PRODUCT FOCUS

Project A

10 Project Managers

Steve Jobs' new *"machine for the '90s"*

The NeXT Computer

- **25-MHz 68030** • **Optical Drive**
- **Math and Digital Signal Processors**
- **8 Megabytes of RAM**
- **Windowing Unix**

IN DEPTH
Parallel Processing

PLUS
Dick Pountain on:
Linn's Innovative Rekursiv Chip
Parallelizing Prolog

PC Backup Power Supplies
5 Short Takes

The NeXT Computer

With an optical drive,
a 25-MHz 68030,
built-in floating-point,
digital signal processing,
8 megabytes of RAM,
Unix, and more,
this is a power user's
dream machine—
but will you be able
to buy one?

Editor's note: In August, Nick Baran, Tom Thompson, and I attended a marathon, all-day briefing at NeXT's headquarters in Palo Alto. It was the first time a publication was given an in-depth look at what surely is one of the most eagerly anticipated machines in recent memory: the NeXT Computer.

On this and several follow-up visits we saw beta versions of the hardware, system software, and some early applications. We met with many of the engineers and programmers who developed the machine's hardware and software, and we spoke with the managers who are determining where NeXT is going and what role it will play in the microcomputing community.

We weren't disappointed. This is a milestone machine—one that in all likelihood will cop machine-of-the-year honors all around.

BYTE will have ongoing coverage of the NeXT Computer in upcoming issues. We'll report definitive performance figures, for example, after we receive and test a production unit. Here are our first impressions of the beta hardware and software.—FSL

I t's been a long wait, but it has finally arrived. In early October, Steve Jobs's NeXT, Inc. unveiled the fruit of its creative efforts: a workstation referred to as "the cube."

NeXT asserts that the cube, having been designed to meet the computing needs of the next decade, is "the machine for the nineties." A bold statement, to be sure, but the cube goes a long way to bolster that claim: It sports the first commercially available erasable optical drive and advanced VLSI (very-large-scale integration) technology, and it comes with a built-in digital signal processor. On the software side, the Unix-based cube features an object-oriented version of C as its standard programming environment. It uses Display PostScript to present a graphical user interface that shields users from the traditionally user-hostile Unix command syntax, and it offers easy access to the cube's considerable power.

Targeted initially for the higher-education market, NeXT built the cube with the feedback of an academic advisory council that consisted of researchers and professors from schools such as Carnegie-Mellon, Stanford, and the University of Michigan.

The academic bent shows throughout. For example, the digital signal processor can be programmed for real-time labora-

tory work and demonstrations. The cube's large mass storage and memory capacity make it ideal for accessing substantial libraries of information. And Unix is the multitasking operating system of choice in academia.

Although the cube delivers a lot of bang for the buck, it's priced in the neighborhood of $6500 (all prices quoted are aimed at the higher-education market), which may, at least initially, limit its availability to its intended user base: students. The cube's rich features list would surely be appealing to those in nonacademic settings (engineering and science applications come to mind), but we were surprised to learn that for now, NeXT has no firm plans to pursue these markets.

Outward Appearances

The cube is starkly simple in appearance and physical layout. The main computer unit is a matte-black cube measuring 1 foot to a side. There are no switches, and no indicator lights. There are two panels covering bays that can hold two 5¼-inch full-height devices. One bay is occupied by a full-height drive with a wide slot: a magneto-optical drive. The main system unit is a power user's dream: the latest generation Motorola 68030 processor and 68882 math coprocessor, plus 8
continued

megabytes of RAM as *standard* hardware (a 4-megabyte version of the system is available). An army of connectors (such as a SCSI [small computer system interface] connector and "thin" Ethernet connector) located along the rear of the computer can hook the cube to nearly any peripheral device (see photo 1).

The system is designed to avoid the rat's nest of wiring all too common with complex systems. The entire cube system requires just one power cable, which connects the main unit to a wall socket.

A single 10-foot-long shielded umbilical connects the black 17-inch monochrome monitor to the main unit (see photo 2). This cable carries power for the monitor, video, keyboard, mouse, sound I/O, and auxiliary input signals in a complex shielded array. The black keyboard attaches via a connector to the base of the monitor, whose housing also contains a small speaker, stereo earphone jack, two stereo channel jacks, and a microphone jack. A two-button mouse (also black) connects to the keyboard (see photo 3). The beta cubes we looked at were FCC Class A certified.

This arrangement is very convenient: Your desk need only accommodate the monitor, keyboard, and mouse, and the ample length of the umbilical gives you the freedom to place the main unit well away—say, on a shelf. A key on the keyboard switches the system's power on or off so you don't have to touch the main unit at all.

Fine-Tuned for High Throughput

The cube's internal construction mirrors the simplicity of its exterior (see photo 4). The main unit's cubic housing is made of lightweight magnesium. Inside are four 32-bit NuBus slots, one of which holds the system's main CPU board. All the cube's system electronics reside on this densely packed CPU board, which makes heavy use of surface-mount devices; the cube is essentially a single-board computer. With the exception of a bipolar array used to manage the video display and perform Manchester encoding/decoding for Ethernet communications, all the CPU board's parts use low-power CMOS components.

A power supply mounts inside the housing on two screws; the entire box is cooled by a large, quiet, low-speed fan. The nonswitching power supply can handle voltages ranging anywhere from 90 volts to 260 V, and frequencies from 50 Hz to 60 Hz. This means that you can plug in the same hardware almost anywhere in the world without having to set switches. The cube should also prove

resistant to the vagaries of commercial electrical power. Its power supply generates 200 watts, of which the monitor uses 50 W, and 25 W is allocated for each slot.

NeXT's design for a workstation for the nineties used four important strategies. First, when possible, high-performance components were used. The CPU board is built around the 68030 processor and 68882 floating-point unit, both running at 25 MHz. For SCSI peripherals, the NCR 53C90 SCSI interface chip provides a maximum 4-megabyte-per-second transfer rate. That's considerably

Photo 1: *The cube's I/O ports. Top to bottom: DSP port, two serial ports, SCSI port, laser printer port, Ethernet port, and monitor port.*

Photo 2: *The monitor I/O ports, left to right: stereo earphone jack, left and right stereo channels, umbilical connection, keyboard connector, microphone jack.*

faster than the 1.5-megabyte-per-second rate of the older NCR 5380 chip. For mass storage, an optional high-speed hard disk drive using the SCSI bus is available. This hard disk holds 670 megabytes of formatted data and has an average seek time of 18 milliseconds.
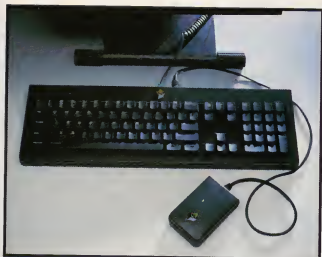
However, even a high-performance processor can be slowed to a crawl if it must service every I/O call, or wait on slow peripherals. (Steve Jobs put it this way: "MIPS is only one-third of the equation; sustained system throughput is the key.") So, the second part of NeXT's design strategy was to minimize the

Photo 3: *The cube's keyboard. The keys above the cursor keys control the monitor's brightness, the system's power, and sound volume.*

Photo 4: *The inside of the cube. At top center is a bay for an additional full-height peripheral; at center is the magneto-optical drive. The power supply is at the bottom. To the right of center is the main CPU board.*

Photo 5: *The NeXT 17-inch monitor. The screen can be tilted forward or back; the tractor-style wheels allow it to be rolled across a table.*

overhead of communicating to the outside world by offloading as much I/O from the CPU as possible onto smart I/O processors managing each peripheral (see figure 1). This happens to be a matter of necessity given the amount of I/O the cube is doing. Consider that the cube's synthesized digital sound is handled by a Motorola DSP56001, a 20-MHz digital signal processing (DSP) chip. The DSP56001 provides the cube with its ability to synthesize compact-disk-quality stereo sound—no mean feat when you consider it must handle two channels of 16-bit data sampled at 44.1 kHz. Although the primary function of the DSP is to minimize system overhead while processing high-quality sound, you can program the DSP56001 to manipulate any sort of digital data, say, signal filtering or image processing (see the text box "The Cube's Digital Signal Processor" on page 166). The DSP makes the cube an excellent machine for laboratory and experimental work.

That's only part of the I/O traffic. Looking at the back of the cube, we counted no less than seven I/O ports. These include the following:

• *A DB-19 monitor port* carries all video signals, video data, control signals, mouse movement, stereo sound, and 12-V DC power to the NeXT monitor. Both the sound I/O data and video data (1 pixel every 10 microseconds) are managed by dedicated DMA (direct memory access) channels.

• *A "thin" coaxial Ethernet port* operates at 10 megabits per second and is driven by an AM7996 Ethernet transceiver chip.

• *A DB-9 serial printer port* drives the NeXT laser printer (see the text box "The NeXT Laser Printer" on page 168). This port transfers data at 1.8 mbps when printing at 300 dots per inch, and 3.2 mbps when printing at 400 dpi.

• *A DB-25 SCSI port.* Its signals are identical to those of the Apple Macintosh SCSI port. As mentioned earlier, the SCSI bus can transfer data to a peripheral at up to 4 megabytes per second.

• *Two serial ports* that use the Macintosh mini DIN-8 serial connectors and signals. Both serial ports can handle up to 230.4K bits per second synchronously (the same as Apple's LocalTalk), and 38.4K bps asynchronously.

• *A DB-15 DSP port* connects to both the asynchronous (SCI) and synchronous serial (SSI) channels on Port C of the digital signal processing chip. This port can be used to receive or output digital data.

continued

NeXT COMPUTER
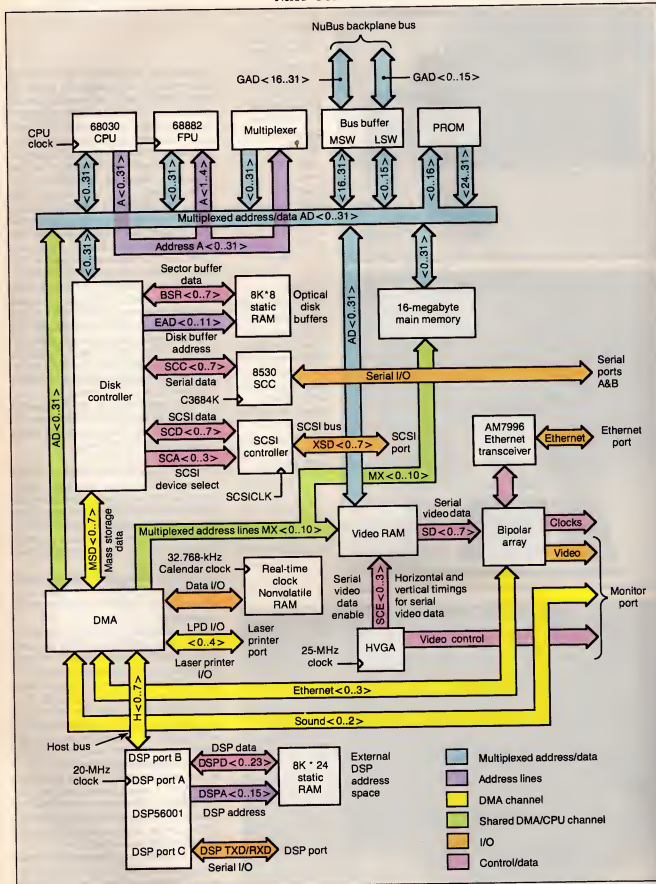
Figure 1: Block diagram of the NeXT system. Note that the PROM is read by driving 16 bits of address onto the bus and reading bytes off the most significant address lines.

The Only Character Recognition System That Outperforms Ours.

Consider the facts:

Nature's character recognition system can be trained to read all characters and languages it sees. **SPOT** (Flagstaff Engineering's Optical Character Recognition program) is trainable too. It reads most printed documents in English and more than 130 other languages.

Like nature's original, **SPOT** is very discerning. It can glance over an entire page or zoom in on a few lines of text. **SPOT** can read newspapers, magazines, books, manuals, invoices, contracts, government documents and much, much more.

Your eyes are teamed up with that marvelous computer, the brain. **SPOT** uses the latest man-made personal computers in conjunction with the most popular low-cost scanners, and requires no additional hardware.

SPOT's advanced logic closely approximates what the human eyes and brain do. That's why **SPOT** will continue to be *the leader in OCR* ... with an intuition inspired by the original.

Seeing is believing. Since 1982, Flagstaff Engineering has helped thousands of international customers read information from various computer and printed media. Demonstration programs are available. Call today.

FLAGSTAFF ENGINEERING

Helping People Read a World of Information
1120 Kaibab Lane • Flagstaff, AZ 86001
602-779-3341 • FAX 602-779-5998

S.P.O.T.
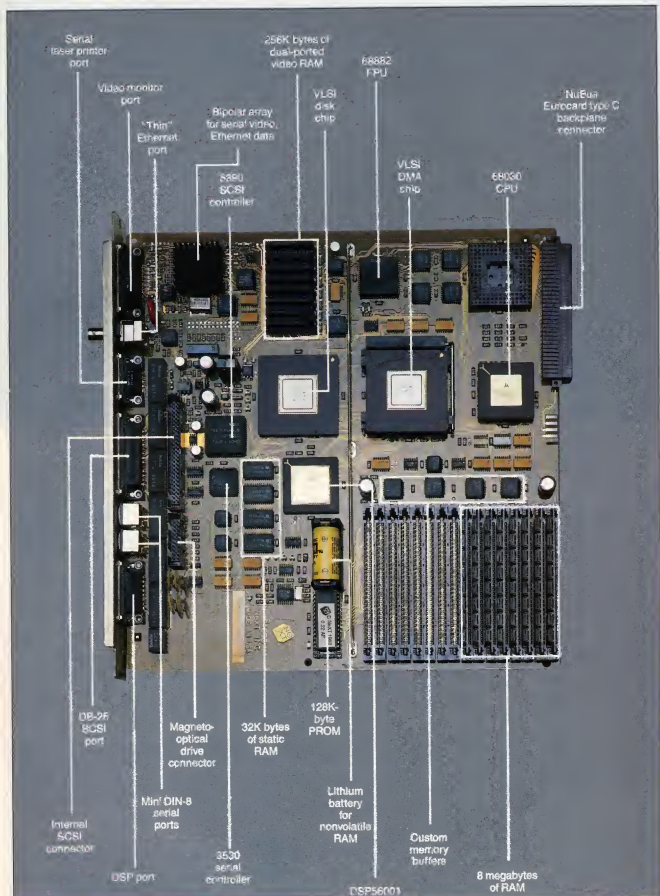SYNTACTIC PATTERN OPTICAL TRANSLATOR

Circle 130 on Reader Service Card
(DEALERS: 131)

Serial
laser printer
port

Video monitor
port

"Thin"
Ethernet port

8390
SCSI
controller

Bipolar array
for serial video,
Ethernet data

VLSI disk
chip

256K bytes of
dual-ported
video RAM

68882
FPU

VLSI
DMA
chip

68030
CPU

NuBus
Eurocard type C
backplane
connector

DB-25
SCSI
port

Internal
SCSI
connector

Magneto-
optical
drive
connector

Mini DIN-8
serial
ports

DSP port

32K bytes
of static
RAM

8530
serial
controller

128K-
byte
PROM

Lithium
battery
for nonvolatile
RAM

DSP56001

Custom
memory
buffers

8 megabytes
of RAM

Figure 2: *The NeXT Computer's main CPU board, with its two custom VLSI chips.*

Looking inside the case, the main CPU board has two more ports: a 20-pin connector for the optical disk drive, and a 50-pin SCSI connector for a hard disk drive. Finally, inside the cube's housing are four 32-bit NuBus slots. Each slot uses a Eurocard type C connector. NeXT has implemented a CMOS NuBus with twice the data rate of the standard NuBus for its backplane bus. The CPU board assumes the ID of the slot it occupies. Although they're not used for outside communications, each of these devices can make demands on the system.

For digital sound synthesis, there happened to be an off-the-shelf component—the DSP56001—that could be assigned the job. Unfortunately, there aren't high-speed processors available that could deal with the rest of the system's I/O, and certainly none that could handle the magneto-optical drive. Two custom VLSI chips were designed to manage the cube's remaining I/O subsystems. These chips handle the SCSI interface, the magneto-optical drive (including error-correction logic), the serial ports, and Ethernet transfers.

Both these chips pack a lot of components: According to NeXT, each chip contains about *10 times* the amount of logic circuitry used by an entire Mac II.

But there's still a problem lurking here, subtly related to I/O: how to manage data to and from these I/O processors. If the CPU must periodically transfer data between memory and the various I/O processors, the system's performance is still degraded.

NeXT's third design strategy was to improve data throughput within the system itself by managing these transfers with custom DMA hardware. This DMA hardware is implemented in one of the same VLSI chips that helps manage the system I/O. There are no less than 12 DMA channels on the main CPU board. They include the following:

• two Ethernet channels (one for transmitted data, one for received data),
• one video channel,
• one serial channel (for both serial ports),
• one DSP channel,
• two disk channels (one for the magneto-optical drive, one for a SCSI hard disk drive),
• one printer channel,
• one memory-to-DMA register channel,
• one DMA register-to-memory channel, and
• two sound channels (one for input, one for output).

For the memory-to-register and register-to-memory DMA channels, "register" corresponds to a 16-byte register buffer in the DMA hardware. The contents of these registers can be copied repeatedly under DMA control to memory. An example of this would be to copy a background pattern for the video display into the DMA registers, and then use the register-to-memory DMA channel to copy the pattern into all of the video memory.

The final aspect of NeXT's overall design strategy to improve throughput is that when the 68030 processor must access memory, it attempts to do it efficiently. The 68030's *burst read cycle* is used where possible, since this mode allows four long words (128 bits) to be transferred in 9 clock cycles, instead of 16 clock cycles—roughly twice as fast.

Memory and Mass Storage
One way to improve system performance is to keep as much of the executable code in memory as possible, particularly where multitasking is concerned. The cube has no problem in this area: It comes equipped with 8 megabytes of 100-nanosecond SIMM-mounted RAM (see figure 2). The main CPU board has 16 SIMM (single in-line memory module) sockets, and 8 of these are populated with the standard 8M.

You can add additional 1-megabit-density SIMMs in 4-megabyte increments to expand system RAM to either 12 megabytes or the maximum of 16 megabytes.

Also located on the main CPU board are 32K bytes of 45-ns static RAM. 8K bytes of this SRAM are used for the magneto-optical disk buffers, and 24K bytes are allocated for the DSP56001. There are also 256K bytes of dual-ported video RAM for the video display. A 128K-byte PROM contains the bootstrap and some

> *The optical cartridges resemble overgrown 3½-inch floppy disks, but hold a whopping 256 megabytes of data.*

diagnostic code for the cube. This bootstrap code simply loads the Unix kernel and starts it. There are no special graphic or system functions similar to the Macintosh Toolbox embedded in this ROM. The operating system, drivers, and custom display software reside on the boot drive.

The most interesting peripheral on the cube is its read/write magneto-optical drive. The optical drive fits into a 5¼-inch full-height bay on the cube and has a slot to accept an optical cartridge. The cartridge is removable through a software-actuated eject mechanism using an internal motor.

The optical cartridges themselves resemble overgrown 3½-inch floppy disks, complete with a rigid shell and shutter door, but the resemblance ends there; each optical cartridge holds a whopping 256 megabytes of user data. This allows you "to take your entire world with you" since the Unix kernel, the bundled applications software, and lots of user data will fit on a single cartridge.

The optical platter is composed of the same clear rigid polycarbonate material that's used in CD-ROMs. Embedded within the platter is a layer of reflective aluminum backing that's overlaid with a magneto-optical substrate. The platter rotates inside the cartridge at 3000 revolutions per minute, 10 times the rotation speed of a CD-ROM, and almost as fast as a hard disk drive.

How does the magneto-optical drive work? A single laser performs both read and write operations. To write data to the disk, the drive first applies a magnetic field to the platter. The orientation of the magnetic field determines the data to be written to the platter—either a 0 or a 1. The magnetic field is first oriented to write 0s at the start of what's called the *erase pass*.

The laser uses a high-power beam to heat a sector on the platter's substrate to its *Curie point*—the temperature at which the crystals in the substrate "forget" their previous orientation and reorient themselves to the surrounding magnetic field. All the data in the target sector is thus erased to 0s.

Next, the magnetic field is oriented to write 1s in the *write pass*, and at every spot in the sector where a bit must be set to a 1, the laser again heats the substrate to the Curie point. Finally, the sector is read in a *verify pass* to check the accuracy of the data.

To read data off the platter, the drive removes the magnetic field, and the laser directs a low-intensity beam at the plat-
continued

The Cube's Digital Signal Processor

The Cube comes equipped with a Motorola DSP56001, an 88-pin CMOS chip designed for data-intensive real-time signal processing applications. At the core of the chip are three execution units—data arithmetic logic unit (ALU), address-generation unit, and program-control unit—that operate in parallel to provide the necessary throughput.

The DSP works with 24-bit digital data, providing 144 decibels of dynamic range. Two internal 56-bit accumulators provide 336 dB of dynamic range during arithmetic operations so the precision of the intermediate results is retained during data processing.

The DSP56001 is programmable, allowing it to be tailored for a specific purpose. The 16-bit address-generation unit, combined with hardware select lines for program code or data, can access three separate 64K words of an external memory space (192K words total, where a word is 24 bits of data).

The DSP56001 has on-chip program memory composed of 512- by 24-bit-wide RAM cells, of which the bottom 64 cells are used for interrupt vectors. DSP programs can occupy the remaining memory, or if they're large, they can reside in the external program space. In the latter case, the on-chip program memory can serve as a fixed cache. Program instructions are 24 bits wide, and each bit is significant.

On the cube, the DSP56001 is clocked at 20 MHz, and instructions execute every two clock cycles to give the chip a 10-MIPS (millions of instructions per second) rating. The DSP instruction set consists of 62 mnemonics that include math, logical, bit-manipulation, loop, and program-control instructions. The math instructions encompass such operations as absolute value, add, subtract, shift left/right, shift left/right and add (useful for implementing the butterfly computation in certain fast Fourier transforms), compare, signed multiply, signed multiply and accumulate, and signed multiply accumulate and round (MACR).

All these instructions—notably some of the math instructions just mentioned—are not pipelined and execute in one instruction cycle (two clock cycles). For example, as the MACR instruction executes, an instruction prefetch, 24- by 24-bit multiply, 56-bit add with convergent rounding, two data moves, and two pointer updates are performed, and all within one instruction cycle. Such powerful instructions are possible because of the parallel operation of the three execution units. These powerful arithmetic instructions, coupled with its high throughput, allows the DSP56001 to literally process data on the fly.

Inside the DSP56001 are four 24-bit bidirectional data buses: X, Y, program, and global. Digital data is split into X and Y components and can be treated as such in two separate 64K-word external memory spaces. On the cube, 24K bytes of static RAM provides 8K words of contiguous scalar data, or 4K words of X and Y data. How this data is ordered in SRAM on the cube is determined by what range of addresses you write into the chip's external memory space.

The two 56-bit accumulators in the data ALU can operate on the X and Y data sets in parallel. Breaking the data into X and Y components provides certain advantages. For example, the data can be treated as X and Y coordinate data for image processing or graphics, or as real and imaginary components for complex math, or as coefficients and data for digital filtering. Each X and Y data bus has an on-chip memory composed of 256- by 24-bit cells that is used to improve performance. The program bus prefetches DSP program instructions into the on-chip program memory. The global bus is used for internal data routing within the DSP.

The DSP56001 has three I/O ports: A, B, and C. Port A has a 24-bit bidirectional data bus, and the address unit can access external memory for off-chip program code or data. Various control lines determine operations such as whether to access program or data memory, X and Y data, and if the operation is a read or a write.

Port B handles 8-bit data to and from a host processor that could be a CPU, DMA (direct memory access) hardware, or even another DSP. Control signals for this bus permit interrupt-driven or DMA transfers of data.

Port C consists of two full-duplex serial ports. The first port is the serial communication interface (SCI) that provides standard asynchronous rates up to 312.5K bits per second, and up to 2.5 megabits per second for synchronous data transmission. Although these signal timings are RS-232C-compatible, the voltage levels range from 0 volts to 5 V, so a line driver is required to produce a true RS-232C signal.

The second port is the synchronous serial interface (SSI) and is a programmable serial interface. You can set the number of bits per word, protocol, clock rate, and mode as required to transfer data at up to 5 megabits per second to and from a variety of peripheral devices.

An example of the DSP56001's processing capability is given by one of Motorola's application notes, where the chip is used as a 10-band graphic equalizer for a digital stereo system. In this document, a compact-disk digital stereo signal (two channels of 16-bit data sampled at 44.1 kHz or 88,200 16-bit digital samples a second) goes through the DSP56001's SSI on port C. Next, real-time digital filtering is performed on 20 bands (10 bands per channel), and the filtered data returns to the stereo system, again via the C port's SSI. This admittedly down-to-earth example shows the processing power that the DSP56001 can bring to bear on a problem. The sampling rate of the DSP56001 depends on the amount of data processing going on at the same time, but it can reach a maximum of 1.66 megawords per second.

As a computer peripheral, you could use the chip in any number of applications: speech synthesis, voice recognition, high-speed modems, image processing, two-dimensional graphics, and real-time filtering of digital data. Although the signed 24-bit resolution may seem limiting for some scientific and engineering applications, you can always use the cube's math coprocessor. But for those problems that do fall within this range, the DSP56001 will be more than adequate.

The NeXT Laser Printer

Let's face it: There are certain situations in your computer work where you must have printed output. NeXT's answer to this problem is a low-cost 400-dot-per-inch laser printer. There's no entry-level dot-matrix printer offered; NeXT is banking on users preferring laser-printed output. Since the cube handles screen imaging with Display PostScript, it also makes sense to take advantage of a high-resolution PostScript-compatible printer. The printer costs $1995.

The NeXT printer is built around a custom-designed laser engine based on the Canon LBP-SX laser engine. It can print eight pages per minute and uses the same toner cartridge as the Apple LaserWriter II printers. A user-selectable printing mode lets the printer produce pages at either 300 or 400 dpi. The printer has its own power cord, and the

power supply is set for 110 volts or 220 V levels with a switch.

The printing process involves imaging the page inside the cube using Display PostScript, and then bit-blasting it to the printer. This is similar to the method used by Apple's LaserWriter IISC, except that the cube uses Display PostScript, and the Mac uses QuickDraw. Since massive amounts of data must be transferred to the printer to produce a page, the printer port has its own direct-memory-access channel.

One limitation of the printer is that it will only work with the cube. Also, you cannot network it like PostScript printers that use Apple's LocalTalk, although you could use a cube with a NeXT laser printer to act as a print server on a network. The cube can print to non-NeXT PostScript printers using its serial ports and Unix printer drivers.

the new technology needed to make this storage device possible was considerable. NeXT admitted that it had literally "gambled the company" on this technology becoming available for use in the cube.

But it did work, and one magneto-optical drive comes standard on the cube. While the drive is designed to boot and run the operating system, its 96-ms average seek time may prove a bottleneck in some applications. For the beta software, if you were using the magneto-optical drive as the system disk, you could not remove the cartridge without rebooting the system. However, NeXT plans to modify the software so you can copy files to another optical cartridge with a single magneto-optical drive.

Optical cartridges are expected to cost $50 initially, although the price may fall as they are produced in volume. Since the cube has room for an extra 5¼-inch full-height device, you can purchase either a second optical drive for $1495, or the 670-megabyte hard disk drive for $3995.

Getting the Picture

As we used the cube, we couldn't help being impressed by the crisp quality of its display. The 17-inch NeXT monochrome monitor has an ample 1120- by 832-pixel display that contains more pixels than most 19-inch monitors (which usually have 1024 by 768 pixels). The monitor has a 94-dpi screen, as compared to the Macintosh's 72-dpi screen. However, this display is only 2 bits (four gray levels). The graphic interface looks very good and makes effective use of the four gray levels.

A 17-inch monitor was chosen for the video display as a compromise between display size and weight. On the monitor's base are two small tractor-style wheels that let you move the monitor easily across a table surface (see photo 5).

The video display has a bandwidth of 100 MHz, with a vertical refresh rate of 68.3 Hz. The monitor uses the positive and negative 12 V DC supplied by the cube's monitor port for power. Inside the monitor's housing are two boards. A step-up transformer on the first board generates the high voltages required to drive the video tube. The second board handles the rest of the I/O managed by the monitor: keyboard, mouse, and sound.

The 84-key keyboard connects to a port located on the monitor's base. The keyboard also has cursor keys, a numeric
continued

ter. The beam travels through the substrate and is reflected off the aluminum backing. However, in a phenomenon known as the Kerr effect, the crystal alignment in the magneto-optical substrate alters the polarization of the reflected beam. The amount of beam polarization determines its intensity as it passes through a polarizing filter to a photodetector. The beam intensity indicates whether a 1 or a 0 was read at the spot on the platter.

The optical drive's I/O processor uses a robust error-correction coding to pro-

tect the integrity of the data read from the platter. (In addition to the 256 megabytes of user data, each cartridge carries a 30 percent overhead just for the error-correction code.) Data and its associated ECC information is read from the disk and fed into one of two 1296-byte buffers located in high-speed SRAM. As the data is checked and corrected for errors, it is transferred to the second buffer. It's the contents of this second buffer that is actually used by the system.

While the operation of the magneto-optical drive seems simple in principle,

This is the year of the dragon.
Harness its power
and reap its good fortune.

it's a year that begins with a powerful array of computer systems designed, engineered, and manufactured exclusively by us...business/professional systems that proudly carry our name.

Leading the way is our new TCS 8000. Designed to harness advanced 80386 microprocessor technology, it is remarkably powerful, extremely fast and extraordinarily versatile. The TCS 8000 gives you all the strength of the MS-DOS,™ concurrent DOS 386,™ PC MOS 386,™ and the incredible capabilities of OS/2.™

The TCS 8000 is equipped with 1MB of memory, but it will easily accommodate up to 16MB on the motherboard by utilizing memory modules...standard modules that dramatically reduce the cost of expanding memory and increasing memory access speed.

As part of a network, in multi-user system, or as a stand-alone system the TCS 8000's performance proves that every detail has met Tatung's uncompromised quality controls.

Like the ancient dragon, the TCS 8000 is a symbol of power and good fortune. It will provide your customers with tremendous value, and you with superior profitability. Indeed, the exciting new TCS 8000 is destined to be a genuine "fire breather."

For complete details and specifications, call toll-free:
1-800-421-2929 Ext. 256.

The remarkable TCS 7000 AT compatible.

The TCS 4000...
AT compatibility in a very small footprint.

TATUNG
Advanced thinking is an ancient art.

TATUNG
80386

Tatung Science & Technology
2060 Ringwood Ave.
San Jose, CA 95131
(408) 435-0140
Outside California:
(800) 421-2929

All trademarks are the property of their respective manufacturers.

Tatung Company of America, 2850 El Presidio Street, Long Beach, California 90810 (213) 979-7055

Display PostScript

Display PostScript is an extension of Adobe's PostScript page-description language (PDL) and is designed as an imaging model for graphics displays. In theory, software developers could write the display portions of their applications just once using Display PostScript: These applications would run without modification on any computer and operating system that supports Display PostScript. Another major benefit is that the image on the screen would reproduce identically on a printer supporting the PostScript PDL.

Display PostScript is device-independent, an important feature when you consider that specific dimensional sizing is display-dependent in most graphics handlers. For example, if you write a Display PostScript routine to draw a 2-inch square on the screen, the routine will always draw a 2-inch square on any display supporting Display PostScript, regardless of the resolution, color capability, or size of the output device. In other words, Display PostScript permits a "non-unitized" description of an image until it is interpreted for a particular display.

This non-unitized approach is in contrast to pixel-based graphics handlers that can only handle proportional sizing. Of course, you can also specify proportional sizing in Display PostScript. Additionally, Display PostScript automatically uses the maximum color capabilities of the host display, whether it has just black and white, or 16 million colors. The programmer does not have to worry about the characteristics of the output device while writing the application.

The core of Display PostScript is called the DPS Kernel. The DPS Kernel is an interpreter that translates Post-Script routines into the images on the screen and is designed to be machine-independent. The DPS Kernel is supplied precompiled in object format to the OEM.

In addition to the Kernel, Adobe supplies the OEM with "front- and back-end" adapters that consist of source code for interfacing to the display devices, the operating system, and the windowing system. The Display Post-Script adapters become part of the host computer manufacturer's system software. Of course, modifying these adapters for the host computer system is not a trivial task, and it is usually undertaken by the OEM as a joint or cooperative effort with Adobe Systems. Again, the important point here is that the software developer need not worry about these "adapters."

The main underlying concept of Display PostScript is to isolate the display operation from not only the host computer's operating system but also from its windowing system. The core of Display PostScript fits inside the host windowing system, which in this case is the NeXT windowing system, although it could be anything from Microsoft Windows to X-Windows to QuickDraw.

While the windowing system handles functions such as cut, paste, and copy, and manages the window boxes on the screen, Display PostScript handles the actual painting of the window's contents. Thus, routines for displaying icons, text fonts, and graphics images have to be written only once using Display PostScript. However, the software developer still has to write separate window calls for each windowing system.

Programmers can use the Display PostScript language directly, or they can use a library of C procedures called PSWrap, which is recognized and interpreted by the DPS Kernel.

NeXT fully supports the PSWrap library, but has added many of its own procedures. Some of these are used by the Application Kit to create and manage windows; other procedures handle events, mouse, and cursor operations; and still others support "compositing."

The compositing procedures are multibit pixel operators designed by NeXT's sister company, PIXAR. Each pixel has two values associated with it: its data value (or color), and its alpha value (the data's transparency or opacity). On the cube's 2-bit display, compositing makes an icon transparent as it moves over another object on the screen. These compositing operators are easily extendable and will allow the NeXT software to migrate to color displays when the time comes.

From brief glimpses of alpha versions of Display PostScript, several industry observers have concluded that Display PostScript has serious performance problems—it is too slow. Adobe Systems vehemently denies this and says critics have jumped to conclusions based on these preliminary demonstrations. Adobe says Display PostScript is very fast provided the code is written properly.

A number of techniques have been developed to improve Display PostScript's performance, including a binary preprocessor (described below), graphic state objects (multiple PostScript graphic states that can be switched quickly by changing a pointer), and user paths (an aggregate of PostScript drawing commands that represent a PostScript path). NeXT uses these techniques and its own compositing functions to boost the speed of the display.

---

keypad, a power-on/power-off key, and pairs of keys that control the volume and screen brightness (pressing one key increases the chosen output; pressing the other decreases it). There are two Command keys and two Alt keys (located on opposite sides of the keyboard) that are mapped separately. There are no PC-style function keys. A two-button opto-mechanical mouse also connects to a port on the keyboard.

There are also left- and right-channel analog stereo jacks, and a jack for stereo headphones on the monitor's base. There's also a jack for a microphone so you can record sounds through the monitor, say, for voice mail. This port uses a telephone codec input that's sampled at 8 kHz, and it uses 8-bit Mu-law scaling for the digitized data. The data is saved within a Sound object that can be utilized by the NeXT Unix mail facility or by NeXT applications.

The Software

As much as the NeXT hardware represents an impressive step forward in areas

As an example of how binary encoding works, say we want to issue the PostScript operator 72 426 moveto. Normally the DPS kernel would have to translate the ASCII digits 72 and 426 into a floating-point format, and the ASCII moveto operator into a binary code. A lookup table uses this binary code in the DPS Kernel to steer execution to the routine that implements the moveto operation. A NeXT application normally calls a PSWrap function, PSmoveto( ), that passes the IEEE 754 floating-point values of the numbers to the DPS Kernel, along with the corresponding binary code for moveto. This effectively eliminates the overhead of the ASCII translation stage for the DPS Kernel. The NeXT DPS Kernel can process ASCII PostScript commands if required.

Display PostScript has one major limitation in that it does not support three-dimensional imaging. It is therefore not suitable for CAD software. Adobe admits that Display PostScript is not intended for high-end mechanical design applications. (Steve Jobs said that NeXT will support the Renderman Standard, which he called "the PostScript of three-dimensional graphics.")

Display PostScript has some very compelling features for software designers and for end users. It could greatly facilitate the porting of software applications across incompatible hardware systems.

But the various competitors for display standards—such as IBM and Apple—will have to make some compromises before Display PostScript can succeed. Until these compromises are made, both the end user and the software developer will continue to be plagued by an incompatible world of competing display standards.

such as digital signal processing, optical disk storage, and VLSI technology, the NeXT system software is a step forward for software technology. The system offers an easy-to-use graphical interface to Unix and an object-oriented programming environment for programmers and software developers.

It's an understatement to say that

NeXT expects Unix to catch on. Steve Jobs told us, "I believe this with every bone of my body: Unix will be the prime operating system of every major company in the 1990s."

So it's not surprising that the cube is a Unix-based system. It features a proprietary windowing system that is designed to shield the Unix command-line interface (CLI) from the user, substituting simple point-and-click mouse operations to manage files and execute applications. NeXT also uses Adobe Systems' PostScript imaging model (often referred to as Display PostScript) for displaying all text and graphics on the screen. Display PostScript is an extension of the PostScript page-description language (see the text box "Display PostScript" at left).

The NeXT system software also includes development tools for building application interfaces and integrating objects into application programs. These tools are called the Interface Builder and Application Kit, respectively.

The Operating System

NeXT uses the Mach Unix kernel developed at Carnegie-Mellon University. The Mach kernel is compatible with BSD (Berkeley Standard Distribution) Unix version 4.3, but provides major enhancements such as shared memory, fast interprocess communication, and potential multiprocessing support through the use of threads. Shared memory allows multiple processes to share common segments of memory. IPC allows processes to communicate with other processes and to transmit messages and data between them. Threads are "lightweight processes" that have their own execution stack, but within the context of a task that created it (i.e., the thread has access to all the resources made available to the parent task such as memory, and opened files).

Multiprocessing support is possible by assigning threads to particular processors. However, multiprocessing is not supported in the initial release of the NeXT operating system. Since you can add multiple CPU boards to the cube's backplane, we can expect to see multiprocessing support in later releases of the operating system.

In NeXT's first release, the operating system consists of a single kernel with the Mach implementation of IPC, scheduling, and virtual memory operating as a layer within the BSD Unix kernel. However, the ultimate goal of the Mach implementation is to provide a modular architecture for Unix that would allow for a much smaller kernel with separate pro-

cesses dedicated to file handling, networking, and Transmission Control Protocol/Internet Protocol (TCP/IP).

Like most Unix-based systems, the cube implements virtual memory using a paged-memory system to allow applications to run even if their memory requirements exceed the available physical memory. Idle portions of a running application are "paged" (i.e., written to disk) in 8K-byte blocks, called pages. However, as in all virtual-memory systems, it is possible to overload the system with too many applications, causing excessive paging or "thrashing," which can bring the system to a crawl. While NeXT was not ready to provide numbers for the amount of memory consumed by the system software, the 8-megabyte base memory configuration is designed to allow "three or four" applications to run

It's an understatement to say that NeXT expects Unix to catch on: The cube is a Unix-based system.

simultaneously in addition to the system software.

For networking, NeXT uses TCP/IP and Sun's Network File System, which has become the standard Unix file-sharing system. Since the cube comes with an Ethernet interface, it is "network-ready" for TCP/IP-based networks. The thin Ethernet cabling allows up to 600 feet of cabling and connection of up to 30 machines without gateways or repeaters. While NFS does not require one, a dedicated server is preferable in networks of more than a few machines, due to performance degradation. In other words, if you're planning to network a bunch of cubes, you'll need a dedicated NFS server, or a cube to serve that purpose.

User Interface and Window Server

NeXT provides a graphical windowing interface to Unix that hides the laborious Unix commands from the user. While veteran Unix users still have the option of

issuing those intuitive commands (like grep and ls) within a Unix CLI window called the Console, most cube users should never have to deal with Unix. The windowing interface, called the Workspace Manager, provides all the necessary functions for file management, opening and closing applications, and communicating with other resources on the system such as peripherals or nodes on the network.

The main interface screen is called the Workspace (see photo 6). Noticeably absent from the screen is the ever-present menu bar found on the Macintosh screen, or on a PC running Windows. Unlike the Macintosh Desktop, menus can be moved anywhere on the Workspace and float above any open windows.

Menus are hierarchical, and you can split off subhierarchies from their parent menus. Windows have scroll bars located on the left and bottom, and there are small boxes on the window frame for resizing or closing the window. It also has a "miniworld" function that collapses the window and its menus into an icon while the process owned by the window continues to run.

Icons become transparent when they overlay other icons, allowing you to always see everything that's currently available on the Workspace. The icons of frequently used applications can be "docked" along the right side of the Workspace for easy recall.

The Workspace is similar to the Desktop metaphor on the Macintosh, and the Workspace Manager is analogous to the Mac's Finder. However, no one will accuse NeXT of copying Apple's look and feel.

> **T**he Workspace resembles the Desktop, but no one will accuse NeXT of copying Apple.

File management operations are similar to those used by the Mac Finder. When you click on a directory in the Workspace, you can examine the directory in a number of ways. There is a "browser window," which displays the directory tree in a window with the directory hierarchy ordered from left to right on the screen. This browser window normally lets you see three levels deep, but you can position the point in the hierarchy where you wish to view files, or resize the window to examine additional levels.

You can also choose to view the directory as icons with subdirectories represented by folders, or as a conventional text-only Unix directory listing.

The version we saw was definitely beta, so the final word on the Workspace will have to wait. Nevertheless, it seems very intuitive and easy to learn. Its performance seems good, and the display quality is excellent.

Rather than use an existing Unix window server such as X-Windows, NeXT designed its own proprietary Window Server. The Window Server manages all interactions between the windows, keyboard, and mouse for all applications attached to it. The Window Server obtains events from the operating system and handles the ones it can (e.g., resizing a window or moving it to another part of the screen). If it's not an event that it can service, the Window Server determines which application and can dispatches it to that application.

Embedded inside the Window Server is the Display PostScript interpreter, which acts on the PostScript commands passed to it. This embedded interpreter executes the PostScript commands it receives and writes the results into the cube's video RAM, making it appear on the monitor.

The Window Server supports Mach IPC connections as well as connections through TCP/IP, allowing other cubes on a network to access another machine's Workspace. The proprietary Window *continued*

Photo 6: *The cube's Workspace. Each window represents a running application. Note that the menu has its own Close box. The icons at the right represent "docked" applications.*
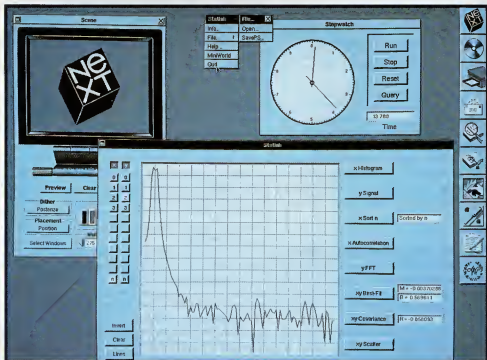
In 1988, $3.5 billion in micro-computer software will be sold worldwide. During that same time, another $3.0 billion in sales will be lost to free distribution — better known as software piracy. And right now, Rainbow Technologies' Software Sentinel™ is protecting close to $1.0 billion in software for developers who never wanted to be part of the free software distribution network in the first place. ⟨⟩ The Software Sentinel hardware key is "execution control" software protection. It ships with the software and simply plugs into the PC's parallel port to be one hundred percent invisible to both user and the soft-ware. Users can make as many copies as they want. Make working submasters. Use a hard disk. Virtually anything that can be done with unprotected software. Except start freely distributing that software to other users. ⟨⟩ The Rainbow fam-lected by the ily of Software Sentinel products. Se-very big to the not-so-big developers of DOS, OS/2 and Xenix software in worldwide markets. To the cool tune of close to a billion dollars. So far.

Come see us at COMDEX, Booth W747.

RAINBOW TECHNOLOGIES

Bundled Software

H ere is a list of the software that is scheduled to be bundled with the NeXT Computer. It includes the Mach operating system and its development software. Also included are the works of Shakespeare, a dictionary, and a thesaurus.

You can call up quotations or the dictionary entry for a specific word at any time by using the cube's Find function. This capability would be valuable not only to college students and faculty members, but to anyone who has to write frequently—whether it's a business proposal or a technical document.

System software:
Mach operating system
PostScript Window Server and fonts
System administration tools

Development tools:
GNU C compiler
GNU debugger
GNU EMACS
Objective-C 4.0
Berkeley Unix utilities
Terminal emulator
Window-based text editor
Interface Builder

Object-oriented software kits:
Application Kit
Sound Kit
Music Kit

High-speed text-retrieval application (called "Find") for
Standard reference works:
  Merriam-Webster's Ninth
    New Collegiate Dictionary
  Merriam-Webster's Collegiate
    Thesaurus
  The Oxford Dictionary
    of Quotations
Documentation for all bundled
  software: user's manuals
  and programmer's manuals
Literature:
  Oxford University Press' William
  Shakespeare: The Complete Works
User-created text files,
  such as mail or documents

Applications:
Personal text database
Electronic mail application
  with graphical interface and ability
  to attach voice messages
Word processor
Window-based file manager
Mathematica
  (Wolfram Research, Inc.)

program. An object consists of data (called instance variables) and executable code. If the object is to be visible on the screen (a window, for example), the code also contains an entity called draw-Self:: that's composed of C code, Objective-C code, and PostScript code, which is used to describe the appearance of the object to the Window Server.

Probably the key concept with respect to user interfaces is that objects can respond directly to messages generated by user actions. Rather than having to write lines of conditional statements in C code to respond to user actions, the user actions are interpreted as messages other objects can understand. For example, you might have an object in your program called "Window," which can understand the "Close" message sent by a user response.

NeXT provides a program called Interface Builder that allows you to interactively build user interfaces for your programs. Interface Builder lets you design the layout of a graphical user interface by selecting buttons, menus, and other objects from an object library to include in your application.

This function is somewhat similar to ResEdit on the Macintosh. However, Interface Builder goes further—it allows you to define connections between objects. That is, Interface Builder lets you specify actions for the objects to perform in response to user actions on other objects. For example, you could build a Beeper button object into your program interface simply by selecting a prototype button from Interface Builder's on-screen inventory, moving it to where you want on the screen, giving it a label, and assigning an action (say, emit a beep) to be performed when a user clicks on the button.

This is similar to the function of HyperTalk in Apple's HyperCard program. The big difference, however, is that Interface Builder generates the binary description of the object that you can integrate into programs.

You can also create custom objects by selecting an object that most closely resembles what you want and customizing its appearance and behavior. NeXT's goal is to supply enough objects so that a programmer could select objects and define their connections, making it possible to built an application from scratch writing little or no code.

In addition to the Application Kit and Interface Builder, the NeXT system software includes kits for working with music and sound. The Music and Sound Kits provide objects for integrating these

Server means that existing applications that run on other Unix windowing systems will have to be modified to run under the NeXT windowing system. However, a Unix application that uses conventional console I/O will run inside the Console window without modification.

The Development Environment
The primary objectives of the NeXT programming environment are to simplify the development of interactive user interfaces and to simplify the creation of new applications through the use of object-oriented programming.

Other systems employing graphical interfaces—like the Macintosh, for example—are great for the end user but extremely complex for programmers, particularly in developing a working user interface. To ease the burden of this task for the developer, the NeXT system in-

cludes tools for building interfaces to the NeXT windowing system, and also tools for object-oriented programming.

The NeXT system software includes an ANSI C compiler and an object-oriented preprocessor called Objective-C, developed by Stepstone Technologies. Objective-C allows you to define objects as groups of C procedures.

NeXT provides several libraries of ready-to-use objects, calles *kits*, for integration into Objective-C programs. These kits provide a library of around 34 objects for implementing the core functionality of a NeXT application, although a programmer would normally use only a small subset of these objects. This library is known as the Application Toolkit, and the Objective-C interface can access it directly.

Object-oriented programming allows a one-to-one correspondence between objects on the screen and objects in your

features into your programs. There is also a number of library functions (not objects) that allow you to tap into the processing capabilities of the DSP. These libraries provide some 50 functions for performing tasks like fast Fourier transforms, and spectral filtering.

NeXT supports the concept of "shared libraries" in its development environment. This means that multiple applications and processes can share a single copy of executable code from the object library. Although library sharing was not implemented when we saw the cube, it should improve performance and reduce the memory and storage requirements of applications.

Applications
NeXT will bundle several applications with the machine. These include the word processor, WriteNow, that is owned by NeXT and is currently distributed by T/Maker for the Macintosh. The system software also includes the standard Unix Mail program equipped with a graphical front end that can attach voice messages to mail files, a file-searching program called Find, C and Objective-C, a symbolic debugger, and on-line documentation. It also has educational and reference tools such as *Webster's Dictionary*, the complete works of Shakespeare, and Mathematica from Wolfram Research (see the text box "Bundled Software" at left). A personal text database allows you to automatically index all your word processing and electronic mail communications so you can recall documents or memos based on keywords instantly.

An important goal of the NeXT software environment is the development of "digital libraries." With its erasable 256-megabyte magneto-optical disk, NeXT hopes to promote the idea of easily accessible text databases. In the educational market, these databases will include encyclopedias, dictionaries, textbooks, and other reference works.

NeXT's first software release lays the groundwork for the company's plans for the nineties. The DSP and the kits for programming it offer exciting possibilities for new real-time applications. It will be interesting to see how the software will be used and what new applications will be developed.

One Giant Step Forward?
The cube is an impressive technical achievement. We liked the carefully thought-out design that didn't just use fast components, but covered every aspect of moving information through the

system. The choice of NuBus for the backplane bus is an excellent one; it goes a long way toward providing the hardware support for the cube's planned multiprocessing capability.

Considering the amount of information that the machine is expected to use, the high-capacity magneto-optical drive is a good design choice. The graphical interface uses the well-documented PostScript imaging language and goes a long

> **C**onsidering the machine's capabilities, we can't help but wonder if NeXT is being too conservative in its marketing plans.

way toward hiding the uglier side of Unix from the user. The facility with which NeXT's object-oriented programming environment reduces the work needed to write an event-driven program is also impressive.

It is indeed a machine for the nineties. It represents a bold step forward both in hardware and software design and effectively redefines what constitutes "standard equipment."

However, as we go to press, some big questions remain unanswered. One relates to the performance of the machine. In our limited time with several beta cubes, it was difficult to judge the overall performance. Display PostScript operations were very fast, putting to rest the controversy of Display PostScript's performance, at least as far as the cube is concerned.

However, disk read/write operations seemed pretty slow—perhaps because so much beta debugging code was being carried along as baggage, and because library sharing was not yet implemented. We saw the magneto-optical disk drive in operation, but it still had some operating bugs, and its 96-ms access time might be a source of frustration if it's used as the main system drive. At this point, we cannot comment on its reliability.

Another question is whether software

developers will support NeXT. The primary obstacle to the acceptance of Unix in the general marketplace has been the lack of software applications. Software developers are faced with choosing between Macintosh, OS/2, DOS, and now a new version of Unix with a proprietary windowing system. To be successful, NeXT will need substantial support from software developers; at the time of our visit, only about 10 developers had signed on, and NeXT would not release their identities.

The concern about outside development is perhaps tempered by two facts. First, the object-oriented environment should simplify moving existing Unix programs to the machine. Second, each cube is a complete development system, since all the development tools—compilers, object libraries, and Interface Builder, are bundled with the machine.

Then there's the question of NeXT's target market—higher education. While the machine is certainly a perfect fit for the university community, universities are not known for being big spenders. Certainly, many students will have a hard time coming up with $6500 or more for a computer, let alone another $1995 or so for the laser printer, and perhaps $1495 for a second magneto-optical drive for backups.

Of the cube's design, Jobs told us, "If you want to make a revolution, you have to raise the lowest common denominator." That's true, but you also have to get the product into the hands of enough revolutionaries to make a difference. Yet it's clear NeXT is thinking small, at least in terms of initial marketing.

Dan'l Lewin (NeXT's vice president of marketing and sales) told us, "We built the company not to need huge numbers." And Jobs said, "We'll focus on other markets in the future, but we're not going to do it today. There's no reason why we can't do very well in [the educational] market alone."

Perhaps. But considering the machine's capabilities, we can't help but wonder if NeXT is being too conservative in its marketing plans. If so, it seems that NeXT may have to be able to endure some lean years until the machine catches on in the early nineties. ∎

Tom Thompson has a BSEE from Memphis State University and is a BYTE senior technical editor at large. He can be contacted on BIX as "tom_thompson."

Nick Baran has a BSME from Stanford University and is a BYTE technical editor based in San Francisco. He can be reached on BIX as "nickbaran."