From the companion CD-ROM to the IEEE CS Press book,
"The Anatomy of a Microprocessor: A Systems Perspective,"
by Shriver & Smith

Keith Diefendorff
Apple Computer

Pradeep K. Dubey
IBM T.J. Watson Research Center

How Multimedia Workloads Will Change Processor Design

Media processing will force profound changes in current microprocessors. Yet general-purpose processors can and will adapt, making specialized digital signal processors for media applications essentially irrelevant.

For the last two decades, general-purpose processor design has been driven largely by non-real-time, standalone applications. But we believe applications will increasingly integrate new multimedia component technologies that will eventually shift the demand from traditional MIPS to media MIPS. By "MIPS," we mean traditional computational horsepower; by "media MIPS" we mean media-specific horsepower. We also expect general-purpose processors, enabled by the continued exponential growth in semiconductor technology, to evolve to accommodate this demand. In fact we expect them to be able to process real-time, vector-oriented, media data as easily as they now process traditional scalar data. Thus, general-purpose processors will win out over the specialized digital signal processors now used for media processing.

STATIC TO DYNAMIC MEDIA

Applications for the last 20 years have been largely nonnumeric with little inherent parallelism. These applications were characterized by the scalar processing of 32-bit integer data types, complex control flow, short data dependence distances, and good instruction and data locality. To run these applications efficiently, modern superscalar processors have evolved to emphasize scalar processing, control-flow speculation, dynamic instruction reordering, and large caches. The complexity and the clock frequencies of these processors continue to grow exponentially. However, as they consume more and more of the available program parallelism, and as the gap between their internal speed and memory speed increases, performance growth will become increasingly restricted.

A recent confluence of hardware and software technologies has given computers the ability to process dynamic media data (video, animation, music, and so on) where before they could handle only static media data (images, print, sounds, and so on). Computer users are just beginning to recognize the potential improvement in usability, quality, productivity, and enjoyment that comes from dynamic media. And software developers are moving rapidly to capitalize on this trend by providing applications that offer new, compelling functionality. We expect this cycle to continue, indeed to snowball. This will fundamentally change the nature of almost all computer applications.

As new dynamic multimedia technologies become integrated into applications, the average processing workload will change significantly. Dynamic multimedia component technologies such as videoconferencing, image compression, video authoring, image processing, visualization, 3D graphics, animation, realistic simulation, the Virtual Reality Modeling Language, encryption, speech recognition, and broadband communications will shift demand from traditional MIPS to media MIPS. This, in turn, will profoundly influence the design of the processors that run the applications.

In contrast to traditional applications, multimedia-rich applications will be primarily peer-to-peer, involve significant real-time processing of continuous media data streams, and heavily use vectors of packed 8-, 16-, and 32-bit integers and floating-point numbers.

0018-9162/97/$10.00 © 1997 IEEE

The very nature of many media applications is such that a non-real time response may simply be unacceptable.

WHY GENERAL-PURPOSE PROCESSORS?

We expect general-purpose processors to eventually replace specialized digital signal processors for media processing. We see no semiconductor technology, or microarchitectural limitations to integrating the ability to handle dynamic media into general-purpose microprocessors. With relatively simple architectural support (Intel's MMX, Sun Microsystems' VIS for SPARC, Silicon Graphics' MDMX for MIPS V, Digital's MVI for Alpha, and Hewlett-Packard's MAX2 for PA-RISC), a very significant speedup of media-intensive processing in general-purpose processors is possible. Media processing power comparable to, or even exceeding, special-purpose hardware or solutions based on digital signal processing can be achieved. We further believe that this level of media performance—along with tight, fine-grained, zero-overhead coupling of media processing power into the general-purpose application processor—spells the demise of special-purpose media accelerators in computers (although we expect such processors to remain viable in embedded consumer devices such as DVD players).

CHARACTERISTICS OF MULTIMEDIA-CENTRICITY

To understand the complexity of future potential multimedia scenarios, consider a scenario in which the user of a portable computer, sitting at a beach, is able to select one of hundreds of satellite channels for a live broadcast of a game, and simultaneously hold a video conference with her remotly located friends to discuss the intricacies of the game, without losing the ability to respond to any incoming fax or phone call. Such a scenario would break down into the real-time encoding and decoding of multiple video and audio streams, including encryption/decryption and error correction. The video-stream processing (in the spirit of MPEG-4) may also imply real-time 3D transformations. Audio-stream processing could include real-time spontaneous speech recognition to enable interesting searches (indexing) of the content. The MIPS and bandwidth needs of such a scenario (tens of billions of operations per second) are far beyond that of any current portable computer, but we believe them to be within reach in the next five years.

We see several distinguishing characteristics of these multimedia-centric applications that will have profound implications for future processor design.

Real-time response

Multimedia applications such as video conferencing or electronic commerce often, by their very nature, require real-time response. The correctness of processing with media applications, unlike many scientific numeric applications, is less of a quantitative match against some deduceable output, and more of a qualitative perception in real time. Thus, the very nature of many media applications is such that a non-real-time response may simply be unacceptable. For example, an application may benefit more by skipping a frame of video and maintaining a certain throughput in frames per second than by processing with no loss but below a certain throughput threshold.

Continuous-media data types

The input data for multimedia applications often comprises a set of data elements derived from sampling some analog signal in a time domain—either video, audio, or other sensory perception. This is fundamentally different from the noncontinuous media data types prevalent in nonmedia applications, such as the input data to a spreadsheet or a word processor. Driven by the need for a larger numerical range, the noncontinuous media integer data type is mostly 32 bits wide and is expected to increase to 64 bits in the near future. However, what human senses can discriminate is a much narrower range: for many continuous media integer data types it is only 8 or 16 bits. This is a significant waste of data and computational bandwidth for machines with 32-bit- or 64-bit-wide datapaths and architectural support limited to 32-bit or 64-bit scalar integer data types. This problem is exacerbated when even smaller subword data types are handled. An example is the bit-data-type processing that occurs during the real-time handling of very high data rate bitstreams carrying (potentially encrypted) media data over the network.

Significant fine-grained data parallelism

Data parallelism is inherent in almost all signal processing and graphics applications. Input data streams are frequently large collections of small data elements such as pixels, vertices, or frequency/amplitude values. The parallelism in these streams is fine grained. And because elements of these large input data streams tend to undergo identical processing (filtering, transformations, and so on), it lends itself to machines with SIMD hardware units executing in parallel. This is a very different processing paradigm than extracting fine-grained instruction-level parallelism amid complex control and data dependences. The latter is the processing paradigm for all modern superscalar processors and is directly responsible for their high degree of complexity. gcc, a SPECint95 benchmark, is a good example. Significant speedup on this benchmark is often used to justify the complexity of processors with high degrees of superscalar instruction issue, deep instruction lookahead hardware, out-of-order execution, register renaming, and multiple levels of control speculation. But for media processing, simple SIMD execution units with wide data paths would be able to achieve significant speedups without this enormous complexity.

Significant coarse-grained parallelism

Most media applications and scenarios consist of more than one time-critical execution thread. For example, a typical videoconferencing application consists of video encoding and decoding, audio encoding and decoding, and background task threads. These independent threads of execution provide many opportunities for the effective application of both temporally and spatially parallel hardware. Very high frequency, deeply pipelined, and hardware multithreaded processors and symmetric multiprocessors can be brought to bear in ways simply not possible in the stand-alone context of SPECint95 or traditional general-purpose computing applications.

High instruction-reference locality to small loops

Signal- and image-processing applications often consist of small loops or kernels that dominate overall processing time. Within these loops and kernels, instruction references tend to be concentrated and hence exhibit good spatial and temporal locality. Relative to non-media applications, this yields a much higher degree of correlation between overall application speedup and loop/kernel speedup. Furthermore, even without good compiler technology, manual assembly coding and hand optimization of such tight loops is practical.

High memory bandwidth

Typical data sets and working sets for many media applications, especially 3D graphics, are huge. This implies that processors must provide very high memory bandwidth and must tolerate long memory latency. Existing and even future caches will not be large enough to handle these data sets. Cache performance is further degraded by the poor or nonexistent locality of the data. While handling such media data, the cache gets polluted rapidly, making it less effective on other tasks in execution. Consequently, data prefetch and cache bypass schemes become even more important.

High network bandwidth

A thousandfold speedup in processor clock or memory data rate is not possible in the next five to 10 years. But such a speedup is a distinct possibility at the network interface because the communication infrastructure of typical households is rapidly evolving from modems that operate at kilobits per second to cable, ADSL (asymmetric digital subscriber line), and RF modems that operate at megabits per second. The software implementation of such high-speed modems may require new hardware and instruction-set architecture primitives aimed at real-time bitstream processing, filtering, and error detection/correction.

Extensive data reorganization

Although the fine-grained data parallelism in media applications is vector or SIMD in nature, traditional rigid SIMD architectures will not show significant speedup over scalar architectures without capabilities for extensive data reorganization. These capabilities will enable SIMD architectures to adapt to a variety of input datastream layouts. Also wide data paths have an inherent architectural advantage during data reorganization operations. For example, reversing the elements of a packed data vector has a much shorter instruction path length on a wide SIMD execution unit than on two superscalar execution units, each with half the data-path width.

Over the next five years, we believe that media processing will become the dominant force in computer architecture and microprocessor design.

Over the next five years, we believe that media processing will become the dominant force in computer architecture and microprocessor design. The dramatically different nature and demands of media processing will force profound changes, yet we expect general-purpose processors can and will evolve to retain their current role as *the* processing element. Indeed, we expect the entire computer system to be put on a single chip, driven by insatiable performance demands and the physical difficulty of achieving them with discrete components. In five years, the exponential growth in semiconductor technology will put several hundred million transistors on a single die and operate them at speeds well in excess of a gigahertz. In this scenario, we believe specialized media digital signal processors will simply become extinct, laid victim by phenomenal technology growth. ❖

Keith Diefendorff *is a distinguished scientist and director of microprocessor architecture and strategy at Apple Computer, where his primary interest is high-performance microprocessor architecture. The chief architect of Motorola's PowerPC and of the 88110 superscalar microprocessor and its SIMD graphics instruction set, he is currently working on the architecture and definition of future PowerPC microprocessors with IBM and Motorola. Diefendorff received an MSEE from the University of Akron and is a member of the IEEE.*

Pradeep K. Dubey *is a research staff member at the IBM T.J. Watson Research Center, where he is now working on topics related to systems aimed at emerging multimedia applications and processors based on multiple control flows. He has worked on the design, architecture, and performance modeling of various microprocessors, including Intel's 80386, 80486, and Pentium. He has published extensively and filed several patents in computer architecture. Dubey received a BS in electronics and communication engineering from Birla Institute of Technology, an MSEE from the University of Massachusetts at Amherst, and a PhD in electrical engineering from Purdue University. He is a senior member of the IEEE.*

Contact Diefendorff at Apple Computer, 1 Infinite Loop, MS 60-PEG, Cupertino, CA 95014; keithd@apple.com; or Dubey at IBM T.J. Watson Research Center, MS 39-154, Yorktown Heights, NY 10598; pradeep@watson.ibm.com.