

MICROPROCESSOR AND DSP TECHNOLOGIES UNITE FOR EMBEDDED APPLICATIONS

MARKUS LEVY, TECHNICAL EDITOR

What is the difference between a μ P and a DSP? According to many industry cohorts, there is no difference. Diverse, high-volume applications, including cell phones, disk drives, antilocking brakes, modems, and fax machines, require both μ P and DSP capabilities. This requirement has led many μ P vendors to build in DSP functionality. In some cases, such as in Siemens Tricore architecture, the functional merging is so complete that it is difficult to determine whether you should call the device a DSP or a μ P. At the other extreme, some vendors claim that their μ Ps have high-performance DSP capability, when in fact they've added only a simple 16×16 -bit multiply instruction. Your key to a successful DSPless system design is understanding how much DSP capability you need and determining whether a μ P can meet those needs.

Many embedded applications require a mixture of μ P and DSP functionality; semiconductor vendors are creating hybrid devices to handle both types of processing. Before you begin your next design with discrete μ P and DSP devices, check out the benefits and limitations that these hybrid devices offer.

Most μ Ps come nowhere near implementing all the standard DSP functions (see box Features that make DSPs DSPs). But how much DSP functionality do you need? The answer depends on the trade-offs you're willing to make between having enough DSP horsepower to get the job done on the one hand and practical factors, such as cost, power consumption, ease of development, and board space on the other. The fundamental choices for implementing DSP in your application range from discrete DSP devices to a DSP core and μ P on the same die to a hybrid μ P-DSP to a μ P with an integrated multiply-accumulate (MAC) unit.

To reach the top of the DSP-performance ladder, you must use discrete DSP devices. But you trade off higher cost and power consumption, increase board space, and deal with two development environments (the μ P and the DSP). Putting DSP and μ P cores on one die yields even better DSP performance and also helps reduce power consumption and board space (see box Dueling with dual-core designs).

Hybrid processors represent the next step down the DSP-performance ladder. The prominent architectures in this category include ARM's Piccolo, Hitachi's SH-DSP, Hyperstone's EI-32, and Siemens Tricore. Although these implementations differ, a hybrid processor is basically a μ P tightly coupled to a DSP (or vice versa, depending on your viewpoint). In general, the hybrid approach permits the μ P and DSP portions to share on-chip resources, such as memory and other peripherals, thereby helping to eliminate redundancies and to reduce power consumption. Furthermore, a hybrid processor typically allows you to use one set of software-development tools, a single RTOS, and a one system task scheduler.

Ps AND DSPs CONVERGE

Although these hybrid processors can perform a single-cycle MAC operation, the key to achieving sustained MAC operations relates to how fast they can access operands. In turn, this speed relates to how the DSP portion of the core interfaces with the μ P portion, as well as the memory configuration. For example, in Tricore, the DSP and μ P portions are inseparable (Figure 1). This superscalar RISC core has two four-stage pipelines; the first bit of every instruction identifies which pipeline that instruction follows. One pipeline does loops, loads, and address-generation arithmetic; the other pipeline does all the math and branches. The absence of separate X and Y memory spaces may require you to perform some loop unrolling to achieve the parallel performance of DSPs. But Siemens engineers added other features to make Tricore look more like a DSP. These features include zero-overhead loop capability, packed multiply to handle single-instruction, multiple-data operations, and a loop cache.

Hyperstone's E1-32, like Tricore, is a load/store architecture with an integrated DSP unit that works in parallel with the ALU: It can perform DSP calculations while the ALU performs loop counts, address calculations, or load/store operations. Hyperstone based the E1-32 on a two-stage pipeline, and the device can issue only one instruction per cycle. The DSP instructions require two or more cycles to complete, and the ALU executes its instructions during the latency cycles of DSP instructions. You or the compiler must arrange your code to take advantage of these latency cycles. And, because the E1-32 does not support separate X and Y memory blocks, you would have to perform all loads and stores during the latency cycles to achieve reasonable DSP performance.

Zero-overhead looping on E1-32 requires you to execute two MAC operations per loop and use the latency cycles to perform the address calculations, data loads, and compare instructions. The 100-MHz operation of the E1-32 helps reduce the penalties incurred by having to execute the extra instructions.

The SH-DSP takes a slightly different approach to integrating its DSP function-

@ a glance

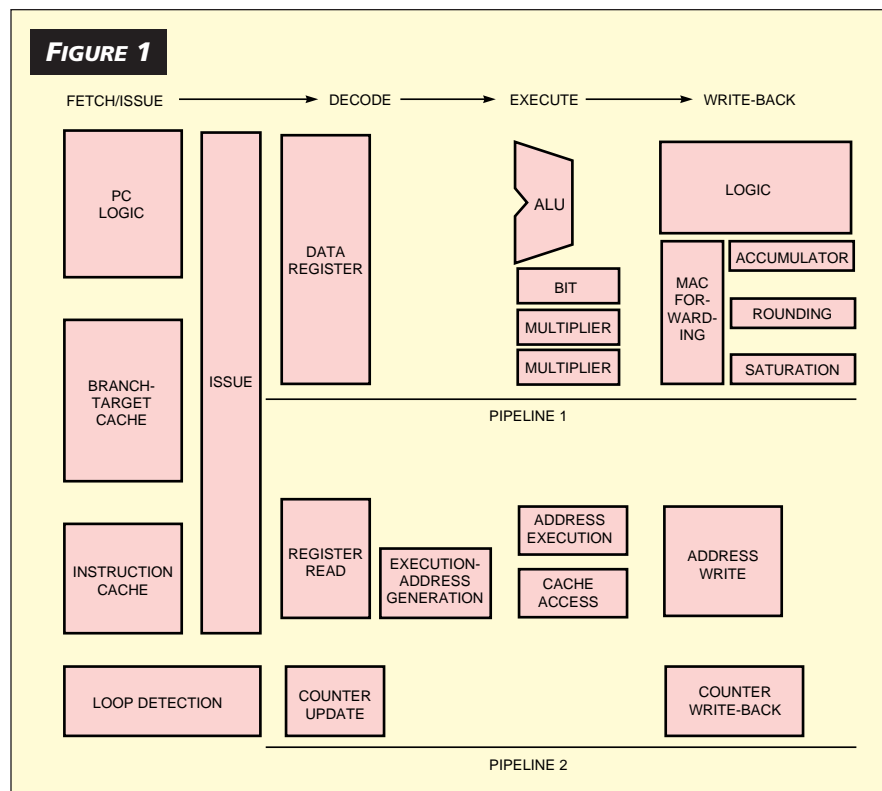
Add your μ P and DSP instructions to determine whether a hybrid device can deliver the performance your application needs.

- Hybrid devices simplify system development because your DSP and μ P share resources and because you need to debug only one instruction stream.
- Most μ Ps with integrated MAC units require higher clock rates to get good DSP performance.

ality. In this case, Hitachi designers grafted the DSP unit onto the RISC's pipeline; the DSP unit shares the five-stage pipeline with the integer unit. Although the SH-DSP is a single core, the instruction stream comprises both integer and DSP instructions. When the CPU fetches

and decodes instructions, it routes instructions to the appropriate unit. Furthermore, each unit has its own set of registers; the DSP unit's registers are visible only to the DSP unit and to DSP extended load/store instructions.

The SH-DSP also differs from Tricore and Hyperstone in that its DSP unit has its own data-memory space. Similar to more traditional DSPs, Hitachi's architecture has separately addressable X and Y memories. The main integer ALU calculates X addresses, and a separate, 16-bit pointer-arithmetic unit calculates Y memory addresses. And, although the integer and DSP units can't execute instructions in parallel because they share the chip's internal address and data buses, the bus structure does allow the DSP to access two data operands and fetch an instruction during one cycle. During that same single cycle, the SH-DSP can also execute one ALU operation and a 16×16 -bit multiply.



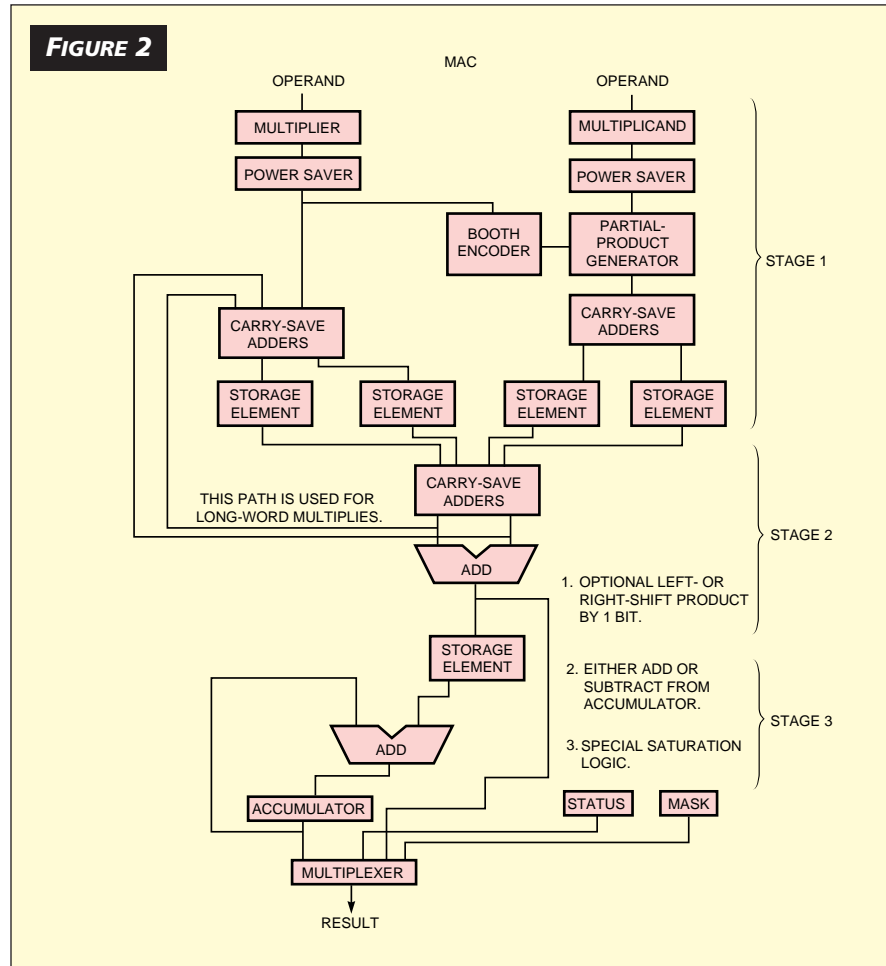
Siemens Tricore architecture has one pipeline to perform loops, loads, and address-generation arithmetic and another pipeline to perform all the math and branches. A third pipeline supports DSP looping.

Ps AND DSPs CONVERGE

The SH-DSP contains an instruction bus (I-bus), which the CPU also uses to load/store the RISC or DSP registers. If the CPU is loading or storing registers over the I-bus, the DSP cannot simultaneously perform other parallel operations. However, the DSP does allow parallel operations if you are using the X and Y buses for the moves to or from the on-chip X and Y memories, thus yielding sustained single-cycle MAC operations.

On another note, ARM's Piccolo (SP7) is more of a DSP coprocessor module than a hybrid processor. ARM licensees can attach SP7 to an ARM-7TDMI core to add DSP functionality to an ARM chip design. SP7 instructions are incompatible with the standard ARM instruction set; the coprocessor executes them in a different pipeline, which means that you must debug two separate instruction streams.

SP7 features a 16×16-bit single-cycle multiplier, a 32-bit barrel shifter, four 48-bit extended-precision accumulators, a saturation unit, and other DSP functions, but the coprocessor lacks separate X and Y memories for operand access. To help SP7 achieve sustained single-cycle MACs, its interface includes a tagged input-queue structure and an output FIFO buffer. The ARM7TDMI core performs all the address generation for the DSP functions. The input queue, or reorder buffer, enables the ARM7TDMI to preload SP7 with data before SP7 requires the data, essentially demulti-



The multiply-accumulate (MAC) unit for Motorola's ColdFire architecture comprises a three-stage arithmetic pipeline that can perform a MAC operation every one to two cycles, depending on the location of the data operands.

FEATURES THAT MAKE DSPs DSPs

Performing efficient DSP on a μ P is tricky business. Although the ability to support single-cycle multiply-accumulates (MACs) is the most important function a DSP performs, many other functions are critical for real-time DSP applications. Executing a real-time DSP application requires an architecture that supports high-speed data flow to and from the computation units and memory through a multiport register file. This execution often involves the use of DMA units and dual data-address generators (DAGs) that operate in parallel with other chip resources. DAGs, which perform address calculations, allow the DSP to bring in two pieces of data per clock critical for real-time DSP algorithms.

It is important for DSPs to have an efficient looping mechanism, because most DSP code is highly repetitive. The architecture allows for zero-overhead looping, in which you use no additional instructions to check the completion of the

loop iterations. Generally, DSPs take looping a step further by including the ability to handle nested loops.

DSPs must typically handle extended precision and dynamic range to avoid overflow and minimize round-off errors. To accommodate this capability, DSPs typically include dedicated accumulators with registers wider than the nominal word size to preserve precision. DSPs must also support circular buffers to handle algorithmic functions, such as tapped delay lines and coefficient buffers. DSP hardware updates circular-buffer pointers during every cycle in parallel with other chip resources. During each clock cycle, the circular-buffer hardware performs an end-of-buffer comparison test and must reset the pointer without overhead when the pointer reaches the end of the buffer. FFTs and other DSP algorithms require another DSP feature, bit-reversed addressing.

Ps AND DSPs CONVERGE

plexing multiple input-data streams for DSP algorithms. The reorder buffer allows ARM code to fetch DSP data or coefficients from memory and allows the DSP code to consume the items in the required order. The ARM7TDMI core can transfer as many as 16 16-bit words of data in nine 32-bit bus cycle. SP7 uses a remapping scheme to automatically and transparently refill its registers from the reorder buffer as it uses and replaces old data. Because Piccolo reloads only from the reorder buffer, a data-intensive algorithm may starve the register file. Furthermore, the programmer must ensure that the ARM core responds to the needs of the DSP. In other words, Piccolo cannot interrupt or notify the ARM core when

Piccolo needs data or has full output buffers.

The ARM core performs bit-reversed addressing in software, but it can accomplish this task while SP7 is executing the first stage of the FFT. ARM claims that this approach eliminates any overhead. Register remapping allows ARM to logically remap physical registers within a loop to other register values. This approach effectively creates small circular buffers in the register banks.

TMS320C6x: hybrid μ P or DSP?

Despite the fact that TI is known as a DSP company, its TMS320C6x blurs the line between μ Ps and DSPs. The C6x, which TI based on a very-long-instruc-

tion-word architecture, contains two 16×16 -bit multipliers and six 32-bit arithmetic units with a 40-bit ALU and a 40-bit barrel shifter. You can use these functional units for integer and DSP operations. To maintain the flexibility of its functional units, the C6x lacks a dedicated MAC unit, which is fundamental to most DSPs. It instead performs MAC operations by using separate multiply and add instructions; the pipeline allows the C6x to effectively execute MAC operations in one cycle. Also unlike most DSPs, the C6x does not support separate X and Y memory spaces. Instead, it provides a single data memory with two 32-bit paths for loading data from memory to the register banks. The C6x lacks dedicated

DUELING WITH DUAL-CORE DESIGNS

Designing a multiprocessing system with discrete μ P and DSP chips has its challenges. From a hardware perspective, you must deal with the interprocessor communication by implementing features such as semaphores, mailboxes, and other types of handshaking mechanisms. You may also need some dual-port RAM to allow the chips to share some of each other's memory space. Some systems present additional hardware challenges when the μ P and DSP must communicate over a system bus. Companies such as Pentek (Upper Saddle River, NJ; www.pentek.com) and Spectrum Signal Processing (Burnaby, BC, Canada; www.spectrumsignal.com) dedicate a large part of their resources to improving the data-transfer rate between a host μ P and the DSPs on their boards.

Multiprocessing μ P/DSP systems also have challenges from a software perspective. One of the biggest challenges is developing and debugging code on two different architectures. Generally, OEMs solve this problem by employing separate design teams for the μ P and DSP portions. You can further simplify this problem by purchasing canned software from companies such as HotHaus (Richmond, BC, Canada; www.hothaus.com) and DSP Software Engineering (Bedford, MA; www.dspse.com). However, you still face the task of meshing the two operating environments.

Dual-core designs present unique challenges

Silicon-process technologies are making it more practical to put a μ P and a DSP core on one die. So, in addition to all the challenges that discrete-chip multiprocessor designs incur, more designers are facing the unique challenges that these dual-core devices bring with them.

In designs with multiple chips, you can use traditional lab equipment, such as logic analyzers and oscilloscopes, to monitor the interaction between the devices. In a dual-core design, the need for a small die and package size forces chip

designers to bury the interface between the cores in silicon. The debugger developers must provide visibility into the interaction between the cores. Furthermore, the two cores must access external memory through a common address and data bus; while one core accesses external memory, the other core stalls. Vendors typically solve this problem by giving the DSP higher priority, but this approach restricts design flexibility.

Ideally, because a dual-core design is a single chip, you should have one debugger. A user should at least be able to work with both debuggers as if they were talking to discrete devices, uncovering the hidden interface between the cores. A few vendors of dual-core devices either build in special features or offer special tools to help you with your development work. The most common debugging feature is the use of one JTAG port to access both cores. The port typically includes a switch pin to allow you to direct control into either core. For example, Motorola offers this capability with its Redcap product, combining an M-Core RISC core with a DSP56600 DSP core. Redcap is a custom dual-core baseband processor that Motorola developed for one of its sister divisions. The integrated debugging support allows you to separately single-step each core. Motorola also adds features to start and stop both cores simultaneously. The On-Chip Emulation circuitry (OnCE) design allows the debugger to prime both cores before releasing the cores to begin running.

VLSI Technology offers a dual-core chip with an ARM μ P and DSP Group's Oak DSP core; this chip also includes a single JTAG port. The company provides its \$7500 Vector Multicore Development System (V/MDS), which comprises a development platform with a development chip and software-development tools (Figure A). Although the software tools do not meld the two instruction streams, VLSI uses a dynamic-link-library approach to tie the two software-development environments together. The development platform maintains

address-generation units and must calculate addresses, including circular buffers, using one or more of its functional units.

The third step down the DSP performance ladder is a μ P that integrates a MAC unit. Similar to the hybrid approach, this DSP implementation helps to eliminate redundancies; in most cases, the MAC unit becomes part of the μ P's pipeline. Depending on the processor supporting the MAC unit, this approach can yield enough DSP performance to handle low-end-disk-drive servo-control loops, soft mo-dems, software JPEG compression and decompression, and other types of low-data-rate DSP algorithms. For example, using software from

AltoCom (Mountain View, CA; www.alto-com.com), Integrated Device Technology's (IDT's) RV4640 μ P can run a 56-kbps soft mo-dem, but the μ P may limit processing-power head room, according to IDT.

Although an integrated MAC unit lacks the functionality of a hybrid or discrete DSP, vendors play tricks to increase DSP performance. One obvious trick is implementing a high-frequency clock. For example, Digital runs its StrongARM μ P at 200 MHz to compensate for the μ P's inability to achieve sustained single-cycle MAC operations.

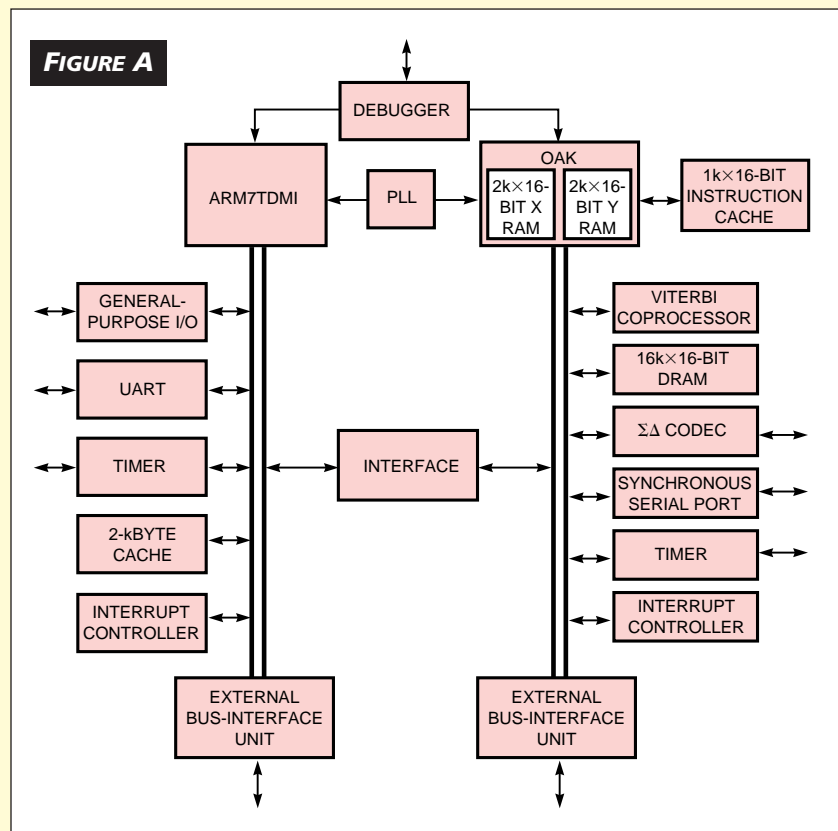
SGS-Thomson's ST10x262 is one of the few 16-bit μ Cs to include a MAC unit. The 262's MAC unit comprises a 16×16 -

bit parallel multiplier with a 40-bit accumulator and a 40-bit ALU to help support saturation. The MAC has two 16-bit data-paths to deliver two operands each cycle; these operands come from the 262's 256-word, dual-port RAM. This approach allows the 262 to deliver single-instruction-cycle MACs with two cycles per instruction until it has to reload the RAM. To improve DSP performance, the 262's MAC unit contains an interruptible repeat unit that supports as many as 8196 iterations through a loop. The 262 also includes special MAC instructions comprising multiply and MAC, 32-bit arithmetic, shifts, compares, and transfer instructions. The MAC unit first sign-extends each signed 32-bit product, then

the clocking relationship between the cores. In debugging mode, a single step from the host clocks the board once, allowing the cores to maintain their relationship relative to that external clock.

For \$2200, GEC offers a similar development board, the GEM301 Assessment Platform 1 (GAP-1). This board houses GEC's GEM301 Global System for Mobile communications (GSM) baseband processor that contains an ARM7TDMI μ P and the DSP Group's Oak cores. The GEM301 supports the digital-baseband functionality that a GSM handset requires. Although both cores have their own debugging systems, GEC designed a memory interface that allows the ARM's debugging system to supply values for signals that would normally originate from the Oak DSP and vice versa. But hardware support for interprocessor-debugging coordination is limited to start/stop synchronization; the ARM subsystem is responsible for restarting an application after a debugging breakpoint.

Another interesting twist, as a customer-specific Siemens Triocore device demonstrates, is to put two of the same core on one die. This approach eases debugging because it lets you use one debugger. But this approach also offers performance benefits. Using dual Triocore cores provides Siemens customers with a deterministic interrupt-response



The two cores on VLSI's WS3670 multiprocessor development chip run asynchronously with independent bus resources. A single JTAG interface links the two cores for debugging.

time. If one device is busy, the shared interrupt controller issues the next-priority-level interrupt to the other device.

Ps AND DSPs CONVERGE

you can optionally either negate or round the product before the MAC adds it to the 40-bit accumulator register.

At the other end of the μ P spectrum, IDT's 64-bit RV4640 supports a two-cycle MAC operation; high-frequency operation allows this chip to achieve good DSP performance. However, you can achieve a sustained two-cycle MAC only by loading operands into the register file during the intervening clock cycle while the CPU executes the MAC instruction. The hardware automatically interlocks.

As with many μ Ps with integrated MAC units, you must also perform loop unrolling to obtain two-cycle MAC operations. Although the RV4640 supports a MAC instruction, the lack of zero-overhead-looping capability makes it impossible for you to write a single-instruction inner loop; instead, you must include a branch with a decrement at the end of the loop. A predicted branch consumes one instruction cycle, but you can make up for this extra cycle by increasing the μ P's operating frequency.

Another issue common to μ Ps with integrated MAC units is the lack of circular buffers; you must perform address manipulation in software. For example, using the RV4640's support for register and offset addressing, you can partially unroll a loop and use explicit offsets to an address pointer within the unrolled loop. The bottom of the loop can implement the pointer update and circular addressing.

Motorola has a MAC unit for its ColdFire architecture that serves as another example of μ Ps with integrated MAC units (Figure 2). The ColdFire's big MAC is not a whopper, consuming only 8500 gates, or approximately 10% of the core. The MAC comprises a three-stage arithmetic pipeline that can achieve an effective 1.5-cycle MAC. A MAC-with-load instruction simultaneously dispatches a 16 \times 16-bit MAC operation and a 32-bit memory-to-register load. Although the MAC operation executes in one cycle, the load takes a minimum of two. Therefore the load holds up the MAC execution. But if your DSP algorithm uses sta-

tic coefficients that you can preload into registers, then the MAC can fetch two 16-bit operands (one 32-bit register) at a time.

The ColdFire's MAC unit supports no zero-overhead looping. This lack means that its inner loop is a minimum of three instructions: the MAC, decrement the loop counter, and the branch. The ColdFire's 90- to 100-MHz operation helps to compensate for some of these performance disadvantages. Circular-buffer support on ColdFire's MAC comprises a mask register with an autoincrement address mode. Without additional overhead, the CPU ANDs the mask register with the contents of the address register to constrain addresses into a circular loop; that is, the CPU prevents carry generation. This addressing scheme uses a 16-bit register, which yields circular buffers as large as 64 kbytes.

Another important ColdFire feature for DSP performance, as well as other time-critical applications, is that it lets you store code in internal SRAM or cache. The ColdFire Version 3 core allows you

MANUFACTURERS OF DSP-ORIENTED MICROPROCESSORS

For information on products such as those described in this article, circle the appropriate numbers on the Information Retrieval Service card or use EDN's Express Request service. When you contact any of the following manufacturers directly, please let them know you read about their products in EDN.

Advanced RISC Machines
(ARM) Los Gatos, CA
1-408-399-8853
www.arm.com
Circle No. 550

Digital Semiconductor
Maynard, MA
1-978-568-6868
www.digital.com/info/
semiconductor
Circle No. 551

DSP Group
Santa Clara, CA
1-408-986-4315
www.dspeg.com
Circle No. 552

GEC Plessey
Semiconductors
San Jose, CA
1-408-451-4700
www.gpsemi.com
Circle No. 553

Hitachi America Ltd
Brisbane, CA
1-415-589-8300
www.hitachi.com
Circle No. 554

Hyperstone Electronics
Cupertino, CA
1-408-257-1057
www.hyperstone.com
Circle No. 555

Integrated Device
Technology
Santa Clara, CA
1-800-345-7015
www.idt.com
Circle No. 556

Mitsubishi Electronics America
Sunnyvale, CA
1-408-730-5900
www.mitsubishichips.com
Circle No. 557

Motorola
Phoenix, AZ
1-800-441-2447
www.mot-sps.com
Circle No. 558

NEC Electronics Inc
Santa Clara, CA
1-800-366-9782
www.nec.com
Circle No. 559

SGS-Thomson
Microelectronics
Lincoln, MA
1-781-259-0300
www.st.com
Circle No. 560

Siemens
Cupertino, CA
1-408-777-4500
www.sci.siemens.com
Circle No. 561

Texas Instruments Inc
Dallas, TX
1-800-477-8924, ext 3555
www.ti.com
Circle No. 562

VLSI Technology Inc
San Jose, CA
1-408-434-3000
www.vlsi.com
Circle No. 563

VOTE ...
Please also use the Information Retrieval Service card to rate this article (circle one):
High Interest 581
Medium Interest 582
Low Interest 583

Super Circle Number

► For more information on the products available from all of the vendors listed in this box, circle one number on the reader service card. Circle No. 564

Ps AND DSPs CONVERGE

to lock the cache after storing the desired code. This approach minimizes the latency for time-critical algorithms.

NEC's V83x family also uses a pipelined MAC unit, but it runs at twice the internal frequency of the chip to help it maintain an effective single-cycle throughput; a MAC takes a minimum of three clock cycles to run through the pipeline. Again, operand availability is crucial to achieving this level of performance. Because the V83x's MAC operation is a three-operand instruction, the CPU should supply three 32-bit operands during the register-fetch stage of the pipeline. In one clock cycle, the CPU fetches two operands from the chip's dual-ported register file. The third operand must be available through internal forwarding from a previous operation for the MAC to hit its three-cycle latency. Usually, the compiler changes the instruction order to help minimize data dependencies. However, some code

does not allow the CPU to supply three operands during one clock cycle. This code imposes a one-clock penalty, bringing the MAC latency up to four clocks.

Reference

1. Turley, Jim, *Selecting a High-Performance Embedded Microprocessor*, Micro-Design Resources, Sebastopol, CA, 1997.



You can reach Technical Editor Markus Levy at 1-916-939-1642, fax 1-916-939-1650, markus.levy@worldnet.att.net.

VOTE

Please use the Information Retrieval Service card to rate this article (circle one):

**High
Interest**
581

**Medium
Interest**
582

**Low
Interest**
583