom R. Halfhill,  
Senior Analyst, Microprocessor Report

Adding DSP extensions to the MIPS32 24K family is a logical move. As MIPS Technologies points out in its Spring Processor Forum white paper, general-purpose processors are gradually subsuming DSP functions, and consumer electronics—a key market for MIPS—are at the crossroad of this convergence. What's surprising is that MIPS has taken so long to embrace DSP. Competitors such as ARC International, ARM, and Tensilica have offered DSP extensions or DSP coprocessors for years. The MIPS DSP Application-Specific Extensions (ASE) are welcome but overdue.

Another surprise is that MIPS isn't striving to outdo the competition in terms of sheer throughput. Independent benchmarks and our own analysis lead *Microprocessor Report* to believe that the DSP options from ARC, ARM, and Tensilica are more powerful than those from MIPS. However, the MIPS white paper offers a good reason for the 24KE's limitations: MIPS sought to improve signal-processing performance without compromising the best-in-class clock speed of the 24K processor core and without significantly

inflating the core's die area and power consumption. Remarkably, the MIPS 24KE achieves all those objectives.

As a result, the 24KE can execute some signal-processing tasks twice as fast as an unenhanced 24K core, yet the 24KE doesn't sacrifice performance when running control code and general-purpose application code. In some instances, a single MIPS 24KE processor core will be able to perform the dual roles of a general-purpose processor and a DSP core in the same SoC—a desirable goal.

The new 24KE family is a sensible union of the 24K family announced at Microprocessor Forum 2003 (see *MPR 10/20/03-03*, "MIPS Reveals 24K Core Family") and the DSP ASE announced at Fall Processor Forum 2004 (see *MPR 11/1/04-02*, "MIPS Takes Aim at Low-Cost DSP"). Essentially, MIPS has strengthened its fastest 32-bit synthesizable processor cores with extensions that can shoulder part of the signal-processing burden or even eliminate altogether the need for a separate DSP core. This allows MIPS to nibble on the \$18 billion DSP market while simplifying

Feature	MIPS 24KEc	MIPS 24KEf	MIPS 24KEc Pro	MIPS 24KEf Pro	MIPS 24Kc	MIPS 24Kf	MIPS 24Kc Pro	MIPS 24Kf Pro
Architecture	MIPS32	MIPS32	MIPS32	MIPS32	MIPS32	MIPS32	MIPS32	MIPS32
CPU Core	24K	24K	24K	24K	24K	24K	24K	24K
Core Freq (0.13µm)	400–625MHz	400–625MHz	400–625MHz	400–625MHz	400–625MHz	400–625MHz	400–625MHz	400–625MHz
Core Size (Base)	3.0mm <sup>2</sup>	4.8mm <sup>2</sup>	3.0mm <sup>2</sup>	4.8mm <sup>2</sup>	2.8mm <sup>2</sup>	4.6mm <sup>2</sup>	2.8mm <sup>2</sup>	4.6mm <sup>2</sup>
Synthesizable	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Pipeline Depth	8 stages	8 stages	8 stages	8 stages	8 stages	8 stages	8 stages	8 stages
MIPS16e	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
I-Cache	0–64K	0–64K	0–64K	0–64K	0–64K	0–64K	0–64K	0–64K
D-Cache	0–64K	0–64K	0–64K	0–64K	0–64K	0–64K	0–64K	0–64K
MMU	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integer Mul-Div	32x32-bit	32x32-bit	32x32-bit	32x32-bit	32x32-bit	32x32-bit	32x32-bit	32x32-bit
Bus Interface	OCP 2.x 64-bit	OCP 2.x 64-bit	OCP 2.x 64-bit	OCP 2.x 64-bit	OCP 2.x 64-bit	OCP 2.x 64-bit	OCP 2.x 64-bit	OCP 2.x 64-bit
Coprocessor I/F	64-bit COP2	64-bit COP2	64-bit COP2	64-bit COP2	64-bit COP2	64-bit COP2	64-bit COP2	64-bit COP2
FPU	—	32/64-bit	—	32/64-bit	—	32/64-bit	—	32/64-bit
DSP ASE	Yes	Yes	Yes	Yes	—	—	—	—
CorExtend	—	—	Yes	Yes	—	—	Yes	Yes
Performance (Dhrystone 2.1)	576–900 DMIPS	576–900 DMIPS	576–900 DMIPS	576–900 DMIPS	576–900 DMIPS	576–900 DMIPS	576–900 DMIPS	576–900 DMIPS
Power (Base Core, 1.2V)	0.58mW per MHz	0.61mW per MHz	0.58mW per MHz	0.61mW per MHz	0.58mW per MHz	0.61mW per MHz	0.58mW per MHz	0.61mW per MHz
Gen. Availability	3Q05	3Q05	3Q05	3Q05	2004	2004	2004	2004

MIPS has doubled the number of MIPS32 24K-series processor cores by introducing the 24KE family. All are based on the same microarchitecture. The main differences, highlighted in purple, are the DSP extensions, CorExtend interface, and FPU. Those differences affect the die area and power consumption of each core.

hardware and software development for customers. If a single 24KE processor core can do the job of two cores, programmers can write all their code using a single tool chain.

Target applications include speech processing (voice over Internet, speech recognition, echo/noise cancellation, channel equalization); digital audio (MP3, Dolby Digital, Windows Media, Advanced Audio Coding); video (MPEG, Windows Media); imaging (JPEG, printing); graphics; and Internet access (cable modems, DSL modems, soft modems). MIPS already has a strong presence in many of those markets.

In all, the 24KE family debuts four new processor cores, starting with the 24KEc base core. Like all 24K-series cores, the 24KEc has configurable caches, optional scratchpad memory, a configurable memory-management unit (MMU), and the option of replacing the MMU's translation look-aside buffer (TLB) with fixed memory mapping. The 24KEc Pro adds CorExtend, which allows customers to create application-specific instructions. (See *MPR 3/3/03-01*, "MIPS Embraces Configurable Technology.") The 24KEf adds a single- and double-precision FPU to the base core, and the 24KEf Pro has both the FPU and CorExtend. See the table for a feature comparison of all 24K and 24KE processor cores.

### MIPS Preserves General-Purpose Performance

When creating its DSP extensions, MIPS tried to stay true to the RISC philosophy by providing enhancements for common signal-processing tasks while avoiding operations that would reduce the core's clock frequency. Critical inner loops will benefit most from these extensions. The DSP ASE adds more than 60 new instructions (not counting minor variations for different data types and addressing modes); three new 64-bit high/low accumulators (in addition to one existing high/low accumulator in the MIPS32 architecture); and a new DSP control register (for condition codes and other state information).

These extensions are nothing to sneeze at. They support saturating and fractional math, 32×32-bit multiply-accumulate (MAC) instructions, dot-product operations, and variable bit-inserts and extractions. New SIMD instructions handle 8-, 16-, and 32-bit data types. MIPS has applied for a patent on one unique feature: virtual circular buffers. This feature conserves die area and power by supporting circular buffers without the postmodified pointers and states typically found in DSPs.

As the white paper explains, MIPS was intent on adding the DSP extensions without significantly altering the 24K pipeline, impairing the processor's high clock frequency, or drastically inflating the core's area and power. It's impressive that MIPS has added DSP extensions without sacrificing the processor's ability to reach 400–625MHz (worst-case, depending on the synthesis libraries) in a 0.13-micron CMOS process.

In this regard, competitors fare less well. Adding DSP extensions to a Tensilica Xtensa LX processor can reduce the clock frequency from its worst-case maximum of 350MHz to

about 270MHz—a 23% reduction. ARC's configurable processors, too, lose significant clock speed after adding DSP extensions. Of course, the trade-off is that DSP extensions usually compensate for the loss of clock frequency by improving signal-processing performance to an even greater degree. That's why ARC, ARM, and Tensilica have opted for more powerful DSP extensions and coprocessors than MIPS has.

### MIPS Faces Tough Competition

For example, ARC's XY Advanced DSP Extensions for the ARC 600 and ARC 700 processors include one or two banks of X/Y data memory—a common feature in dedicated DSPs. Using this on-chip SRAM, an ARC processor can fetch two input operands and write back the results in a single clock cycle. An internal DMA controller moves data back and forth without impeding the processor's main pipeline. Multiple addressing modes (variable offset, modulo, and bit reversing) with pre- and postindexing provide additional flexibility. ARC's DSP instructions can also execute zero-overhead loops, another valuable feature of dedicated DSPs. However, adding all these extensions can easily double the size of an ARC processor core and reduce the clock frequency by about 25%.

ARM offers multiple DSP options. In 1999, ARM introduced the ARMv5TE DSP extensions, which can run signal-processing algorithms about two to three times faster than regular ARMv5 instructions. In 2001, ARM introduced the ARMv6 SIMD media extensions, which add more than 60 new instructions, including single-cycle 32×16-bit and 16×16-bit MACs—but no 32×32-bit MACs, as found in the MIPS DSP ASE. (See *MPR 1/2/01-03*, "ARM Embraces SIMD Support.") Neither the ARMv5TE nor ARMv6 SIMD extensions have the X/Y memory, zero-overhead looping, or special addressing modes found in ARC's extensions.

Seeking better signal-processing performance, ARM announced two more options last year: OptimoDE and NEON. OptimoDE is a configurable VLIW data engine that ARM acquired from Adelante Technologies in 2003. (See *MPR 6/7/04-01*, "ARM's Configurable OptimoDE.") NEON extends the ARMv7 architecture with more than 100 new 32-bit ARM and 16-bit Thumb-2 instructions, including integer and single-precision floating-point operations. However, both OptimoDE and NEON are large coprocessors, not mere core extensions, so it's unfair to compare them directly with the MIPS 24KE. The ARMv6 extensions are more comparable.

ARM certainly likes to compare ARMv6 with the 24KE. The same day MIPS announced the 24KE at SPB, ARM released independent benchmark scores for an ARM1136J-S processor with ARMv6 extensions. According to benchmark results certified by Berkeley Design Technology Inc. (BDTI), the ARM1136J-S achieved a BDTIsimMark2000 score of 1,230 when simulated at 350MHz. ARM claims its BDTIsimMark2000 score is about 25% better than preliminary BDTI benchmark results for a MIPS32 24K processor with the DSP ASE. However, BDTI points out that those preliminary MIPS results are not a certified BDTIsimMark2000 composite



score. Instead, they are based on only four of the twelve benchmark tests in the suite. MIPS and BDTI are still working on the remaining benchmark tests.

Tensilica can boast of having the most powerful DSP option for a licensable RISC core. Last year, Tensilica introduced the Vectra LX engine for Xtensa LX. Vectra LX has 64-bit instruction words containing three issue slots for ALU, MAC, and load/store operations. Tensilica ran the BDTI benchmark suite on an Xtensa LX with Vectra LX and additional extensions optimized especially for the benchmarks. This configuration outperformed every other licensable DSP or CPU core ever tested by BDTI, scoring 6,150 in the BDTI<sub>sim</sub>Mark2000 suite. (See *MPR 5/31/04-01*, "Tensilica Tackles Bottlenecks.")

However, Tensilica's trophy comes at a price. The Xtensa LX base configuration tested by BDTI would occupy about 4.42mm<sup>2</sup> when fabricated in a 0.13-micron process, and its estimated worst-case clock frequency is 369MHz. In contrast, the base configuration of the MIPS 24KEc is only about 3.0mm<sup>2</sup> and could reach 625MHz in the same process.

### MIPS Loyalists Have Additional Options

Of course, benchmark results are always subject to interpretation. Even BDTI notes that maximum throughput isn't necessarily the most important factor when selecting a DSP or signal-processing extensions. For many embedded applications, a particular level of throughput is sufficient, and developers seek the lowest-cost and lowest-power solution that handles the workload.

Although *MPR* believes the MIPS DSP extensions are less powerful than the DSP offerings from other licensable-IP companies, we also think the DSP ASE has several advantages. It maintains the high clock speed of the MIPS 24K core and doesn't hamper the processor's ability to execute control code and general-purpose application code. It adds more new instructions than ARC's Advanced XY DSP Extensions.

### Price & Availability

The MIPS32 24KE core family is available now to early adopters and will become generally available in 3Q05. MIPS Technologies has not disclosed licensing fees. For more information, see [www.mips.com/content/Products/Cores/32-BitCores](http://www.mips.com/content/Products/Cores/32-BitCores).

It requires less silicon and power than higher-performance competing solutions. And it supports 32-bit signal processing, unlike Tensilica's 16-bit Vectra LX.

Nevertheless, DSP-hungry customers uncommitted to a particular CPU architecture will be tempted by the DSP options from ARC, ARM, and Tensilica. Existing MIPS customers will surely welcome the 24KE family—assuming the performance improvements of the DSP ASE are sufficient for their needs. If the DSP extensions prove insufficient, MIPS customers have two alternatives to switching CPU architectures.

One alternative is to license a 24KE Pro-series core and use CorExtend to supplement the DSP ASE with custom DSP extensions. The second alternative is to license a DSP engine from a third-party IP vendor (such as LSI Logic) and attach it to the MIPS core. Both options can significantly boost signal-processing performance, although both require more design work and verification. The second alternative also incurs the cost of additional licensing and the nuisance of dividing software development across multiple tool chains.

Obviously, MIPS hopes customers will find enough horsepower in the 24KE cores to satisfy their signal-processing requirements without resorting to any alternatives. This union of the 24K family and DSP ASE should help attract new licensees and prevent existing ones from defecting to other CPU architectures. ♦

To subscribe to Microprocessor Report, phone 480.483.4441 or visit [www.MDRonline.com](http://www.MDRonline.com)