Media-Friendly Microprocessor Architectures and Tools

By Ralph Biesemeyer

Video, audio, and graphics are increasingly important elements of digital communication. The chip industry is designing microprocessors to better handle new software functions that support these elements. Software engineers require better tools to minimize development time and maintain platform compatibility for media applications. Revolutionary system architectures allow clustered desktop processors to become fault-tolerant high-speed server and rendering systems. Recent improvements in these three areas can help engineers build COTS (Commercial Off-the Shelf) open platform media systems that perform and scale better than ever before.

What does a microprocessor do? According to Little Man Computer, devised in 1965 at MIT by Stuart Madnick (Fig. 1), only nine things: load, store, add, subtract, read, write, halt, skip on condition, and jump.[1] Modern processors still do those same nine operations, albeit faster, so they can do a lot more in the same amount of time.

In 1981 the 8086 microprocessor ran those nine instructions at a clock rate of 4.77 MHz. The popularity of that processor wrapped in a personal computer framework was astounding. Seven years after its introduction, there were hundreds more software titles for the 8086 than all the programs developed for digital computers before 1981. Eighteen years later, a DOS 3 1/2-in. boot diskette can be inserted into the A: drive of a 2 GHz Pentium 4 processor PC: by booting DOS 2.1, the same program from 1983 can run. It will really run!

New processors support a software legacy to increase the value of existing software and provide tool continuity to develop new software. But to achieve higher performance with more demanding applications while maintaining backward compatibility, significant microprocessor ingenuity is required.

Today's video, audio, graphics, and multimedia applications are pushing previous generation digital architectures to the limit. The proliferation of digital 3-D, video, animation, and high-quality audio files has resulted in an exponential growth of the amount and complexity of data flowing through today's PC.

Microprocessor designs keep pace with this data not only from sheer transistor density and megahertz improvements, but also changes to the micro-architectures, which enable new ways of handling large amounts of complex data in more efficient ways. Higher-performance microprocessors permit more accurate simulations, more realistic animations, and better modeling of more sophisticated 3-D graphics.

The latest generation 32-bit processor has 42 million transistors and features a micro-architecture, which provides hyper-pipelined technology to allow for higher frequencies and rapid instruction execution. Doubling the length of the execution pipeline from 10 to 20 processes, the new micro-architecture also features an enhanced advanced dynamic execution engine that improves the branch prediction capabilities, thus reducing the penalties traditionally associated with deeper pipelines.

The micro-architecture also adds an execution trace cache that reduces the overall time required to recover from a branch that has been mispredicted.[2] The processor's arithmetic logic unit runs at twice the core frequency, allowing it to execute commands in one-half a base clock cycle, and it has a 400-MHz system bus that operates at 3.2 Gbits/sec,

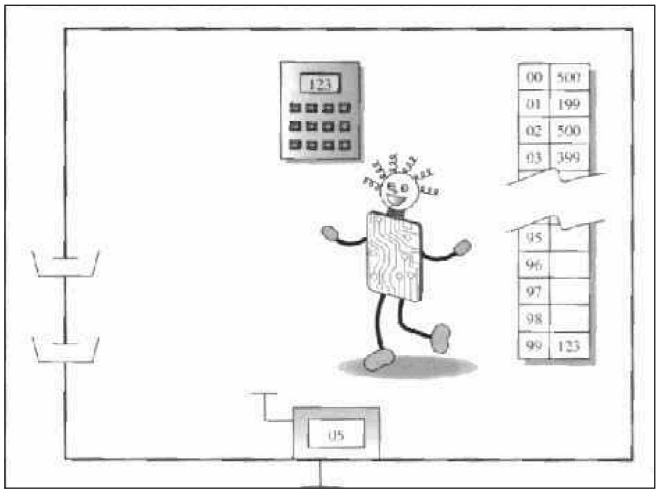

Figure 1. Little Man Computer has an inbox, outbox, and calculator. He reads his task list and executes them one at a time, incrementing the program command counter.

Presented at the 143rd SMPTE Technical Conference (paper no. 143-S-2), in New York City, November 4-7, 2001. Ralph Biesemeyer is with Intel Corp., Beaverton, OR 97006. An unedited version of this paper appears in *Pixels, Packets, Processing, and Infrastructure,* SMPTE, 2001.

which is over three times faster than its predecessor.

Media Tuned Instruction Set

Above and beyond these increases in pipeline depth, memory transfer, and clock speed, software developers who are always finding ways to challenge faster processors, have a more powerful instruction set, which performs similar operations in a parallel fashion. This has improved instruction throughput up to eightfold.[3]

Microprocessor architects and software developers collaborated on defining the 144 new instructions. They analyzed a range of software applications, including internet applications, games, 2-D and 3-D graphics, video compression/decompression, image processing, digital-content creation, and streaming video. The results showed common characteristics across the seemingly diverse software applications: computationally intensive tasks that use regular and recurring memory-access patterns and localized, recurring operations performed on the data.

The full set of instructions are referred to as the IA-32 SIMD (Single-Instruction Multiple-Data) technologies. They include the Intel MMX technology and the SSE and SSE2 extensions. SIMD gives a programmer the ability to develop algorithms that can combine operations on packed 64 and 128-bit integer and single and double-precision floating-point operands. The evolution of these instructions is summarized in Table 1.

The SSE and SSE2 instruction sets exploit high-speed memory by introducing a set of cache-ability and memory-ordering instructions that can improve cache usage and application performance. This capability improves the performance of 3-D graphics, speech recognition, image processing, scientific, and other multimedia applications.

Figure 2 shows how use of only the SSE2 instruction set can provide a 35% increase in performance in an MPEG-4 encoding process—from 14.03 frames/sec (bar 4 P4 1.5 GHz x87 FP iDCT) to 18.96 frames/sec (bar 3 SSE2 iDCT). Migrating the encoding application from 1 GHz PIII (bar 5), which does not use the SSE2

Table 1—The Evolution

MMX™ Technology	Streaming SIMD Extensions	Streaming SIMD Extensions 2
Introduces 64-bit MMX™ brand registers.	Introduces 128-bit XMM registers.	
Introduces support for SIMD operations on packed byte, word, and doubleword integers.	Introduces 128-bit data type with four packed single-precision floating-point operands	Adds 128-bit data type with two packed double-precision floating-point operands. Adds 128-bit data types for SIMD integer operation on 16-byte, 8-word, 4-doubleword, or 2-quadword integers
	Introduces data prefetch instructions.	
	Introduces non-temporal store instructions and other cacheability and memory ordering instructions	Extends support for cacheability and memory ordering operations. Extends support for data shuffling.
	Adds extra 64-bit SIMD integer support.	Adds support for SIMD arithmetic on 64-bit integer operands
		Adds instructions for converting between new and existing data types.

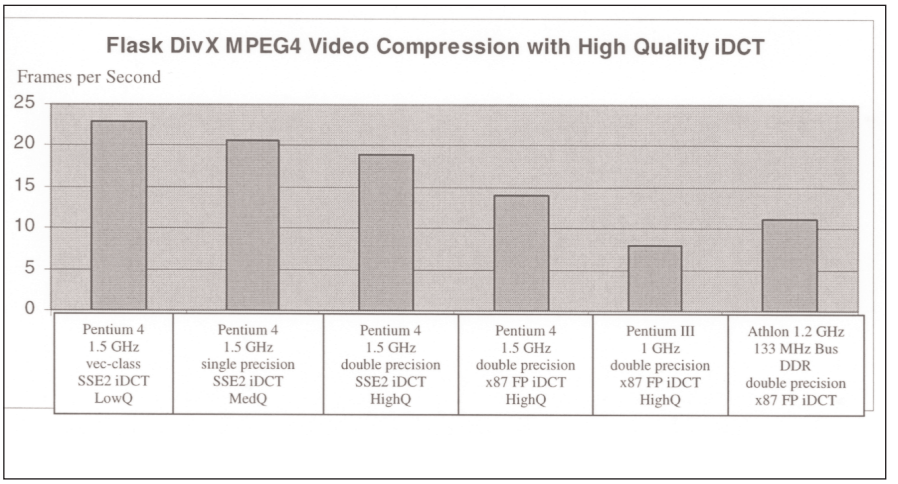

Figure 2. Use of SSE2 instruction set can provide a 35% performance increase in an MPEG-4 encoding process.[4]

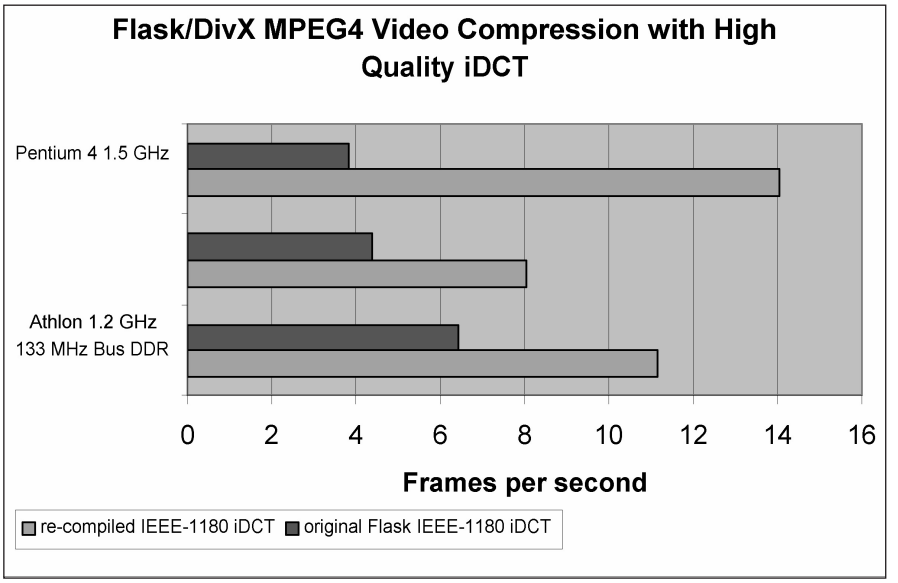

Figure 3. MPEG-4 compression software runs more than 300% faster after re-compilation.[5]

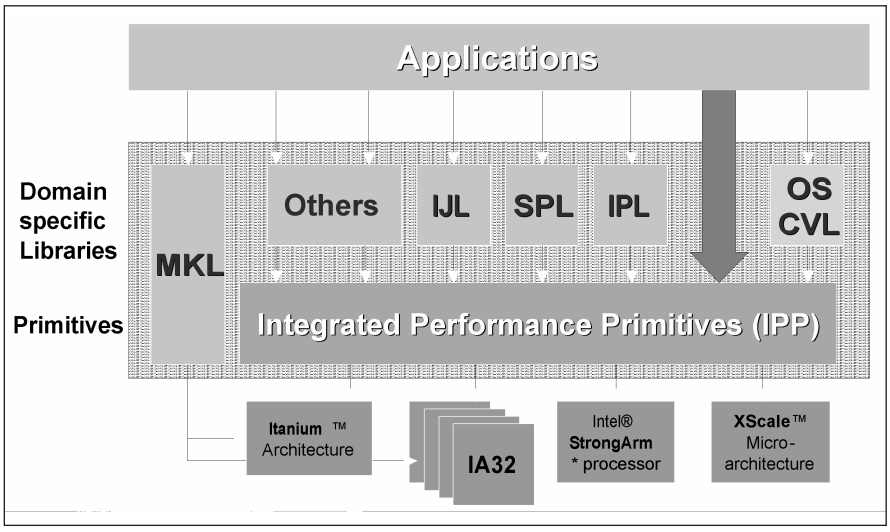

Figure 4. Integrated performance primitives (IPP).[6]

instruction set, to 1.5 GHz P4 (bar 4), a 50% increase in clock speed shows a jump from 8.03 frames/sec to 14.03 frames/sec, a 74% increase in performance.[4]

Compiling applications to take advantage of the new instruction set is necessary to reap the performance benefits of the new micro-architecture and instruction set. Figure 3 demonstrates: MPEG-4 compression software runs more than 300% faster after re-compilation.[5]

The IA-32 SIMD floating-point instructions fully support the IEEE* Standard 754 for Binary Floating-Point Arithmetic.[6] All SIMD instructions are accessible from all IA-32 execution modes: protected mode, real address mode, and virtual 8086 mode, insuring critical backward compatibility with legacy applications.

These instructions, extensions in the IA-32 architecture, allow all existing software to continue running correctly, without modification, on IA-32 microprocessors that incorporate the SIMD technologies. Existing software also runs correctly in the presence of new applications that incorporate these technologies.

Abstracting the instruction set from the physical layout of the chip allows the SIMD instructions to be deployed independently of micro-architectures. This allows the instructions to upwardly scale with higher processor frequencies and more advanced micro-architectures as they become available. Such micro-architectures might include the ability to perform multiple operations in parallel (superscalar) or execute instructions speculatively (ahead of the instruction pointer—little man computer becomes a prescient).

Optimizing Software Performance

To make media application authoring simpler, and performance gains more accessible, a library of optimized software routines called the integrated performance primitives (IPP) was created (Fig. 4).[7]

The system provides a cross-platform low-level software layer that abstracts multimedia functionality from the processor underneath. This allows transparent use of recent architecture enhancements from multiple processor types. The software can determine the type of processor it is running on and utilize the facilities available, for maximum performance on that processor.

IPP provides a highly optimized code for complex, compute-intensive media processes: more than 1,300 image processing functions and greater than 1,100 signal processing functions. These include arithmetic and logical operations, audio processing, geometric operations (scale, rotate, warp), color conversion, alpha composite, gamma correction, filters—FFT, DCT, wavelet transform, computer vision functions, image pyramid (resample and blur), Laplace, Sobel, Scharr, erode, dilate filters, motion gradient, flood fill, and canny edge detection.

Some indications of performance improvements from the integration of these libraries can be noted:

• Beatnik, an audio playback application, realized 22 times performance boost from utilizing MP3 Primitives for

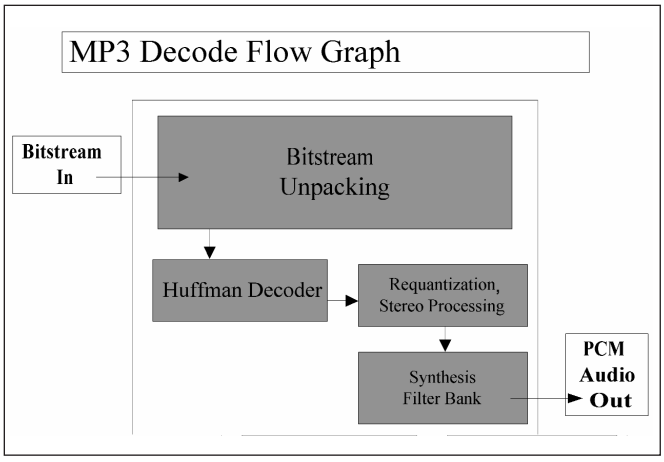

Figure 5. MP3 decode flow graph.

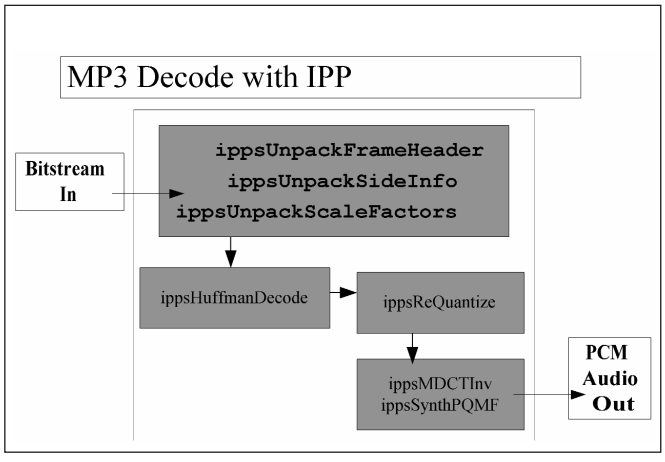

Figure 6. MP3 decode with integrated performance primitives.

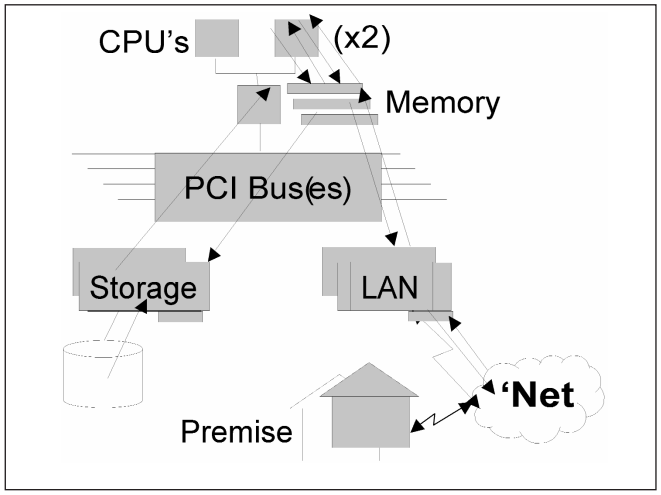

Figure 7. In current computer architecture, data requires multiple traversals of the PCI bus for each frame transfer.

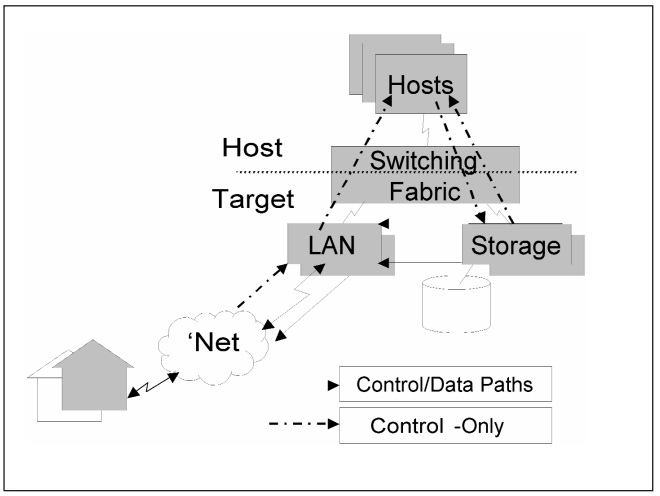

Figure 8. Example of a future server using Infiniband, with data moving from memory to the network, bypassing the PCI bus.

SA-1110, a Strong-Arm processor.

• Fonix, a speech recognition software program, boosted recognition accuracy from 93 to 99% via use of higher accuracy neural net.

• Solid Streaming, an MPEG-4 video decoder, boosted frame rate 1.8x by employing the IPP libraries.

To illustrate the benefits of the IPP library, consider the process of MP3 decoding, which starts with bitstream unpacking to parse the data for decoding. The Huffman decoder reverses the encode process and passes the data to the requantizer and channel separator. Finally, digital filters are applied to clean up any encoded noise (Fig. 5).

Coding this process in a high-level language from scratch is a 6000-line multi-man-month write, test, debug project.

Using the IPP library, the code required is reduced to 1,000 lines, mostly header preparation, and requires no assembly at all. The written code is not only smaller but easier to maintain and optimized for eight different types of processors (Fig. 6).

With the introduction of off-the-shelf computers and software in many video production facilities, users have experienced the need for RGB color space conversion to Y, Cr, Cb, for output to component video recorders. IPP libraries supply this much needed conversion in a hand-coded optimized software module. 4:2:2 to 4:2:0 and other conversions are also available.

Color Space Conversions using IPP:

RGB To / From…
YCbCr, YCbCr422

ippiRGBTOYCbCr[422]_[8u|16s|16u|32f][C3|AC4|P3]R
YUV, YUV422, YUV420

ippiRGBTOYUV[422|420]_[8u|16s|16u|32f][C3|AC4]R
YCC

ippiRGBTOYCC_[8u|16s|16u|32f][C3|AC4]R
HLS, CIELab, CIELUV

ippiRGBTOHLS_[8u|16s|16u|32f][C3|AC4]R
ippiRGBTOLUV_[8u|16s|16u|32f][C3|AC4]R
ippiRGBTOXYZ_[8u|16s|16u|32f][C3|AC4]R

Component Interconnection and System Design

Operators, software engineers, and accountants can recognize the benefits of Moore's law, which predicts a doubling of chip densities every 12 to 18 months, but input-output speed can severely restrict the functional performance of a system, decreasing overall efficiency. Systems engineers need a reliable, predictable plant with a measure of scalability. To create these useful media systems, high-speed connections must be made between devices. Linking powerful compute centers together with large storage disks and I/O devices is the function of the next generation interconnection fabric, Infiniband.[8] This interconnection system merges both storage area networks and system area networks into a higher speed fabric.

A fundamental I/O issue in current computer architecture is that data movement requires multiple traversals of the PCI and memory buses (Fig. 7) for each frame transfer. The arrows show the inefficiency of data transfers. The cycle is usually repeated for every frame. Even when buffered, the basic operations are the same, except for the repeated PCI bus traversals.

Figure 8 illustrates a future server if Infiniband or similar technology becomes the accepted standard. In this approach, at least where media is concerned, all "nodes" (CPU, NIC, storage) are intelligent. As a result of this, and the fact that a switched fabric inherently supports peer-to-peer data transfers (traversals of host memory are not required), it is possible to build a truly media-optimized solution. Using this approach, it's possible to construct a media server from low-cost building blocks that has no single points of failure and is scalable with simple operational characteristics.

Computers will still have an internal path, like PCI bus, for communication within the box, but I/O interfaces talk directly to the memory controller, bypassing the bus. The

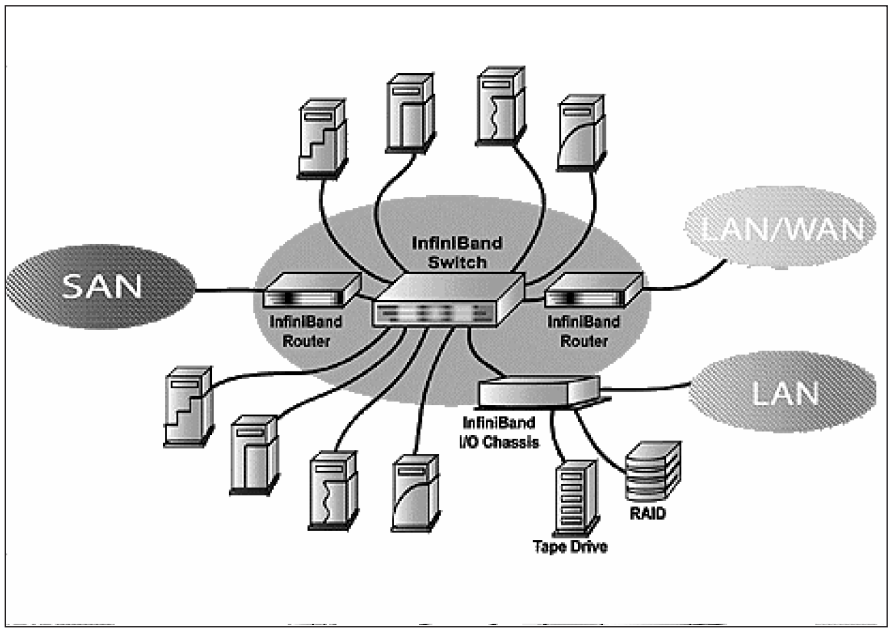

Figure 9. The Infiniband network.

end nodes in the network can be computers, disk storage, routers, or I/O devices such as video or audio A to D converters or internet servers.

The network uses a switched, point-to-point fabric employing virtual channels for communication at up to 30 Gbits per channel. There may be redundant paths through the fabric to promote fault tolerance, and multiple Infiniband subnets can be connected via its routers (Fig. 9).

Conclusion

Fixed function hardware designs currently limit the flexibility of media facilities, which may be better designed as scalable resources, managed on a demand or per-job basis. If more CPU power is needed to render a sequence or transcode between compressed formats, CPU power can be delegated to the task, independent of storage and I/O. If increased I/O capacity is warranted because a new service is being added, new I/O resources can be incorporated into the switched fabric. Storage is similarly scalable, like the storage area networks of today. Switched fabric like Infiniband is the routing switcher of the future, allowing greater control of facilities' day-to-day operations.

Twenty years ago, two computers were used to control a videotape machine to edit frame accurately. Ten years ago, computers were used to manage dedicated switchers, VTRs, special effects boxes, and audio boards. Five years ago, computers were making online edits with CPU rendered effects on hard disks and serving on-air signals. Today, it is possible to design the core of a media production and distribution facility from standard rack-mounted clustered computers, with appropriate off-the-shelf software, storage, and video I/O cards.

Entire moviemaking facilities are currently designed around thousands of commodity computers. In the future, media consumption will be pervasive on wireless PDAs and cell phones, from websites and broadband, on home PCs, high-definition displays, and even analog televisions.

By employing smart, fast technologies from the sub-micron to the system level, new media services can be designed on systems that are more customer-centric, easier to deploy and scale, evolutionary with software enhancements, and resolution independent. Advanced processors are a critical component of these systems.

Endnotes

1. Englander: *The Architecture of Computer Hardware and Systems Software,* 2nd Edition Chapter 6, Figure 06-05.
2. In software programs, state conditions can cause jumps in the program sequence. A deeper pipeline can increase speeds, and make decisions "out of sequence;" Decisions can branch to a limited degree before all factors are input. The execution trace cache is a tool for quickly backtracking through a decision tree with a conditional branch, and the decision path was based on an incorrect branch prediction. A combination of capable compilers and the execution trace cache can minimize the speed impact of mis-predictions.
3. Netburst contributions from Merlin D. Kister, Intel Software Solutions Group.
4. Data from tom's hardware guide, http://www4.tomshardware.com/cpu/00q4/001125/p4-07.html.
5. Data from tom's hardware guide, http://www4.tomshardware.com/cpu/00q4/001125/p4-06.html.
6. IEEE-754—Before 1984, when 754p became a "de facto" standard, (IEEE standard in 1985) arithmetic processing aberrations in commercial computers were programming "features" that were too important for programmers to ignore. IEEE 754 provides a rigorous open standard for floating point operations that achieves consistent results across many micro-architectures. Good background at http://http.cs.berkeley.edu/~wkahan/ieee754status/754story.html.
7. The Intel Integrated Performance Primitives API, Stewart Taylor, Microcomputer Software Lab, Intel Corp., Intel Developer, *Update Magazine*, Sept. 2001.
8. Infiniband contributions from Lester Memmot, Clayne Robison, Dale Taylor, Intel Corp. Software Solutions Group.

THE AUTHOR

Ralph Biesemeyer runs Intel's worldwide solutions marketing to the digital media and entertainment industries, developing Intel-based systems for digital production, media management, and digital distribution. Previously, he worked as director of production systems planning and engineering at NBC Network.

Biesemeyer has managed digital video products for Tektronix, Grass Valley Group, and Panasonic Broadcast USA. He is a member of SMPTE and has spoken at numerous industry conferences.