MICROPROCESSOR REPORT

THE INSIDERS' GUIDE TO MICROPROCESSOR HARDWARE

VOLUME 6 NUMBER 10 JULY 29, 1992

Windows NT Offers RISC a Chance on the Desktop

SDK Released to Crowd of Over 4,500 at Developers' Conference

By Brian Case

At the Win32 Professional Developers Conference in San Francisco early this month, Microsoft delivered pre-release copies of Windows NT to over 4500 attendees, setting in motion a frenzy of development activity without parallel in the history of computing. The attendees—application developers and members of the press—each received a Win32 Software Development Kit (SDK) CD-ROM containing two versions of NT: one for Intel 386/486 processors and one for MIPS R4000 processors. A third architecture—DEC's Alpha—will be supported by a version of NT currently under development at DEC and scheduled for release early next year along with the first end-user release of the x86 and MIPS versions.

The forthcoming release of Windows NT is bad news for OS/2: with NT so close to general availability, IBM will find it more difficult to convince users they need OS/2. OS/2 has significant support from IBM's large corporate users, but there seems to be far more interest from application developers in Windows NT.

The release of Windows NT could well mark the end of UNIX's chance for breaking into the high-volume desktop market. For years, UNIX has been far more capable than PC operating systems, but it has been hard to use and the standards have been fragmented. The UNIX camp is finally beginning to get its act together for the broader desktop market—with products such as Sun's Solaris and AT&T/Novell's Destiny nearing release—but it has taken too long, and the opportunity may have already been missed. Windows NT will give PC users the multitasking, networking, and other advanced features that UNIX offers, while at the same time tapping into the unstoppable momentum of Windows applications.

For the microprocessor market, Windows NT marks the end of Microsoft's dedication to Intel microprocessors for its operating systems. While the 486 and P5 will undoubtedly be the dominant platform for Windows NT, MIPS and Alpha systems will capture some piece of the market. Windows NT represents the best chance yet (aside from the PowerPC Macintosh) for RISC processors to enter the broader desktop computer market.

NT To Dominate Desktops?

NT is a threat to OS/2—and every other operating system for desktop and portable PCs— for several reasons. One reason is simply that it is available from Microsoft—an independent software supplier—instead of a computer company. Several versions of UNIX are also available from independent software suppliers, but that is the problem: UNIX is not sufficiently standardized. While Microsoft is not the most revered software company in the industry, developers and end users probably find it easier to commit to NT than to an OS offered by IBM, Apple, or another system competitor.

Another reason NT threatens to dominate desktops

Continued on page 6

In This Issue

 ISSN 0899-9341.

Windows NT

Continued from front page

is that it will be compatible with existing Windows 3 applications, tapping into the software base and market momentum of this environment. In fact, Microsoft is calling NT the most powerful member of its "scalable Windows product line." Windows "scalability" at the low end is limited, however; current versions of Windows are simply too slow to be used on low-end machines. (Recent versions of the Macintosh OS are also starting to overtax Apple's low end machines.)

Nonetheless, NT itself should be quite scalable since it can be used with a variety of processor architectures and can take advantage of multiple processors. Also, Microsoft seems to be providing a new degree of scalability through a plausible upgrade path that developers can use to make applications compatible with current versions of Windows and future versions of NT. (This upgrade path is discussed in more detail later.)

Technical advances are another reason NT may dominate: NT provides some sought-after workstation and network capabilities. NT finally does away with the segmented memory model in software, although Intel gave x86 processors a flat 32-bit address space years ago. This one feature can ease program development and improve execution speed tremendously. Among NT's other important features are pre-emptive multitasking, multithreading within a task, address space protection, and built-in user-access security.

The portability of UNIX has been demonstrated over the years; a decade ago, OS portability was so unique and compelling that it provoked yearly forecasts of explosion in the UNIX market from industry observers. Ultimately, the lack of standards kept UNIX from fulfilling its potential: the proliferation of incompatible versions prevented standard hardware and an application base from appearing. Sun made great strides in changing the situation, but the hardware has been simply too expensive.

With NT, the situation will be different: Microsoft is in complete control, so the fragmentation that crippled the UNIX market will not occur. While the Macintosh OS is also well standardized, it is not licensable. Apple's staunch refusal to license the Mac OS may have been a wise decision only in the short term; NT could destroy Apple's desktop market.

Windows NT eliminates one advantage PC operating systems have had over UNIX—more modest hardware requirements. Windows NT is every bit as RAM-hungry as UNIX, and it requires comparable processor capabilities and disk capacity.

Windows NT Overview

From the user's point of view, NT will look very much like Windows 3.x. The biggest change may be in how the OS is booted at power-up: unlike other versions of Windows, NT does not require MS-DOS to operate. Consequently, NT is truly an operating system instead of just a window system.

Software developers and users alike will appreciate NT's pre-emptive multitasking and address space protection—features standard in UNIX but rare in the PC arena. Currently, the standard in PC operating systems (Windows and Macintosh) is so-called "cooperative" multitasking in which the active application allows other applications to get processor time only when it explicitly tells the operating system it wants to relinquish the processor. A preemptive system, on the other hand, interrupts applications asynchronously and allocates processor time according to operating system rules. In some ways, preemption facilitates real-time tasks, since it is possible to interrupt a foreground task to service a real-time requirement for, say, video decompression. Getting it right can be tricky, however, and it remains to be seen how well NT accommodates real-time tasks.

Address space protection will help prevent buggy application software from crashing an entire system; only the offending application will crash.

In concert with preemption, NT provides transparent multiprocessing. When running on a multiprocessor system, the NT kernel will automatically distribute the processing load across the available processors. This will allow a user who has overloaded a single-processor PC to buy a multiprocessor system and realize immediate benefits (as long as the workload consists of multiple tasks or threads).

While some features of NT raise the standards of PC operating systems, other features are indications of just how far behind MS-DOS and Windows have been. For example, Microsoft is proud of the simplified mechanism for selecting a target printer. What they have implemented is basically the Macintosh Chooser, a feature Macintosh users have been enjoying for years.

Win16, Win32, And Win32s

The strategy used in NT to provide Windows compatibility centers around three APIs (Application Programming Interfaces): Win16, Win32, and Win32s. The current, 16-bit, Intel-only API is now called Win16, and the full-featured native NT API is Win32. NT will run application binaries compiled to the Win16 API in the Win16 subsystem. Current Win16 binaries will also run on RISC-based machines using emulation technology from Insignia Solutions.

A problem with Win32 is that applications compiled to this API will not run on the existing installed base of Windows 3.x machines. The reason is that the operating system calls in Win32 use 32-bit data and pointer val-

The Biggest Operating System Conference Ever?

On July 6, Microsoft opened its three-day Win32 Professional Developers Conference at the Moscone Center in San Francisco. By any measure, the conference was an unqualified success, with nearly 5000 attendees. The largest assembly hall at Moscone had to be supplemented with a healthy-sized overflow room (with a video link) for the keynote presentations.

What made the conference so appealing was the chance to get a preliminary look at Windows NT and to learn about it in detail in technical breakout sessions. All attendees received an impressive package of goodies: a spiral-bound conference schedule, three volumes of conference overheads, a folder containing an NT overview and the first two chapters of the forthcoming book "Inside Windows NT," a 67-page Win32 Development Tools Guide, a developer resource kit on CD-ROM, a Windows International Versions CD-ROM (Windows 3.1 and 3.0), the July release of the Win32 Preliminary Software Development Kit, a Win32 golf shirt, and a healthy shoulder bag to hold all this stuff.

Inside the SDK pouch is a hardware compatibility list (naming x86 and R4000 systems known to work with this release of NT), 59-pages of release notes, 3-1/2" and 5-1/4" floppies with installation startup software, and NT on CD-ROM. The CD-ROM includes Windows NT, full development environments, and 7500 pages of documentation in PostScript form. The development environment includes C/C++ compilers (though C++ is not yet available for MIPS), the Microsoft foundation class library (MFC, an object-oriented application framework to decrease development time for new applications), a linker, debuggers, header files, sample code, assemblers, editors, and on-line help. There are some holes; the new NT file system, compatibility with the full range of DOS and Windows applications, and POSIX compliance are all areas needing work.

On the first day, Bill Gates gave a keynote address and participated in an audience question/answer session. When asked his opinion of the current crop of Apple ads that claim making a PC windows-ready costs $3000, Gates said that the ads were not entirely factual. He then, however, went on to say that for $3000, a user could purchase a 486 box more powerful than anything Apple offers. This, of course, is also not entirely factual—the Macintosh Quadra is faster than any 486 system on many applications, though it is more expensive.

In the session following Gates, a general introduction to NT and its feature set was given by Paul Maritz, Senior VP of Microsoft's Systems Division. NT was shown running on a 386, a 486, and an R4000. The common demo was an application that rotated a bit-map picture of Gates' face. While not intended as performance benchmarks, the demonstrations did show a clear performance advantage for the R4000. An application was shown running on the R4000 under 286 emulation. The emulation software is based on Insignia Solutions' SoftPC, which Microsoft licensed.

One of the most impressive demos was of a database application running on an NCR 486-based multiprocessor (with four processors). The database was receiving a continuous stream of queries, and a processor-use graph showed the load being balanced across the processors with occasional spikes corresponding to complex queries.

The favorite demo, judging by audience response, was of the network software update capability. The demo showed the installation of a new version of operating system software remotely over the network. When the installation finished, which took just a few minutes, the remote machine was rebooted remotely. This prompted a smattering of applause in the audience, presumably from administrators of large networks.

The conference had numerous breakout sessions covering many aspects of NT, application porting, etc. Two demo rooms were also available: a large room full of NT systems for hands-on experimentation, sponsored by Microsoft, and another room with booths from software and system vendors. Witnesses said that NT crashed occasionally.

The hands-on demonstration room had about 250 systems running, of which about 30 were R4000 systems; the remainder were almost all 50-MHz 486 systems. Most of the R4000 systems had secondary cache. In a separate venue, DEC showed a prototype Alpha-based PC that can run VMS and will run NT when the port is complete. (Further details on NT systems will appear next issue.)

If you are thinking of ordering the SDK (see Price & Availability box) just to see what NT looks like, beware: the installation guide recommends 16 MB of memory with 12 MB minimum (16 MB minimum for MIPS systems), and it also informs that 386 processors made from the B0 or B1 mask sets will not run NT. Also, a CD-ROM drive and compatible display and SCSI hardware are needed.

ues, while the current version of Windows is limited to 16-bit values. Since the underlying 386/486 hardware is capable of handling 32-bit values, a bridge mechanism that converts 32-bit calls to 16-bit calls will solve the problem.

Microsoft has defined a subset of the full Win32 API, which it calls Win32s (with the "s" suffix indicating that it is a subset). An application binary compiled to the Win32s API will be able to run on both NT and Windows 3.1. The bridge mechanism used on Windows 3.1 consists of two parts: a set of function-call stubs in a Dynamic Link Library that convert, at runtime, Win32s function calls to Win16 function calls (and the reverse for return values), and a virtual device driver that supports the 32-bit flat memory model and improved exception handling support.

While not all the capabilities of Win32 are supported in Win32s, this bridge allows developers to ac-

Price & Availability

The Win32 Preliminary SDK is available by mail for $399 including printed documentation or $69 for the software only (with documentation on the CD-ROM). This price includes updates through the final 1.0 version. The SDK includes not only the Windows NT operating system, but also a macro assembler, editor, debugger, linker, and C/C++ compiler, with on-line references and tutorials—quite a deal for $69! To order, call 800/227-4679 in the U.S., 800/563-9048 in Canada; elsewhere, contact your local Microsoft office.

cess NT and Windows 3.1 with one binary version of an application and take advantage of a flat address-space model on the PC. Microsoft says that the function-call stubs slow execution by about 8%, but that taking advantage of 32-bit addressing and data movement more than compensates.

In addition to the three Windows APIs, Microsoft plans to support at least two others. As shown in Figure 1, NT will eventually run applications written to DOS, Win16, Win32s, Win32, OS/2, and POSIX API standards. POSIX compatibility was not included in this release.

Beyond NT: Cairo

In the past, Microsoft has been little more than a follower in the area of operating system and interface design. MS-DOS was a direct rip-off of an existing system, CP/M, and Windows followed the Macintosh by years with clearly inferior capabilities. Based on early descriptions of their next-generation operating system, code-named Cairo, Microsoft may be poised to become a leader instead of a follower.

Only a few details about Cairo were given. First, it will have a new, database-like user-interface paradigm. The Cairo motto is "information at your fingertips," which refers to its ability to deal with large amounts of data. A user will find information by browsing the system, refining the specification of the desired information until it is located. Information will be easily retrievable even on large networks.

Cairo will be fully object-oriented both in its programming model and user-interface. It has been designed to gracefully handle very large numbers of objects (millions) regardless of size (single bytes to gigabytes) and location (50K network nodes).

Of course, Cairo is vaporware and Microsoft is eager to make it sound as good as possible. Therefore, it is foolish to predict the impact of Cairo, but if Microsoft can meet its promised time frame—beta in 1993, end-user ship in 1994—it could seriously limit the prospects

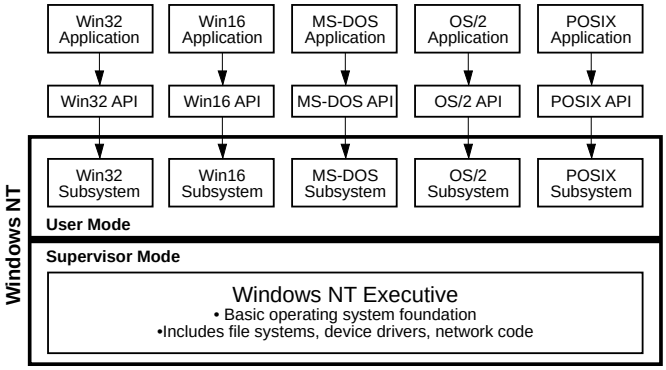

Figure 1. Windows NT system organization.

of other next-generation operating systems, such as Pink from Taligent. Just as NT is bolstered by the momentum of Windows 3.x, Cairo will ride on the coattails of NT. In any case, the promise of Cairo gives developers yet another reason to focus on Microsoft's Windows strategy and forget alternatives like Macintosh, NeXTStep, and UNIX.

Microprocessor Market Implications

Windows NT stands an excellent chance of being a huge market success simply because of its Windows 3.x compatibility and the commitment of existing Windows software developers. If it lives up to its potential, NT could have significant implications for the microprocessor market.

Perhaps the least questionable outcome of NT's success is an increase in the size of the market for high-end microprocessors and DRAM. Since NT will require more power than most of today's PCs and offers appealing new features, it will lure many users into upgrading their systems sooner than they otherwise would have.

Because of the linear address space, multitasking, and support for multiprocessing, systems running NT could seriously cut into the existing workstation market. If engineering software developers port their applications to NT (as Mentor and Viewlogic have already announced plans to do), the competition in the PC hardware business could lead to a significant migration of UNIX workstation users to Windows NT machines.

The most interesting implication of NT is the opportunity it provides to those system makers who have a vested interest in RISC processors. So far, RISC processors have simply been locked out of the largest market with no hope of ever penetrating. Now, Microsoft is providing an opportunity that RISC system vendors could only dream about in their wildest fantasies.

The MIPS and Alpha architectures are the first RISCs to take advantage of Windows NT's portability, but they won't necessarily be the only ones. In fact, li-

Continued on page 13

accumulator-bound architectures.

The ST9 family is sole-sourced, which was once a rarity among 8-bit microcontrollers but is now more common. SGS-Thomson and Siemens have a technology exchange agreement, under which Siemens received flash memory technology and rights to the ST9, while SGS-Thomson received rights to the 80C166 family of 16-bit microcontrollers. After serious consideration, Siemens decided not to second-source the ST9 family because of its overlap with Siemens' 8051-architecture products at the low end and its 80C166 at the high end.

Table 3 compares price and features for a number of high-end 8-bit microcontrollers and two of the recent enhanced versions of the 8051 architecture from Siemens.

The ST9 offers a peripheral set very similar to Motorola's full-featured 68HC11, but with the enhancement of an on-chip DMA controller. It also includes some memory configurations with a small amount of on-chip EEPROM—a standard feature of U.S. high-end 8-bit microcontrollers such as the 68HC11 and TI's TMS370, but rarely available in Japanese families. The ROM sizes range up to 16K, matching TI's largest parts, but smaller than Motorola (24K), Hitachi (32K), or NEC (32K).

The Siemens parts are not directly comparable with the other devices listed in the table, because their largest ROM size (16K) is not available with an on-chip A/D converter. All of the other devices in the table, including Siemens' 8K ROM part, have on-chip A/D. The Siemens parts are not available with OTP ROM. Siemens plans to fill the need for user-programmable devices by introducing parts with flash memory (expected in late 1993).

The ST9 is very competitive on speed. Although not quite as fast on a per-instruction basis as NEC's K2 family and Hitachi's H8, its superior addressing modes should narrow the performance gap. NEC's family is an accumulator-bound Z80 derivative, and Hitachi's family does not allow memory operands for arithmetic and logical instructions. The ST9 allows memory operands and lacks the accumulator bottleneck. Both SGS-Thomson and the Japanese far outpace most U.S. microcontrollers, such as Motorola's 68HC11 and TI's TMS370 families, in terms of performance.

With regard to pricing, the ST9 is competitive with both Japanese and U.S. vendors. Japanese microcontrollers tend to be cheaper than U.S. microcontrollers, but this mostly reflects a difference in pricing for parts with on-chip EEPROM. There are few Japanese parts with EEPROM, while there are few high-end, 8-bit U.S. parts without it (especially in the larger ROM sizes). The ST9 approaches aggressive Japanese pricing in EEPROM-less microcontrollers, and comes close to U.S. pricing in microcontrollers with EEPROM.♦

Windows NT

Continued from page 8

censing Windows NT and completing the port as soon as possible may be the best chance companies such as HP have of broadening their workstation markets.

Of all the workstation vendors, Sun seems to be in the most difficult position with respect to NT. Lately, Sun has not kept up with SGI, HP, and DEC in performance and has been thriving largely on a superior application base. If that application base migrates to NT, Sun may be forced to provide NT on its workstations. Embracing NT, however, would ruin the prospects of the Solaris OS on the 486 and seriously weaken the SunSoft subsidiary. If Sun does *not* do a port of NT, the company that so loudly touted the advantages of open systems and standards may find itself in the embarrassing position of being the one of the only significant RISC system vendors not offering Windows NT on its systems.

In the final analysis, the RISC vendors are at the mercy of application developers. If developers decide not to make versions for RISC processors available, no RISC systems will be sold. Part of the promise of Windows NT is that the consistent OS and development environment for a range of processor architectures will make porting easy enough that developers of Intel-based NT applications will provide RISC versions, even though the RISC architectures won't initially have a large enough installed base to justify much development effort. This could dramatically level the playing field, allowing several processor architectures to participate in the general-purpose PC market, each seeking a particular niche.

In the short to medium term, Intel clearly has the best position: whatever happens, the demand for performance is likely to increase, and x86 processors will be able to run whatever succeeds: Windows 3.x, Windows NT, or OS/2 (or NeXTStep, Solaris, or Taligent's Pink, for that matter). In the near term, most NT machines are likely to use 486 or P5 processors, not MIPS, Alpha, or any other RISC. Ironically, while NT is giving RISC-based computers their biggest chance to broaden their market, it will also give a big boost to the x86 family by offering PC users a vastly improved operating system and pulling some workstation business into the x86 fold.

Despite Intel's advantage, NT is giving RISC a chance. Now it is up to the systems vendors to make good on RISC's promises. If NT and Cairo fulfill the promises Microsoft has made, RISC vendors may be able to take market share away from Intel by exploiting portable, standardized operating systems and the advantages offered by RISC architectures—if they are able to deliver on the promise of better price/performance at high-volume price points.♦