NSP Shows Promise on Pentium, PowerPC

Analysis Shows CPU Can Handle Some Signal-Processing Tasks

by Phil Lapsley, Berkeley Design Technologies

At Comdex last fall, Intel CEO Andy Grove outlined his company's strategy for bringing multimedia and communications applications to personal computers. A key component of this strategy is native signal processing (NSP): the execution of real-time digital signal processing (DSP) algorithms on the host processor, as opposed to using programmable or fixed-function DSP chips. Intel followed this by announcing an NSP reference-platform design kit (aimed at PC OEMs) at the Windows Hardware Engineering Conference in March.

These announcements led to consternation in both the PC and DSP communities. Many people did not believe it was possible to perform serious real-time signal processing on a general-purpose processor. Additionally, some said that even if you could do DSP on a Pentium, Microsoft Windows still would not be suitable for handling the real-time aspects of multimedia.

Our analysis indicates that it is possible to perform a reasonable amount of signal processing on a general-purpose processor such as Pentium, although the exact limits depend on the portion of CPU cycles devoted to NSP. However, processor performance is only one aspect of a successful NSP-based multimedia system; other critical components are memory and bus performance, a suitable software architecture for real-time performance, and the availability of applications that use the processor and support software to implement multimedia functions.

Intel's Stated NSP Goal: Raise the Baseline

According to Intel, the goal of its NSP initiative is to raise baseline system features that independent multimedia and communications software vendors can count on when building products. At present, PC multimedia applications are often tied to specific PC add-in hardware. For example, a voicemail application might work only with a specific telephone interface. Similarly, fax software requires the presence of a compatible fax modem.

These requirements complicate life for users, who must worry about hardware compatibility (or pay more money to purchase complete systems), and for developers, who must make (often incorrect) assumptions about what hardware resources are available in a multimedia PC. Intel hopes to improve the "least common denominator" system to provide standard multimedia functions. This change, Intel hopes, will simultaneously increase the market for multimedia applications and simplify the job of application developers.

Intel's NSP reference platform provides improved multimedia capabilities with minimal incremental cost. The reference platform adds a codec to the PC motherboard as a standard peripheral. The combination of the codec with NSP software and libraries allows application developers to build products that do not require the presence of specialized hardware, such as sound cards.

Other observers have attributed less civic-minded motives to Intel's NSP efforts, claiming that Intel is interested mainly in developing new markets for x86 MIPS. As one person put it, "If Intel made programmable DSP chips, you'd see them on the motherboard of every PC right now. But they don't, so what you see is the Pentium chip doing signal processing."

Although Intel certainly is not averse to building demand for more powerful x86 processors, the NSP strategy is not just a slap at DSP chip vendors. By improving PC multimedia capabilities at the lowest possible cost, Intel hopes to build the market for PCs, particularly the home market. Such a plan could benefit PC buyers while ultimately increasing Intel's revenue.

NSP Requires Hardware "Accelerators"

No one—not even Intel—believes that a Pentium processor can, by itself, perform all possible multimedia functions. Instead, add-in cards (containing either programmable or fixed-function DSP chips) will be required for more compute-intensive applications. For example, Intel envisions modems implemented under NSP using an add-in card with a dedicated data-pump chip and telephone data-access arrangement (DAA). In Intel parlance, the data-pump chip is a "signal-processing accelerator."

Obviously, this approach is a far cry from the "no extra hardware required" ideal that many people envisioned when they first heard of NSP. Even in this case, however, the NSP approach can reduce the number of components (and hence the cost) required for a modem card by having the host processor replace the microcontroller that would usually be required to drive the data pump. The use of a real-time operating system (IA-Spox, discussed below) allows the host processor to respond to the needs of the data pump in a timely fashion.

DSP Solutions Expensive, Fragmented

An alternative to NSP is to use a programmable DSP processor, either on the PC motherboard or on an add-in card. Not surprisingly, this approach is favored by a number of DSP vendors, including Analog Devices, AT&T, IBM, Motorola, and Texas Instruments. Apple, NeXT,

 © 1995 MicroDesign Resources

Processor	Max. Clock Rate	FP MAC*	Notes
Intel Pentium	120 MHz	3 cycles	
PowerPC 601	110 MHz	1 cycle	
MIPS R10000	200 MHz†	1 cycle	
Sun UltraSparc	167 MHz†	1 cycle	Can perform two 32-bit, four 16-bit, or eight 8-bit operations in parallel
HP PA-7100LC	100 MHz	1 cycle	Can perform two 16-bit integer operations in parallel

Table 1. General-purpose processor floating-point multiply-accumulate performance. *Best case. Includes time to fetch operands from cache and to update pointers if necessary. †Target performance.

and IBM are all examples of companies that have gone this route and placed DSP processors on the motherboard to support PC multimedia applications.

Unfortunately, this approach suffers from several drawbacks. First, DSPs are expensive: a typical DSP-based solution costs about $50. This cost covers the processor itself, external SRAM, analog I/O, and perhaps (in the case of a modem or fax card) a telephone DAA. NSP provides similar capabilities but lower performance and, except for the DAA and data-pump chip for a modem card, is essentially free. In low-end multimedia PCs, a $50 cost difference is often enough to tip the scales in favor of the cheaper system.

Second, the programmable DSP approach suffers from too many competitors. DSP vendors are hungry to get their chips onto PC motherboards, but all have incompatible instruction sets. As a result, neither users nor software developers can rely on a single DSP chip carrying the day. For example, should an application developer write a speech coder in TMS320C5x assembly language, DSP3210 assembly language, or IBM Mwave assembly language? In contrast, NSP leverages a standard hardware element—the host processor—to provide a common platform for developers and users.

Does this mean that DSPs are doomed for PC multimedia applications? Certainly not. As discussed above, signal-processing accelerators will be necessary for the more demanding PC multimedia applications. For high-end systems, add-in cards containing programmable DSP chips may well be a good choice. However, it now seems unlikely that programmable DSP chips will become ubiquitous in multimedia PCs.

New CPUs Boast Strong DSP Performance

Perhaps the biggest blow to the programmable DSP approach is that many newer general-purpose processors feature surprisingly good signal-processing performance.

A fundamental operation in many DSP algorithms involves multiplying numbers together and adding them to a running total. This so-called multiply-accumulate (MAC) operation is the heart of most digital filters and other operations based on a dot product. These computations are critical for modems, speech compression, and other multimedia functions. Indeed, single-cycle MAC support has traditionally been the main feature separating DSP processors from general-purpose processors.

Table 1 summarizes the best-case number of instruction cycles required to perform a MAC operation on data from cache for a number of general-purpose microprocessors, including Pentium and PowerPC. Pentium can perform a floating-point multiply-accumulate operation in three instruction cycles: one cycle to load one of the multiply coefficients from memory (cache), one cycle to multiply the contents of a floating-point register by the value of another multiplicand stored in memory, and a third cycle to accumulate the result in another floating-point register.

On a 100-MHz Pentium, this equates to a maximum performance of 33.3 million MACs per second—performance comparable to that of today's fastest programmable DSP processors.

The PowerPC 601, 603, and 604 do even better. Unlike Pentium, these processors offer a floating-point multiply-accumulate instruction that can execute in a single instruction cycle for single-precision data. Because the PowerPC MAC operates only on registers, two additional floating-point loads per MAC instruction are required to move multiplicand values from memory to registers. These PowerPC chips can all execute one of these loads in parallel with the MAC instruction, reducing the number of cycles to two. On certain block-based digital filters, it is possible to eliminate one of the memory accesses, resulting in a sustainable rate of one MAC per cycle.

Thus, assuming program and data are cached, a 100-MHz 601 can sustain 100 million MACs per second on some digital filters. This performance is twice that of the fastest commercially available DSPs and three times the performance of Pentium at the same clock rate.

Although MAC operations are an important part of digital signal processing, they're not the only part. Complete DSP applications place significant additional requirements on a processor—for example, delay lines, pointer updates, and control-and-decision code can often be found in DSP software. Table 2 shows the percentage of CPU time that a 100-MHz Pentium needs to execute several different multimedia tasks. These figures demonstrate that Pentium can perform reasonably well on a variety of multimedia tasks.

Clearly, however, the applications listed are at the low end of multimedia performance. For example, most MIDI synthesis hardware today supports 32 voices at 44.1 Ksamples/second. Interpolating from the data in Table 2, such an application would require 56% of a 100-MHz Pentium, emphasizing NSP's appropriateness for entry-level multimedia systems only.

 © 1995 MicroDesign Resources

NSP Saps CPU Performance

Still, these figures ignore an important question: What percentage of their CPU will users be willing to dedicate to signal-processing or multimedia tasks? The answer depends on what users want to do concurrently with these tasks. For example, an individual using an NSP modem to browse an online service may not want to run other applications at the same time. But the user who wants to receive a fax in the background while trying to do other work may be significantly less tolerant of stolen CPU cycles.

Another important issue is how the loss of CPU power to signal processing will manifest itself: Will it be a gradual, fine-grain reduction in processing, so that the system simply seems slightly slower, or will it be a jerky, halting degradation that is intolerable to users?

As multitasking NSP-based systems are not yet available, these questions are currently unanswerable. Perhaps the sole data point in this regard is provided by Apple, whose GeoPort-based Express Modem software implements a V.32bis modem (with V.42 and V.42bis compression and error correction) using native signal processing on Power Macintosh systems. According to Apple, this modem consumes approximately 30% of an 80-MHz 601 CPU. The product received generally poor reviews when first released last year, with users complaining of poor, erratic performance and excessive system loading. Since that time, however, Apple has introduced new versions of the Express Modem software, and users now seem somewhat happier.

Although Apple was the first PC vendor to implement NSP, the company seems to have no clear plan to make more extensive use of this strategy. In an ironic role reversal, Intel is leading the way in promoting a holistic hardware/software approach to multimedia, while Apple is using NSP in a few isolated cases.

IA-Spox Delivers Real-Time Response

For multiple applications to be supported simultaneously, raw performance is not enough for real-time signal processing; software must ensure that tasks meet their real-time deadlines. Intel plans to achieve this with IA-Spox, a version of Spectron Microsystems' Spox real-time operating system that has been ported to Pentium. IA-Spox runs as a ring-zero virtual device driver (VxD) under Microsoft Windows. IA-Spox installs interrupt service routines for interrupts from peripherals in which it is interested (e.g., the motherboard codec on the NSP reference platform).

Unlike Windows 3.1, which is not a preemptive OS, IA-Spox can preempt other tasks (including both IA-Spox tasks and Windows applications) when an interrupt occurs. This ability allows it to provide a real-time response to external events. Windows applications run in the "background," when no IA-Spox tasks need to run. (In some sense, then, Windows actually runs under IA-Spox instead of the other way around.)

Application	CPU Load (Percent of 100-MHz Pentium)
GSM speech coder/decoder (simultaneous encoding/decoding of monophonic speech at 8 Ksamples/second)	7%
Audio mixing (add together eight channels of stereo audio at 22 Ksamples/second)	9%
DigiTalk speech coder/decoder (simultaneous encoding/decoding of monophonic speech at 8 Ksamples/second)	17%
MIDI synthesis (wavetable-based synthesis of 8 voices at 22 Ksamples/second)	7%
ProShare video conferencing encoding/decoding/display (160 × 120 window at 10 frames/second)	22%

Table 2. Various NSP applications require a small but significant portion of a 100-MHz Pentium's processing power. (Source: Intel)

A key IA-Spox application is Intel's Native Audio Kit (NAK). This software interfaces to Windows applications via the standard WAVE API, providing audio recording and playback using codecs on the motherboard or on add-in cards. The NAK also provides audio mixing, sample-rate conversion, and synchronization between different audio streams. In addition, the NAK allows developers to create "mini-filters" that perform signal processing under IA-Spox on audio streams coming from or going to a codec; examples of such filters include speech or audio compression and filtering. The NAK also provides an interface for "mini-drivers" that allow hardware developers to quickly retarget the NAK to a new codec (such as might be found on a telephony board).

Intel and Spectron plan wide distribution of the IA-Spox run-time software, as it enables the NAK and other applications to perform in real time. As such, it is an essential part of Intel's NSP strategy. The cost to OEMs for licensing a run-time version of IA-Spox has not yet been disclosed. Spectron also will license IA-Spox developer's kits to firms interested in developing new IA-Spox applications.

Although IA-Spox provides important capabilities for NSP, like all real-time operating systems it does not guarantee that applications will meet their real-time constraints. In particular, IA-Spox is a priority-based real-time operating system, granting the CPU to the runnable real-time task with the highest priority. There are several methods for choosing priorities to guarantee that tasks meet their deadlines, but, at present, IA-Spox does not support any such method. As a result, IA-Spox applications are free to assign their own (possibly conflicting) priorities. Additionally, it is possible for users to overload the processor, resulting in missed real-time deadlines. Intel and Spectron are working to resolve these shortcomings.

An additional problem with Intel's proposed NSP software architecture is that it leaves a huge market out

 © 1995 MicroDesign Resources

in the cold: computer games. Intel's NSP vision relies heavily on Windows, both for its relatively clean software layering and its interoperability with IA-Spox. Unfortunately, most games today run under DOS and communicate directly with SoundBlaster-compatible add-in cards. Because these programs don't run under Windows, they cannot access IA-Spox. This will be less of a problem as PC games move to Windows, but for now NSP leaves an important class of applications unaddressed.

Finally, with IA-Spox, Spectron and Intel are stepping into PC system-software development, a territory that has been traditionally handled by Microsoft. Although Microsoft has been quiet about NSP to date, there have been rumors of a rift between Intel and Microsoft on the subject *(see **0906ED.PDF**)*. Given the large market that Intel and Spectron hope to address with IA-Spox, one must assume that Microsoft will be giving serious thought to adding competing real-time capabilities to future versions of Windows.

NSP Applications Currently Scarce

The missing link in NSP at the moment is applications: to date, no software vendors have announced NSP-based applications, although several are reported to be developing them. Given the tender age of Intel's NSP initiative, this is to be expected. In some sense, Intel is taking a *Field of Dreams* approach to NSP and independent software vendors (ISVs): "build it and they will come." That is, Intel feels that software vendors will naturally be interested in developing NSP applications, since Intel is trying to make NSP the "least common denominator" of multimedia-capable PCs.

There is good reason to think that ISVs will take advantage of this opportunity: developing applications for NSP-based systems should prove to be simpler and more economically attractive than developing applications for programmable or fixed-function DSP chips. Still, some ISVs are afraid that Intel may use its internal expertise to develop its own NSP applications (such as audio compression and decompression algorithms, ProShare video conferencing, etc.). Such ISVs note that it would be very difficult to compete with Intel's financial resources, marketing presence, and knowledge of NSP internals.

New Designs Could Improve Performance

Assuming that native signal processing is at all successful, it seems natural that general-purpose CPU vendors will move to enhance their processors' microarchitectures and instruction sets to make them more suitable for signal processing. Both HP *(see **081604.PDF**)* and Sun *(see **080103.PDF**)* have already made moves in this direction. Intel in particular could benefit from this strategy, as Pentium's signal-processing performance could stand improvement.

As discussed above, Pentium can perform one MAC every three clock cycles. Part of the problem is the separate multiply and add instructions required to implement a multiply-accumulate operation. Combining these into a single instruction would reduce the number of instructions required to two, a 33% performance improvement.

Further increasing performance would require fetching the data values from memory in parallel with the multiply and add instructions. Pentium has the capability to fetch two data values in a single cycle, but loads cannot be paired with floating-point math instructions. A significant redesign would be needed to allow such pairing, although no new instructions would be required.

Intel's P6 design *(see **090202.PDF**)* delivers a significant improvement over Pentium for NSP. Although the P6 does not feature instructions specifically aimed at signal processing, its 133-MHz target clock frequency, out-of-order execution engine, fast 256K secondary cache, and ability to execute up to five micro-operations per cycle all improve NSP performance. In particular, the P6's enhanced parallelism boosts throughput on the non-MAC code (e.g., loop counting) that is also a part of DSP algorithms. In addition, the P6 includes a new conditional move instruction that, for example, can implement saturating adds and subtracts in just two instructions.

Although the P6's floating-point multiply instruction is only half the speed of Pentium's, with a throughput of two cycles (as opposed to Pentium's one cycle) and a latency of five cycles (versus three), the P6 actually performs better on FP MAC–intensive code than Pentium. This is because a floating-point load (required for one of the two multiplicands) can be performed in parallel with a floating-point add operation, trimming the throughput of a MAC operation to two cycles.

This parallelism, in combination with the P6's faster clock rate (expected to be about 33% better than Pentium's in the same process), gives it a 2× performance boost over its predecessor on floating-point MAC operations. For applications that require even higher performance, the P6 offers a single-cycle 32-bit integer multiply with a four-cycle latency. While integer arithmetic is not as convenient as floating-point arithmetic for application developers, the single-cycle integer MAC may be appropriate for more demanding applications.

A problem with offering different versions of processors is product differentiation: users may wonder whether to buy an "NSP-enhanced Pentium" or instead wait for the P6. Widespread rumors indicate that Intel is indeed working on such enhancements for products that could debut next year. In addition, Cyrix reportedly is working on NSP enhancements for a second-generation M1.

Multimedia Applications Are Moving Targets

Just as processor speeds and their suitability for signal processing can be expected to increase in the future, so too can the computational requirements of multimedia

 © 1995 MicroDesign Resources

applications. For example, only a few years ago the 2400-bps V.22bis modem represented the state of the art. Such a modem took between 5 and 10 MIPS to implement on a typical programmable DSP. A more modern 14,400-bps V.32bis modem needs more like 20–25 MIPS on the same processor, and a state-of-the-art 28,800-bps V.34 modem requires roughly 40 DSP MIPS of processing power. Speech and audio coding and decoding algorithms show similar growth in their computational loads. Video compression and decompression are still in their infancy and can thus be expected to be on the steep part of the "MIPS growth curve."

Because of the growth in CPU power required for multimedia and communications applications, it is far from clear whether increasing processor speeds actually mean that the host CPU will be able to handle more signal-processing tasks in the future. Indeed, the next few years could see a "multimedia MIPS explosion" that swamps the capabilities of even the most powerful host processor. Whether this is a likely scenario is hard to say, but we observe that never in the history of computing have users complained of having too many MIPS.

NSP Will Do Well But Isn't a Panacea

As of this writing, Intel's efforts in NSP seem to meet two of three requirements for a successful PC multimedia system: good signal-processing system performance and a software architecture capable of supporting real-time signal processing. The third leg of the NSP platform—applications—is currently missing. This is not surprising, since NSP is relatively young and can be viewed as an enabling technology. We expect to see more applications for NSP as Intel's NSP reference platform proliferates.

For now, it seems that NSP will be most successful for low-end multimedia PCs. In these systems, it will provide basic multimedia capabilities at reduced cost and provide a common platform on which to build multimedia applications. The range of applications that NSP can address will grow over time as processor speeds increase and vendors tailor their processor designs and instruction sets for multimedia applications. Multimedia applications are a moving target, however, and the CPU cycles required for these applications will also increase. As a result, add-in cards containing fixed-function or programmable DSP processors will be with us for a long time to come. ♦

Phil Lapsley is vice president at Berkeley Design Technology (Fremont, Calif.), a consulting firm specializing in DSP. He is a coauthor of Buyer's Guide to DSP Processors *and* DSP Design Tools and Methodologies*, two industry reports on DSP technology.*

 © 1995 MicroDesign Resources