VOL. 10, NO. 15

Digital, MIPS Add Multimedia Extensions

Digital Focuses on Video, MIPS on 3D Graphics; Vendors Debate Differences

by Linley Gwennap

Support for multimedia data types has become nearly pervasive, as Alpha and MIPS have joined the throng of instruction-set architectures with multimedia extensions. At last month's Microprocessor Forum, Digital announced its motion-video instructions (MVI), which will first appear in the 21164PC and 21264 processors next year. Also at the conference, MIPS Technologies rolled out two sets of multimedia extensions. The first, MIPS V, supports parallel floating-point operations and will mainly benefit 3D graphics. A separate set of instructions called MDMX (MIPS digital media extensions) provides broader support for parallel integer operations.

The MIPS V extensions allow two single-precision operands to be stored in a double-precision floating-point register using the new paired-single (PS) format. Several new instructions can then operate on this data in parallel, effectively doubling performance in this mode. Since many 3D graphics applications, as well as some scientific software, use single-precision FP, most of these applications could see a boost from MIPS V. The company would not discuss what processors will implement MIPS V, but we expect the follow-on to the R10000, code-named H1, will do so.

MDMX, also known as Mad Max, is an optional set of instructions similar to Intel's MMX *(see* **100301.PDF***)* in that they define a set of media registers that is mapped onto the FP registers, new data types that store 8- and 16-bit data in parallel in the media registers, and instructions that operate on this data in parallel. MDMX's unique twist is its 192-bit accumulator that allows integer multiplication and accumulation to occur without any overflows or loss of precision.

Digital's additions are more spare, in keeping with the minimalist nature of the Alpha instruction set. Digital engineer Pete Bannon argued that current Alpha processors are fast enough to handle relatively simple tasks like audio mixing and video decoding without any instruction-set extensions, so why add new instructions that could slow the decoders or the execution units? To meet a 2-ns cycle time, simplicity is a requirement. The new instructions are designed to speed video encoding, a much harder problem, without slowing the CPU on other tasks.

These two vendors join HP, Intel, and Sun in adding multimedia extensions to their instruction sets. Large performance gains on multimedia applications, coupled with the relatively small incremental hardware cost, have led to this widespread adoption. Of the major desktop processors, only PowerPC now lacks such extensions, an incredible oversight given Apple's focus on multimedia.

MIPS V Boosts FP Performance

Most of these extensions aim to improve performance when handling 8- and 16-bit integers. These data types are common in audio, video, and graphics applications, yet traditional ALUs can operate on only one integer at a time. With most processors now implementing 64-bit data paths, up to seven-eighths of this data path is wasted when operating on small integers. Packing four or eight small integers into a single 64-bit register and operating on them in parallel using a SIMD (single instruction, multiple data) approach greatly increases throughput.

MIPS realized a similar opportunity exists on the floating-point side, yet no other vendor has moved to seize it. Modern microprocessors are optimized to handle double-precision floating-point data, meaning that a 64-bit data path exists in the FPU. Yet single-precision floating-point data is common in many applications, including 3D graphics and signal processing; these applications waste half of the FP data path. The same SIMD approach can double throughput on single-precision data.

MIPS V does just that. The new PS format pairs two single-precision values in each FP register. Table 1 lists the

At the Microprocessor Forum, several instruction-set architects gathered to discuss extensions for multimedia processing, including (l to r) Pete Bannon of Digital, Earl Killian of MIPS Technologies, Ruby Lee of HP, Uri Weiser of Intel, and Marc Tremblay of Sun Microelectronics. At right is moderator Linley Gwennap of MicroDesign Resources.

Modified Instructions	
ADD, SUB, MUL, ABS, MOV, NEG	Basic computational operations
MADD, MSUB, NMADD, NMSUB	Multiply add/subtract
C.cond	Parallel compare
MOVF, MOVT	Conditional move
New Instructions	
LUXC1, SUXC1	Load 8 bytes without alignment
ALNV	Realign data
PLL, PLU, PUL, PUU	Rearrange PS data
CVT.PS.S	Convert to/from PS format

Table 1. MIPS V modifies existing FP instructions to use the new paired-single (PS) data type and adds a few new instructions for packing, rearranging, and unpacking PS data.

existing FP instructions that accept PS operands in MIPS V; these include the basic arithmetic operations. Note that the product of two single-precision operands is still a single-precision value, with the exponent adjusted accordingly. Thus, multiplying two paired-single operands results in a paired-single product.

The table also lists a few new instructions in MIPS V. The LUXC1 and SUXC1 instructions (don't try to pronounce them!) load and store 64 bits at a time regardless of the alignment of the address; that is, the lowest three bits of the address are simply ignored. These FP instructions help load pairs of single-precision operands when the operands are part of a vector that is not aligned to a 64-bit boundary. The ALNV instruction can then properly align the data.

The Pxx instructions are useful for copying the upper or lower half of a PS value to the upper or lower half of another PS value. Finally, a new convert (CVT) instruction creates a PS value from two single-precision values. This instruction takes two FP registers as source operands.

These instructions will add a small amount of circuitry, particularly when compared with the size of a high-performance double-precision FPU, yet they will provide a big performance boost on some frequently used algorithms, such as fast Fouriér transform (FFT) and matrix multiplication.

MDMX Goes Beyond Intel's MMX

The basic features of MIPS' MDMX are similar to those of Intel's MMX. MDMX creates a new set of 32 media registers, each 64 bits wide. To reduce storage requirements, they are mapped onto the floating-point registers. Intel uses the same strategy for its MMX registers, but since x86 has only eight FP registers, there are also only eight MMX registers. MDMX adds a set of eight single-bit condition flags, which map onto the MIPS FP condition flags.

The new registers support two data formats: oct byte (OB) and quad half (QH). For the jargon-impaired, the former refers to eight 8-bit values packed into a single 64-bit register, while the latter consists of four 16-bit values. Like MMX, MDMX can boost performance on many multimedia algorithms by 2–4×.

Table 2 lists the standard MIPS instructions that are modified to operate on the media registers using either the

Modified Instructions	
ADD, SUB, MUL, MIN, MAX, MSGN	Saturating arithmetic
AND, XOR, OR, NOR, SLL, SRL, SRA	Logicals and shifts
ALNI, ALNV	Align vectors
C.EQ, C.LT, C.LE	Compare bytes
New Instructions	
SHFL.op	Shuffle bytes
PICKF, PICKT	Combine vectors
ADDL, SUBL, MULL, MULSL	Store result in ACC
ADDA, SUBA, MULA, MULS	Operate on ACC
RZU, RNAU, RNEU, RZS, RNAS, RNES	Round ACC
RAC, WAC	Read/write ACC

Table 2. The MIPS MDMX extensions modify several instructions to use the new vector data types and add several new instructions for accessing the 192-bit accumulator and for arranging data.

OB or QH format specifier. These include arithmetic, logical, shift, and min/max instructions. The new variations always operate in saturating mode, where overflows and underflows are clamped at the maximum and minimum values, respectively. This mode is useful when operating on pixel or amplitude data. These features are also found in MMX.

MIPS calls these operations "vector-to-vector" arithmetic. MDMX also supports a vector-to-scalar mode. In this mode, a single byte (or halfword) selected from any part of a media register is combined with each byte of the second source register, as Figure 1b shows. A third mode allows the scalar value to be specified as a 5-bit immediate value.

The vector-to-scalar modes come in handy when multiplying a vector by a constant value, which occurs in inverse discrete cosine transformation (IDCT) and many signal-processing algorithms. These modes are also useful for motion estimation and other multimedia algorithms.

New Instructions Arrange Bytes

The shuffle (SHFL) instruction, similar to MMX's PUNPACK, performs eight different byte rearrangements. It can convert data between OB and QH formats with signed and unsigned options and can interleave bytes from two registers. Unlike the PERMUTE instruction in HP's MAX2 (see sidebar, page 4), SHFL does not perform arbitrary reorganization.

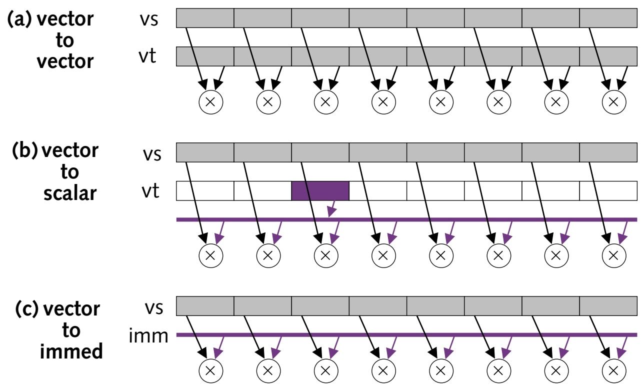

Figure 1. MDMX instructions operate in (a) vector-to-vector mode, (b) vector-to-scalar mode, and (c) vector-to-immediate mode.

For More Information

For more information on these multimedia instruction-set extensions, check the following Web pages. For MIPS V and MDMX, *www.mips.com/ISAV/index.html.* For MAX2, *www.hp.com/wsg/strategies/strategy.html.* For MMX, *www.intel.com/pc-supp/multimed/mmx/index.htm*. For VIS, *www.sun.com/sparc/vis*. (No Web information is yet available for Digital's MVI.)

Other MDMX instructions go beyond the capabilities of MMX. The align (ALNx) instructions can extract eight bytes out of a sequence of bytes in two different source registers. These instructions can realign bytes from an arbitrarily aligned data stream.

Three compare (C.xx) instructions perform a parallel comparison and set the condition flags. There are eight condition flags defined, so each corresponds to a single byte in OB format (only four are used in QH format).

The PICK instructions can then be used to combine the individual values in two registers depending on the contents of the condition bits. If a condition bit is set, the corresponding byte is copied from the first source register; otherwise it is taken from the second source register. The C.xx instruction is similar to the parallel compare (PCMPxx) instruction in MMX, but MMX requires a sequence of three instructions to do the same task as a single PICK.

MDMX Adds Wide Accumulator

The wide accumulator is a completely different approach to multiplication than used in other instruction sets. The fundamental problem: when two integers are multiplied, the product may have twice as many significant bits as the operands. For example, it takes 16 bits to hold the product of two arbitrary 8-bit values. The problem gets worse for a multiply-accumulate operation, common in many signal-processing algorithms. Accumulating, say, one hundred 16-bit products could require 23 bits to avoid any possible overflow.

In MMX, 16-bit data is "promoted" to 32 bits by the multiply-accumulate (PMADDWD) instruction. This restricts the available parallelism once the data is promoted and still doesn't provide enough bits to avoid overflows. Also, there is no corresponding MMX instruction for 8-bit multiply-add.

MDMX solves all of these problems with its 192-bit accumulator. This special register can be partitioned into eight 24-bit values or four 48-bit values, corresponding to the OB and QH formats. In the former case, each 24-bit value can accumulate the product of 256 multiplies of 8 × 8 bits each. In QH mode, the accumulator can handle the sum of 65,536 multiplies of 16 × 16 bits each. In both cases, there is no loss of precision and no possibility of overflow.

Table 2 lists several new instructions that use this accumulator. Vector data can be added, subtracted, and multiplied, with the result placed in the accumulator with no loss of precision. While this is most useful for multiplication, vector data can be added and subtracted in nonsaturating mode, with potential overflows and underflows captured in the accumulator for further processing.

Vector data can also be added directly to the accumulator register, or multiplied and accumulated. There are several instructions that shift and round the final result. Finally, the RAC and WAC instructions move data between the accumulator and the media registers, where it can be copied to memory. These instructions, which access the accumulator in three 64-bit chunks, are used for saving state.

Single Accumulator Can Be a Bottleneck

At the Forum, MIPS architect Earl Killian said the accumulator obviated the need for a parallel 32-bit data type, which is supported in MMX and Sun's VIS. This data type is mainly used to provide adequate precision when operating on 16-bit values; the MDMX accumulator provides the same function. A few algorithms, such as Dolby Digital AC-3 audio, require data with more than 16 bits of precision (AC-3 uses 20 bits); MDMX comes up short for these algorithms, although a MIPS V processor can double throughput if the algorithm is converted to single-precision FP data.

While the accumulator provides the advantages noted above, only one set of calculations can use the accumulator at a time. Other architectures can unroll loops and accumulate into several general-purpose registers at once. Thus, the single accumulator can be a bottleneck. Another problem with the accumulator is that the operating system must be modified to save additional state, which could be a problem in an embedded system using an off-the-shelf real-time OS.

The MDMX extensions are more extensive than the MIPS V instructions and will require somewhat more hardware to implement. If the floating-point registers and data path are leveraged, however, the impact is still fairly small. The 192-bit accumulator must be added, but the other changes require only a few extra buses and multiplexers along with minor tweaks to the arithmetic and shift units.

Although MDMX is optional, MIPS expects it to be used in a variety of processors for computer systems and embedded products. A few specialized embedded processors may not implement MDMX for cost reasons.

Digital Boosts Motion Video

The high native performance of Digital's 21164 processor allows it to perform full DVD decoding (MPEG-2 video and AC-3 audio), as well as a variety of lesser tasks, in software without any special instructions. The only significant remaining hurdle, according to Digital's Bannon, is MPEG-2 video encoding, which overwhelms even a 21164. Analysis of MPEG-2 encoding software showed that a single task, motion estimation, consumes more than 70% of the CPU cycles. Improving performance on this one task would solve the encoding problem for Digital.

New Instructions	
PERR Ra, Rb, Rc	The sum of the absolute difference of each of the 8 bytes in Ra and Rb is written to Rc
UNPACK Rb, Rc PACK Rb, Rc	Unpack bytes into halfwords (4) or words (2) Pack words (2) or halfwords (4) into bytes
MIN Ra, Rb, Rc MAX Ra, Rb, Rc	For each of the 8 bytes in Ra and Rb, write the smaller (MIN) or larger (MAX) to Rc

Table 3. Digital's MVI extensions add only a few instructions that are designed primarily to boost video-encoding performance.

Instead of using the floating-point registers (or a new set of registers mapped to the FP side), Digital chose to add a few integer instructions. This simpler approach, also taken by HP, allows existing instructions such as AND to operate on parallel data. Unlike Pentium, Alpha already has 64-bit integer registers, the same width as the FP registers.

As others have done, the MVI extensions define new formats to pack 8-, 16-, and 32-bit data into the integer registers. The number of special instructions for operating on these data types is, however, quite small, as Table 3 shows.

The key instruction is PERR, which calculates the sum of the absolute differences of eight pairs of bytes, a function also implemented in Sun's VIS. This function is at the heart of motion estimation, which involves comparing blocks of pixels (typically 8 × 8) to find which pair has the smallest difference. A single PERR instruction replaces nine normal Alpha instructions in the motion-estimation inner loop, providing a significant speedup.

No Parallel Arithmetic Instructions

The new PACK and UNPACK instructions convert packed bytes into 16- or 32-bit formats and vice versa. This conversion allows operations on 8-bit data. Digital did not, however, provide any parallel arithmetic instructions except for the new MIN and MAX instructions, which are helpful in video encoding. In a few situations, standard instructions can be used to operate on parallel data, but only if the algorithm avoids all overflows or underflows. This exclusion prevents most algorithms from using parallel arithmetic.

The MVI instructions will help the forthcoming 21264 perform MPEG-2 encoding in real time, a feat beyond the capabilities of any current microprocessor. MVI's impact on other multimedia tasks will be small. Video encoding to less stringent standards, such as MPEG-1 or H.320 video conferencing, will consume a smaller percentage of the enhanced 21164PC than the 21164. With no multiply-accumulate instruction or parallel arithmetic, tasks such as audio processing and communications will see little gain, if any, but these tasks already require little effort from the Alpha chips.

MVI also neglects two performance-critical multimedia tasks: 3D graphics and image processing. Both of these tasks can consume all available performance even on a powerful processor. MVI lacks the dual-FP instructions found in MIPS V, which help 3D. An image-processing application like Photoshop would benefit from parallel arithmetic during pixel manipulation, but MVI has none. This shortfall will reduce the gap between Digital and Intel processors on optimized image-processing applications.

Other Instruction Sets Evolve

At the Forum, HP's Ruby Lee discussed extensions to the PA-RISC architecture that are now shipping as part of the PA-8000 processor. These extensions, called MAX2, are a superset of the original set of extensions *(see* 080103.PDF*)* implemented in the PA-7100LC and PA-7200. These 32-bit processors can operate on two 16-bit halfwords at once; the 64-bit PA-8000 can operate on four halfwords in parallel.

MAX2 also adds two new instructions. MIX, similar to Intel's PUNPACK, interleaves halfwords from two registers, speeding matrix transposition. PERMUTE goes further by allowing any rearrangement, with or without repetition, of the four halfwords in a register. These instructions are useful in the IDCT operation (part of video decoding) and in cryptography, for example.

The VIS instruction set *(see* 081604.PDF*)* has been on the market for about a year; according to Sun's Marc Tremblay, the company is working on improving it in future processors. One feature Tremblay is strongly considering is parallel FP operations, similar to those in MIPS V. Look for these and other new features in future UltraSparc processors.

Intel's Uri Weiser, who led the development of MMX, admitted he is also looking at future upgrades. Although Weiser would not comment on any specifics, the company is widely rumored to be planning extensions called MMX 2, which could begin shipping in processors as early as 1H98 *(see* 1015MSB.PDF*)*.

Vendors Excel in Different Areas

Because multimedia is a diverse concept, comparing these instruction sets is not a simple matter. The best approach is to look at different multimedia applications and assess each instruction set on that application. Each vendor has targeted some applications and neglected others.

For basic manipulation of audio and pixel data, the new extensions (except for MVI) are fairly similar, providing packed 8- and 16-bit data types along with instructions that operate on them with saturating and nonsaturating arithmetic. As Table 4 shows, Sun neglects saturating arithmetic, while HP doesn't support 8-bit arithmetic. Intel offers only two-operand instructions, whereas the RISC architectures support a separate destination specifier. Image processing (e.g,. Photoshop) is the key application in this category.

Rendering 3D images is also image processing and takes advantage of these same basic extensions. Most vendors agree, however, that 3D rendering is best done by a hardware accelerator and not on the CPU.

	MIPS V/ MDMX	Intel MMX	Sun VIS	HP MAX2	Alpha MVI
No. of Registers	32	8	32	31	31
Register Type	MM/FP†	MM/FP†	MM/FP†	Integer	Integer
Parallel Arithmetic	8 × 8 bits 4 × 16 bits	8 × 8 bits 4 × 16 bits	4 × 16 bits 2 × 32 bits	4 × 16 bits	Min/max only
Unsaturating?	No	Yes	Yes	Yes	n/a
Saturating?	Yes	Yes	No	Yes	n/a
Three Operands?	Yes	No (2)	Yes	Yes	Yes
Parallel Multiplies	4 or 8	4	4	None	None
Multiply/Add	8 × 8 → 24 16 × 16 → 48	16 × 16 → 32	8 × 16 → 16	Shift-and-add	None
Vector-to-Scalar?	Yes	No	No	No	No
Parallel Shifts?	Yes	Yes	No	Yes	No
Parallel Average?	No	No	No	Yes	No
Parallel Compare?	Yes	Yes	Yes	No	No
Pack/Unpack?	Yes	Yes	Yes	Yes	No
Interleave?	Yes	Yes	Yes	Yes	No
Permute?	No	No	No	Yes	No
Pixel Error?*	No	No	Yes	No	Yes
Block Load/Store?	No	No	Yes	No	No
Parallel FP?	Yes	No	No	No	No

Table 4. The combination of MIPS V and MDMX offers more features than other multimedia instruction sets, while Alpha's MVI offers fewer. *motion-estimation instruction. †multimedia registers mapped on floating-point register set. (Source: vendors)

Most audio applications require some type of signal processing, such as FFTs and other matrix arithmetic. The key instructions for these algorithms are multiply and multiply-accumulate. MDMX offers the most flexibility here, particularly with its unique 192-bit accumulator, while MMX is close behind. HP's parallel shift-and-add instruction can multiply by only small constant values. In addition to audio, these instructions are good for modem emulation.

Intel's CISC heritage shows up in its lack of MMX registers: only 8 compared with 32 for the other instruction sets. While more registers can improve performance on a range of applications, the extra registers are particularly helpful for matrix arithmetic. With 32 registers, an entire 4 × 4 transformation matrix can be held in half the register set, leaving the remainder free for working data. Since the same transformation is often used for many operations in a row, this ability greatly reduces cache accesses compared with the limited MMX register file.

Decompressing audio and video, such as MPEG-2 decoding, requires different operations, such as IDCT. While these operations make heavy use of multiply-accumulate, they also benefit from instructions that perform byte rearrangement for matrix transformations. All of the vendors but Digital have added such instructions.

Compressing video relies almost entirely on motion estimation. Both Digital and Sun have supplied SAD (sum of absolute differences) instructions to speed this function. Both MIPS and HP claim the motion-estimation problem can be solved in a variety of ways without resorting to a special hardwired instruction. Until these vendors can demonstrate video compression, evaluating these claims is difficult, but hardwired acceleration has proved useful in other cases. Intel has not addressed this problem.

The CPU's main role in 3D graphics is geometry processing. This processing is typically done using single-precision (sometimes double-precision) floating-point math and thus is unaffected by most multimedia extensions. Only MIPS V, with its paired-single operations, provides a large performance boost for 3D geometry. Sun's specialized 3D array instructions assist some 3D applications.

Many of these algorithms require fast access to large data sets. The block load and store instructions in VIS speed these accesses and are unique among these processors. Of course, access times remain largely a function of the processor's memory bandwidth.

Time to Market Is Important

Instruction-set design is important, but other factors will affect the ultimate success of these new extensions. Software must be changed to take advantage of the new instructions, and for the most part, this change involves rewriting the code by hand to operate in a SIMD fashion; the compiler cannot revise the algorithms. Without new software, the finest instruction sets are useless.

Of the current vendors, HP was the first to deliver multimedia extensions, starting with the PA-7100LC in 1Q94. Today, only a few applications take advantage of these extensions, primarily for MPEG decoding and video conferencing. Sun has shipped the most multimedia-enabled processors to date and is slowly building a software base. Intel has been helping software vendors recode to MMX for more than a year, but the first MMX systems will not ship until 1Q97. The Digital and MIPS extensions are further in the future. Digital's software task is easier: the company will probably release a new video-encoding library and call it a day.

Of course, these extensions will be no more popular than the underlying processors, making MMX the clear winner here. MIPS V will solidify Silicon Graphics' position as the pre-eminent 3D platform. Combining high intrinsic performance and the MVI extensions, Alpha is positioned to become the best video-authoring platform. Digital's lead in image-processing performance, however, may shrink. The biggest loser of all could be PowerPC, which apparently still has its collective head in the sand while other vendors, particularly Intel, are poised to take share in markets where multimedia is important. M