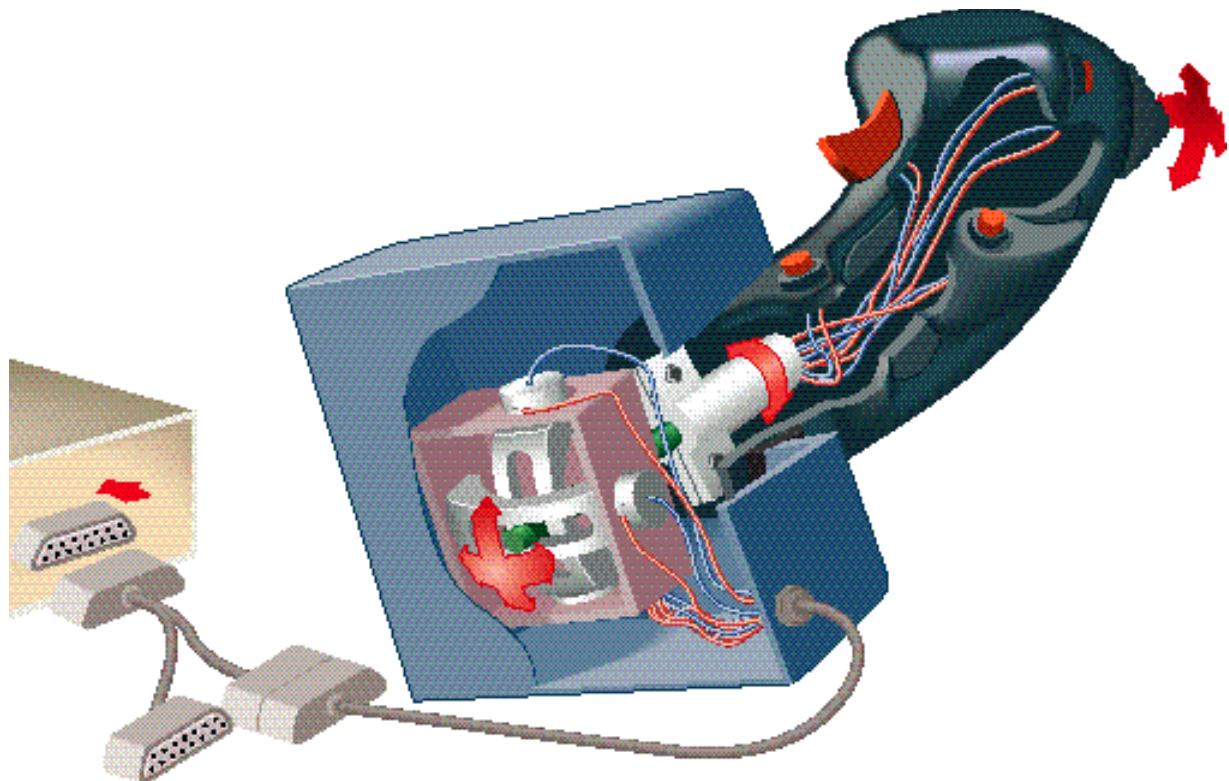
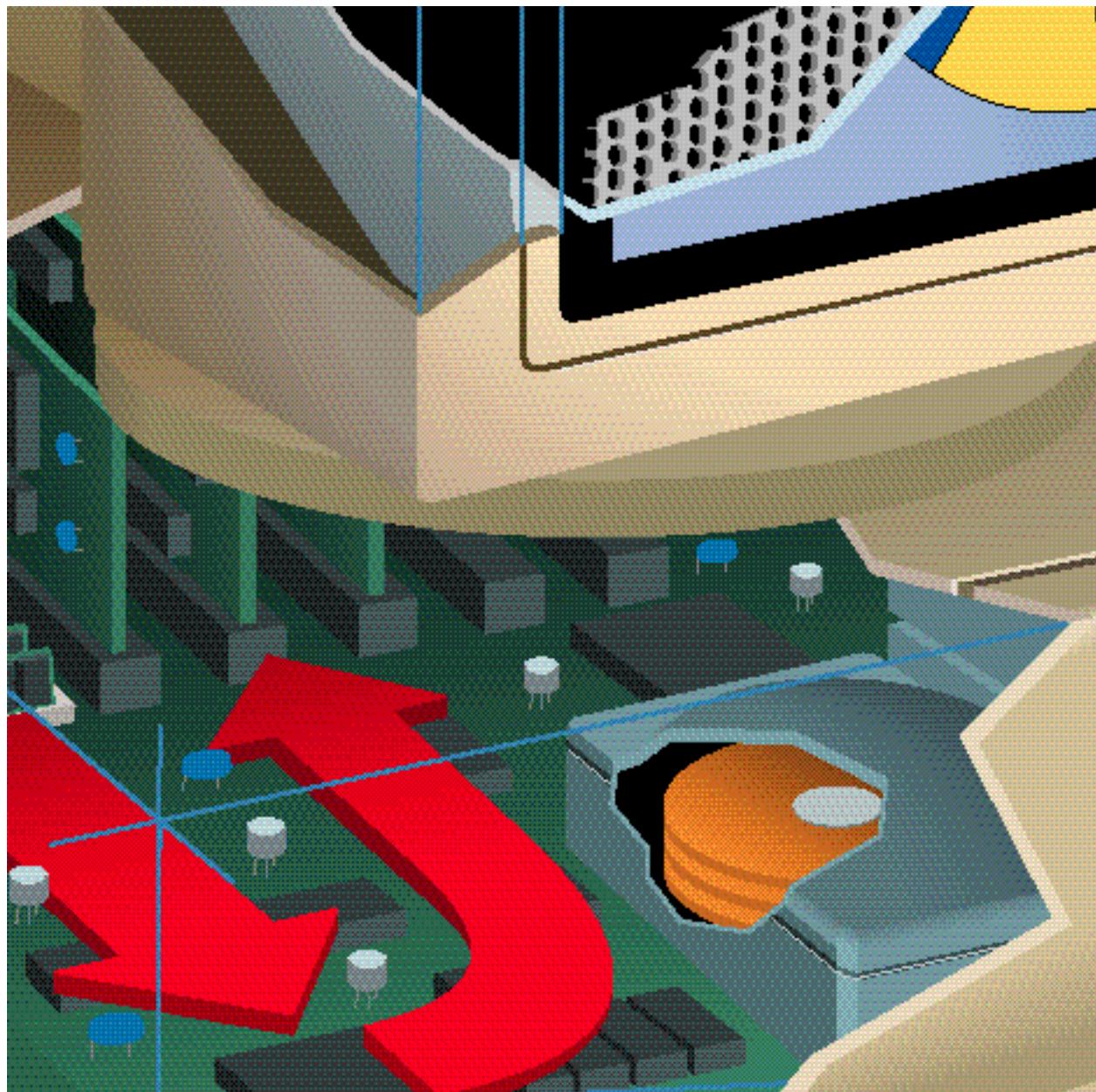


# How Computers Work

Fourth Edition





# How Computers Work

Fourth Edition

Ron White

Illustrated by Timothy Edward Downs, Sarah Ishida,  
and Stephen Adams



A Division of Macmillan Computer Publishing, USA  
201 W. 103rd Street  
Indianapolis, IN 46290

How Computers Work, Fourth Edition  
Copyright 1998 by Que Corporation.  
First printing: September 1998

Publisher	John Pierce
Executive Editor	Angela Wethington
Acquisitions Editor	Renee Wilmeth
Development Editors	Renee Wilmeth and Heather E. Talbot
Managing Editor	Thomas F. Hayes
Project Editor	Heather E. Talbot
Copy Editor	Heather E. Talbot
Technical Illustrators	Timothy Edward Downs, Sarah Ishida, and Stephen Adams
Book Design	Carrie English and Bruce Lundquist
Page Layout	Trina Wurst
Proofreader	John Etchison
Indexer	Lisa Stumpf

All other product names and services identified throughout this book are trademarks or registered trademarks of their respective companies. They are used throughout this book in editorial fashion only and not for the benefit of such companies. No such uses, or the use of any trade name, is intended to convey endorsement or other affiliation with the book.

No part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording, or otherwise, without the prior written permission of Que, except as permitted by the Copyright Act of 1976.

THE INFORMATION AND MATERIAL CONTAINED IN THIS BOOK ARE PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY CONCERNING THE ACCURACY, ADEQUACY, OR COMPLETENESS OF SUCH INFORMATION OR MATERIAL OR THE RESULTS TO BE OBTAINED FROM USING SUCH INFORMATION OR MATERIAL. NEITHER ZIFF-DAVIS PRESS, QUE NOR THE AUTHOR SHALL BE RESPONSIBLE FOR ANY CLAIMS ATTRIBUTABLE TO ERRORS, OMISSIONS, OR OTHER INACCURACIES IN THE INFORMATION OR MATERIAL CONTAINED IN THIS BOOK, AND IN NO EVENT SHALL ZIFF-DAVIS PRESS OR THE AUTHOR BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF SUCH INFORMATION OR MATERIAL. SO THERE.

Manufactured in the United States of America

Library of Congress Catalog No.: 98-85290

ISBN: 0-7897-1728-X

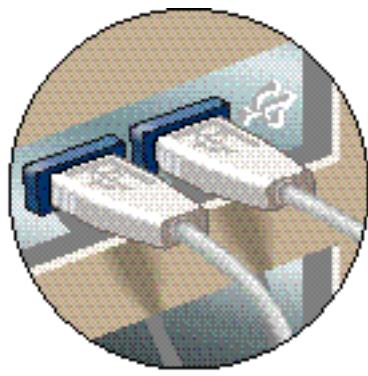
This book is sold as is, without warranty of any kind, either express or implied, respecting the contents of this book, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither Que Corporation nor its dealers or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by this book.

00 99 98 6 5 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 98-1 shows that the first printing of the book occurred in 1998.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark. Que is a trademark of Macmillan Computer Publishing.

This book was written on a Hewlett-Packard 7370V, a Gateway Solo 2100, and an IBM Aptiva Pentium II using WordPerfect, Microsoft Word, Professional Capture System, Paint Shop Pro, Notes 4.5, Windows Explorer, Netscape Navigator, and a BitsURFR ISDN connection. Author's color proofs produced with Hewlett-Packard Officelet Pro 1150C. It was produced on a Macintosh Quadra 900 and a IIfx, with the following applications: Adobe Illustrator, QuarkXPress, Microsoft Word, MacLink Plus, Adobe Photoshop, FrameMaker, Aldus FreeHand, and Collage Plus.



For Shannon and Michael, who  
always kept me honest in my  
explanations, who never took  
“ ‘Cause” for an answer.

Introduction			
<b>Part 1: Boot-Up Process</b>	<b>1</b>	<b>Part 4: Data Storage</b>	<b>68</b>
Chapter 1		Chapter 12	72
How a Computer Wakes Up	4	How Disk Storage Works	
Chapter 2		Chapter 13	82
How a Disk Boot Works	8	How a Floppy Drive Works	
<b>Part 2: How Hardware</b>		Chapter 14	
<b>and Windows 98 Work</b>		How a Hard Drive Works	86
<b>Together</b>	<b>14</b>	Chapter 15	
Chapter 3		How Disk Drives Increase Speed	
How Windows 98 Controls Hardware	18	and Storage	94
Chapter 4		Chapter 16	
How the BIOS and Drivers Work	22	How Other Removable Storage Works	102
Chapter 5		Chapter 17	
How Plug and Play Works	26	How a Tape Backup Works	108
Chapter 6		<b>Part 5: Input/Output</b>	
How Windows 98 Uses Memory	30	<b>Devices</b>	<b>114</b>
<b>Part 3: Microchips</b>	<b>34</b>	Chapter 18	
Chapter 7		How a Bus Works	118
How a Transistor Works	38	Chapter 19	
Chapter 8		How Computer Ports Work	126
How RAM Works	42	Chapter 20	
Chapter 9		How a Keyboard Works	138
How a Computer Performs Math	50	Chapter 21	
Chapter 10		How a Computer Display Works	142
How CISC and RISC Processors Work	54		
Chapter 11			
How a Microprocessor Works	58		

Chapter 22		Part 6: Multimedia	200
How Pointing Devices Work	158	Chapter 28	
Chapter 23	164	How CD-ROM and DVD Work	204
How Game Controllers Work		Chapter 29	
Chapter 24	170	How Multimedia Sound Works	
How a Modem Works		Chapter 30	
Chapter 25	178	How Multimedia Video Works	224
How a PC Card (PCMCIA) Works		Chapter 31	
Chapter 26	182	How Virtual Reality Works	228
How Scanners and Optical Character Recognition Work		Part 7: How the Internet Works	238
Chapter 27	190	Chapter 32	
How High-Tech Input/Output Works		How Local Area Networks Work	242
		Chapter 33	
		How a LAN Connects to the Internet	250
		Chapter 34	
		How E-Mail Works	256
		Chapter 35	
		How Internet Video and Audio Work	260
		Chapter 36	
		How the World Wide Web Works	266
		Part 8: How Printers Work	274
		Chapter 37	
		How Fonts Work	278
		Chapter 38	
		How a Dot-Matrix Printer Works	284
		Chapter 39	
		How a Laser Printer Works	288
		Chapter 40	
		How Color Printing Works	292
		Index	304

**PERSONAL** computers have changed dramatically since the first edition of this book in 1993. Then, 5-1/4-inch floppy drives were still common, an Intel 80386 was a hot processor, and it was still a toss-up to see whether Windows or some other operating system would dominate PCs. Really. What wasn't even mentioned in the first edition and merited only a couple of chapters in the second edition—the Internet—has become an entire section that grows yearly. Many of the topics covered in this edition were not dreamed of back then—gas plasma displays, Internet video and audio, rewriteable DVD drives, accelerated 3-D graphics, the universal serial bus, global satellite positioning, and computers—real ones, not toys—that fit in your shirt pocket and let you write on them. To the extent back then that something like speaking to your computer had been imagined, it was still something of science fiction, something for tomorrow. Well, tomorrow's here.

The change in PCs has been so rapid, so extensive, and so detailed, I could never have learned the tricks behind the new technologies without the help of a lot of knowledgeable people who were not only generous with that knowledge, but patient enough to explain it to me until the mental LED lit up.

I had the privilege of launching the How It Works feature in PC Computing in 1989, but over the years, many people have worked on the series, and several chapters are based on the research and explanations they've done. Deep thanks go to Margaret Ficklen, Herb Brody, Brett L. Glass, Marty Jerome, Raymond Jones, Matthew Lake, Liesl La Grange Noble, Jack Nimersheim, Randy Ross, Stephen Sagman, Jan Smith, Dylan Tweney, Doug van Kirk, Mark L. Van Name and Bill Catchings, Christine Grech Wendum, and Kenan Woods. Thanks to Preston Gralla for too many things to list.

I'm also grateful to the dozens of people in the PC industry who've shared their knowledge, schematics, and white papers to infuse How Computers Work with detail and accuracy. I'm particularly grateful to Joe Vanderwater, Bryan Langerdoff, Dan Francisco, Tami Casey, John Hyde, Bill Kircos, Susan Shaw, and Seth Walker of Intel; Russell Sanchez and Adam Kahn of Microsoft; Jim Bartlett, Ellen Reid Smith, Tim Kearns, and Desire Russell for IBM, Barrett Anderson and Todd-Hollinshead at id Software; Marcie Pedrazzi of Adaptec; Eileen Algaze at Rockwell; A.J. Rodgers, Jennifer Jones, and Susan Bierma of Tektronix; Ben Yoder and Lisa Santa Anna of Epson; Dewey Hou of TechSmith Corporation; Ray Soneira, author of Sonera Technologies' DisplayMate, Kathryn Brandberry and Doyle Nicholas of Thrustmaster; Dani Johnston of Microtek Lab; Lisa Tillman of The Benjamin Group; Dvorah Samansky for Iomega, Brandon Talaich of SanDisk, Chris Walker with Pioneer USA, Carrie Royce of Cirque Corporation, Andy Marken of Marken Communications, and Tracy Laidlaw of Beyond Words. If there are any mistakes in this book, I can't lay the blame on them.

In many ways, writing this edition of How Computers Work has been made much easier by the Internet. I didn't keep a complete list of every place on the Net I found useful information, but a couple of sites were particularly outstanding: Dictionary of PC Hardware and Data Communications Terms by Mitchell Shnier at <http://www.ora.com/reference/dictionary/tsearch.cgi> and the Free On-line Dictionary of Computing maintained by Paul Mayer at <http://wagner.princeton.edu/fodoc/index.html>.

And thanks to the many staff members at PC Computing who shared their knowledge and time and to former PCC Publisher, Mike Edelhart, who started the ball rolling on this book. A special thanks to Jim Seymour and Nora Georgas for starting me on the road to excitement and danger that dog the professional computer journalist. At Que to John Pierce, Angie Wethington, Juliet Langley, Cindy Hudson, Melinda Levine, and Lysa Lewallen for encouragement and advice, and especially Renee Wilmeth and Heather Talbot for pulling everything together. Again. Thanks to John Rizzo for tech editing and researching several chapters. And thanks to my wife, Sue, for her constant encouragement, spirit-raising, and humor while waiting for me to pop out of writing mode.

I learned long ago that a writer's skill depends largely on how well the writer can steal from others. In addition to just about every book in the Que How It Works series, four other books were invaluable for details on PC innards: Understanding I/O Subsystems by W. David Schwaderer and Andrew W. Wilson, Jr.; Inside the IBM PC by Peter Norton; The PC Configuration Handbook by John Woram; and The Winn Rosch Hardware Bible by Winn L. Rosch. Also helpful was The Way Things Work by David Macaulay, not only for its informative explanations of computers, but for its inspiring examples of how to combine text and art into clear explanations.

Finally, this book would not be what it is without the artwork of Timothy Edward Downs, Sarah Ishida, and Stephen Adams. Invariably, they transformed my Neanderthal sketches into clear, informative illustrations, but also managed to make them into wonderful works of art. As the original artist for the book and in his continuing work at PC Computing, Tim especially has created a new, visual way to communicate technology.



Any sufficiently advanced technology is indistinguishable from magic.

—Arthur C. Clarke

SORCERERS have their magic wands—powerful, potentially dangerous tools with a life of their own. Witches have their familiars—creatures disguised as household beasts that could, if they choose, wreak the witches’ havoc. Mystics have their golems—beings built of wood and tin brought to life to do their masters’ bidding.

We have our personal computers.

PCs, too, are powerful creations that often seem to have a life of their own. Usually they respond to a seemingly magical incantation typed at a C:> prompt or to a wave of a mouse by performing tasks we couldn’t imagine doing ourselves without some sort of preternatural help. But even as computers successfully carry out our commands, it’s often difficult to quell the feeling that there’s some wizardry at work here.

And then there are the times when our PCs, like malevolent spirits, rebel and open the gates of chaos onto our neatly ordered columns of numbers, our carefully wrought sentences, and our beautifully crafted graphics. When that happens, we’re often convinced that we are, indeed, playing with power not entirely under our control. We become sorcerers’ apprentices, whose every attempt to right things leads to deeper trouble.

Whether our personal computers are faithful servants or imps, most of us soon realize there’s much more going on inside those silent boxes than we really understand. PCs are secretive. Open their tightly sealed cases and you’re confronted with poker-faced components. Few give any clues as to what they’re about. Most of them consist of sphinx-like microchips that offer no more information about themselves than some obscure code printed on their impenetrable surfaces. The maze of circuit tracings etched on the boards is fascinating, but meaningless, hieroglyphics. Some crucial parts, such as the hard drive and power supply, are sealed with printed omens about the dangers of peeking inside, omens that put to shame the warnings on a pharaoh’s tomb.

This book is based on two ideas. One is that the magic we understand is safer and more powerful than the magic we don’t. This is not a hands-on how-to book. Don’t look for any instructions for taking a screwdriver to this part or the other. But perhaps your knowing more about what’s going on inside all those stoic components makes them all a little less formidable when something does go awry. The second idea behind this book is that knowledge, in itself, is a worthwhile and enjoyable goal. This book is written to respond to your random musings about the goings-on inside that box that you sit in front of several hours a day. If this book puts your questions to rest—or raises new ones—it will have done its job.

At the same time, however, I’m trusting that knowing the secrets behind the magician’s legerdemain won’t spoil the show. This is a real danger. Mystery is often as compelling as knowledge. I’d hate to think that anything you read in this book takes away that sense of wonder you have when you manage to make your PC do some grand, new trick. I hope that, instead, this book makes you a more confident sorcerer.

# Before You Begin

This book has been written with a certain type of personal computer in mind—the “Wintel,” a PC most often built around an Intel processor and running Microsoft Windows. Many of the specifics in these explanations apply only to that class of computer and those components. For Mac users, I suggest John Rizzo’s *How the Mac Works*, and that you do some serious thinking about switching.

In more general terms, the explanations also may apply to Macintosh computers, UNIX workstations, and even minicomputers and mainframes. But I’ve made no attempt to devise universal explanations of how computers work. To do so would, of necessity, detract from the understanding that comes from inspecting specific components.

Even so, there is so much variety even within the Intel/Microsoft world of PCs that, at times, I’ve had to limit my explanations to particular instances or stretch the boundaries of a particular situation to make an explanation as generic as possible. If you spot anything that doesn’t seem quite right in this book, I hope that my liberties with the particulars is the only cause.

Ron White  
San Antonio, Texas

# Critical Acclaim for Ron White's How Computers Work!

"A pleasure to read. White makes even the most complex technologies simple and accessible."

—Bill Machrone, Vice President, Technology Columnist and Contributing Editor, PCMagazine and PCWeek

"Ron White's book is easily the best—and most understandable—work of its type. Its spectacular and easy to follow illustrations are coupled with clear, complete text to provide an engaging look at the guts of these amazing machines. When novices ask me for a recommendation ... I always point them to *How Computers Work*."

—Dwight Silverman, the Houston Chronicle

"Do you wonder how all that stuff happens inside your computer and the Internet? Well, pick up a copy of this book, and wonder no more!"

—Jim Louderback, Editorial Director, ZDTV

"A magnificently seamless integration of text and graphics that makes the complicated physics of the personal computer seem as obvious as gravity. When a book really pleases you—and this one does—there's a tendency to gush, so let's put it this way: I haven't seen any better explanations written (including my own) of how a PC works and why."

—Larry Blasko, The Associated Press

"Read *How Computers Work* to learn about the inner workings of the IBM and PC-compatible."

—Ronald Rosenberg, Boston Globe

"Ron White has been explaining how computers work for the better part of a decade. This new edition brings us squarely into modern times with White's unique combination of clarity and imaginativeness."

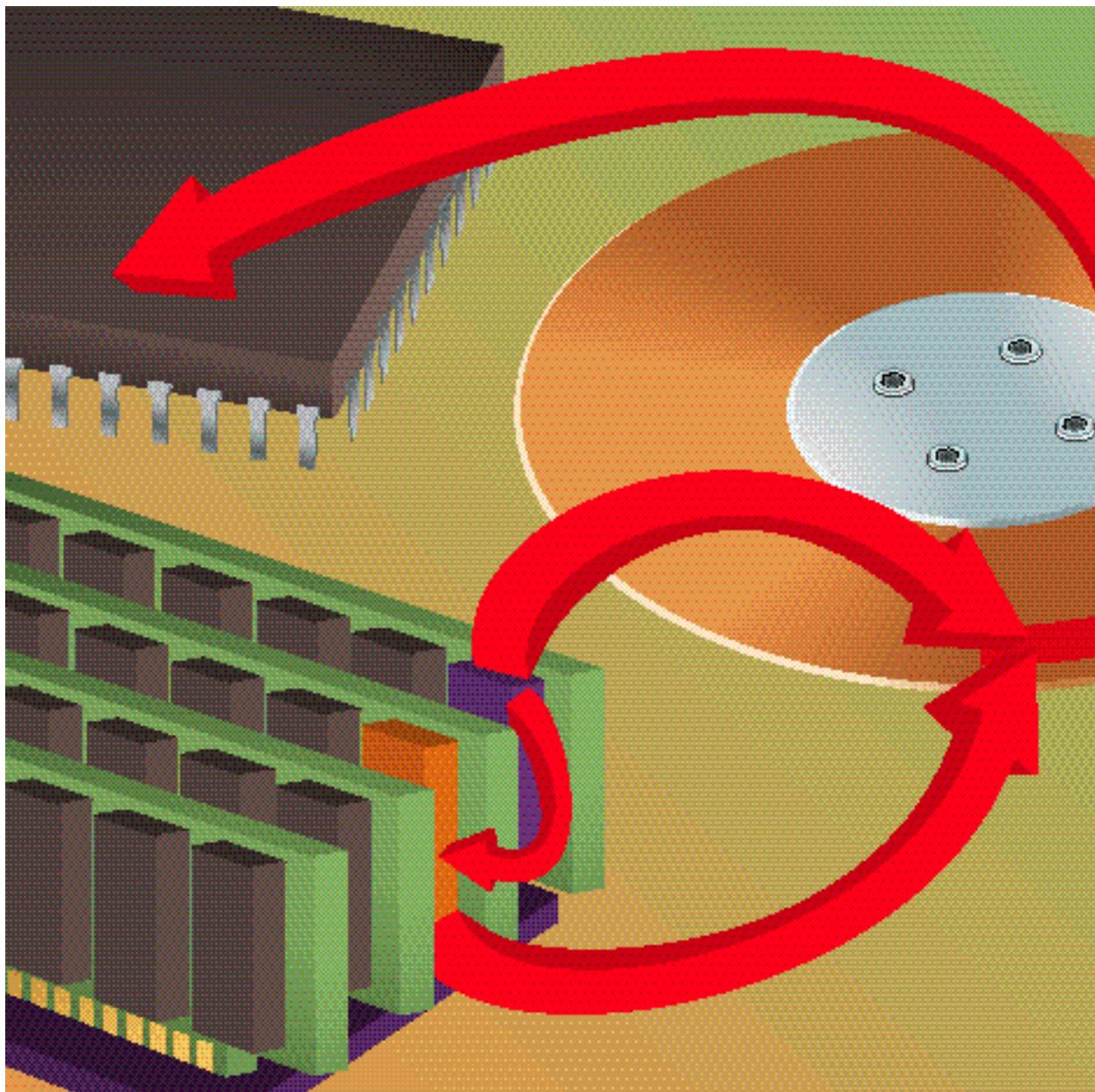
—Robin Raskin, Editor in Chief, FamilyPC

"If you're curious but fear computerese might get in the way, this book's the answer ... it's an accessible, informative introduction that spreads everything out for logical inspection. Readers will come away knowing not only what everything looks like but what it does."

—Stephanie Zvirin, Booklist

"A 'real' book, and quite a handsome one ... The artwork, by Mr. Timothy Edward Downs, is striking and informative, and the text, by Mr. White, [executive editor of PC/Computing], is very lucid."

—L.R. Shannon, New York Times



P A R T



# BOOT-UP PROCESS

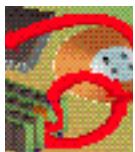
Chapter 1: How a Computer Wakes Up  
4

Chapter 2: How a Disk Boot Works  
8

**BEFORE** your personal computer is turned on, it is a dead collection of sheet metal, plastic, metallic tracings, and tiny flakes of silicon. When you hit the On switch, one little burst of electricity—only about three to five volts—starts a string of events that magically brings to life what otherwise would remain an oversized paperweight.

Even with that spark of life in it, however, the PC is still pretty stupid at first. It has some primitive sense of self as it checks to see what parts are installed and working, like those patients who awake from a coma and check to make sure that they have all their arms and legs and that all their joints still work. But beyond taking inventory of itself, the newly awakened PC still can't do anything really useful, certainly nothing we'd think of as even remotely intelligent.

At best, the newly awakened PC can search for intelligence—intelligence in the form of an operating system that gives structure to the PC's primitive, amoebic existence. Then comes a true education in the form of application software—programs that tell the PC how to do tasks faster and more accurately than we could; a student who's outstripped the teacher.



But not all kinds of computers have to endure such a torturous rebirth each time they're turned on. You encounter daily many computers that spring to life fully formed at the instant they're switched on. You may not think of them as computers, but they are: calculators, your car's electronic ignition, the timer in the microwave, and the unfathomable programmer in your VCR, for example. The difference between these and the big box on your desk is hard wiring. Computers built to accomplish only one task are hard-wired. They are efficient about doing that task, but that means that they are more idiot savant than sage.

What makes your PC such a miraculous device is that each time you turn it on, it is a tabula rasa, capable of doing anything your creativity—or, usually, the creativity of professional programmers—can imagine for it to do. It is a calculating machine, an artist's canvas, a magical typewriter, an unerring accountant, and a host of other tools. To transform it from one persona to another merely requires setting some of the microscopic switches buried in the hearts of the microchips, a task accomplished by typing a command or by clicking with your mouse on some tiny icon on the screen.

Such intelligence is fragile and short-lived. All those millions of microscopic switches are constantly flipping on and off in time to dashing surges of electricity. All it takes is an errant instruction or a stray misreading of a single switch to send this wonderful golem into a state of catatonia. Or hit the Off switch and what was a pulsing artificial life dies without a whimper.

Then the next time you turn it on, birth begins all over again.

## KEY CONCEPTS

**BIOS** (basic input/output system) A collection of software codes built into a PC that handle some of the fundamental tasks of sending data from one part of the computer to another.

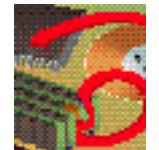
**boot or boot-up** The process that takes place when a PC is turned on and it performs the routines necessary to get all the components functioning properly and then to load the operating system. The term comes from the concept of lifting yourself by your bootstraps.

**clock** A microchip that regulates the timing and speed of all the computer's functions. The chip includes a crystal that vibrates at a certain frequency when electricity is applied to it. The shortest length of time in which a computer can perform some operation is one clock, that is, one vibration of the clock chip. The speed of clocks—and therefore computers—is expressed in megahertz (MHz). One megahertz is 1 million cycles, or vibrations, per second. Thus a PC may be described as having a 200 or 300MHz processor, which means that the processor has been designed to work with a clock chip running at that speed.

**CMOS** An abbreviation for complementary metal-oxide semiconductor—a term that describes how a CMOS microchip is manufactured. Powered by a small battery, the CMOS chip retains crucial information about what hardware a PC comprises even when power is turned off.

**CPU** An abbreviation for central processing unit, it means the microprocessor—also, processor—which is a microchip that processes the information and the code (instructions) used by a computer. The “brains” of a computer.

**POST** An acronym for Power-On Self-Test, a procedure the computer goes through when booting to verify that the basic components of a PC are functioning.



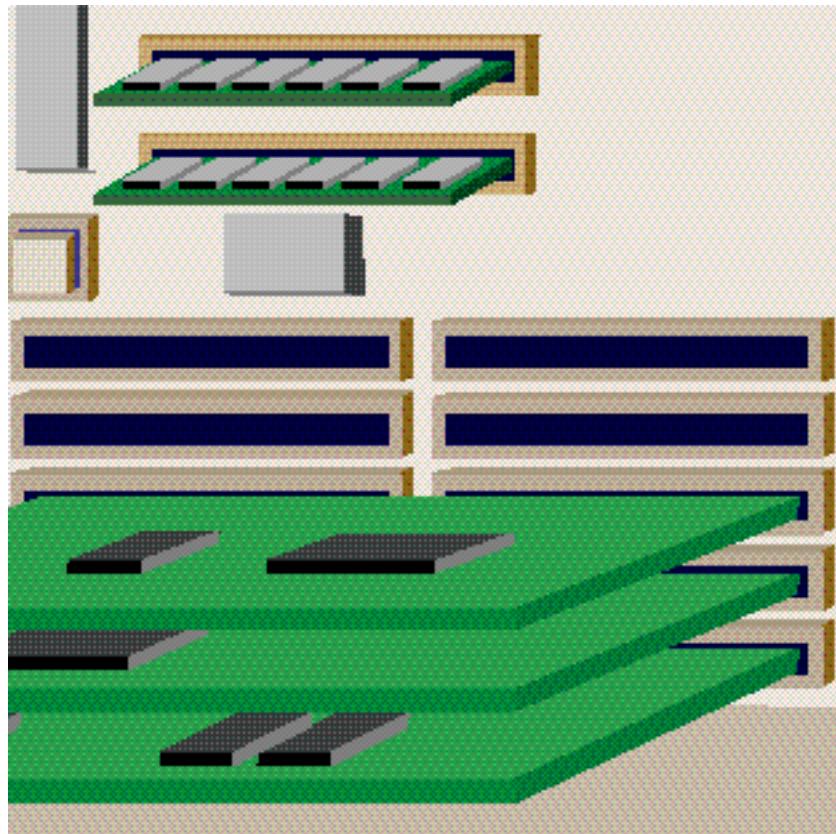
**ROM and RAM** Acronyms for read-only memory and random access memory. ROM is memory chips or data stored on disks that can be read by the computer's processor. The PC cannot write new data to those chips or disk drives. RAM is memory or disks that can be both read and written to. (Random access memory is really a misnomer because even ROM can be accessed randomly. The term originally was used to distinguish RAM from data and software that was stored on magnetic tape, and which could be accessed only sequentially. That is, to get to the last chunk of data or code on a tape, a computer must read through all the information contained on the tape until it finds the location where it stored the data or code it is looking for. In contrast, a computer can jump directly to any information stored in random locations in RAM chips or on disk.)

**system files** Small disk files containing software code that are the first files a computer reads from disk when it is booted. On DOS and Windows systems, the files are named IO.SYS and MSDOS.SYS and are hidden so that ordinarily you cannot see them in a listing of files on a disk. The system files contain the information needed, following the initial hardware boot, to load the rest of an operating system. In DOS, one other system file is COMMAND.COM, which contains the operating system's basic functions, such as displaying a list of files (directory). A boot disk must contain all three files for a PC to start up. System files may also include CONFIG.SYS, which makes some initial settings of hardware, and AUTOEXEC.BAT, a collection of commands that are executed when all other boot functions are through. In Windows 95 and 98, the Registry—consisting of the two hidden files, USER.DAT and SYSTEM.DAT—is also necessary for Windows to run and may be considered a system file.

CHAPTER

1

# How a Computer Wakes Up



**WHEN** you hit your PC's On switch, nothing much seems to happen for several seconds.

Actually, your computer is going through a complex set of operations to make sure all its components are working properly and to warn you if something is amiss. This operation is the first step in an even more complicated process called the boot-up, or simply, the boot.

The term comes from the idea of lifting yourself up by your own bootstraps. In a PC, bootstrapping is necessary because the PC must have some way of bringing all its components to life long enough to load an operating system. The operating system then takes on more complicated tasks that the boot code alone can't manage, such as making the PC's hardware interact with software.

But before your PC can even attempt to load an operating system, it has to make sure that all the hardware components are running and that the CPU (central processing unit) and memory are functioning properly. This is the job of the power-on self-test, or POST.

The POST is the first task your PC performs when you turn it on, and it's your first warning of trouble with any of the components. When the POST detects an error from the display, memory, keyboard, or other basic components, it produces an error warning in the form of a message on your display and—in case your display is part of the problem—in the form of a series of beeps. Usually neither the beeps nor the onscreen message is specific enough to tell you exactly what is wrong. All they're intended to do is to point you in the general direction of the component that has a problem.

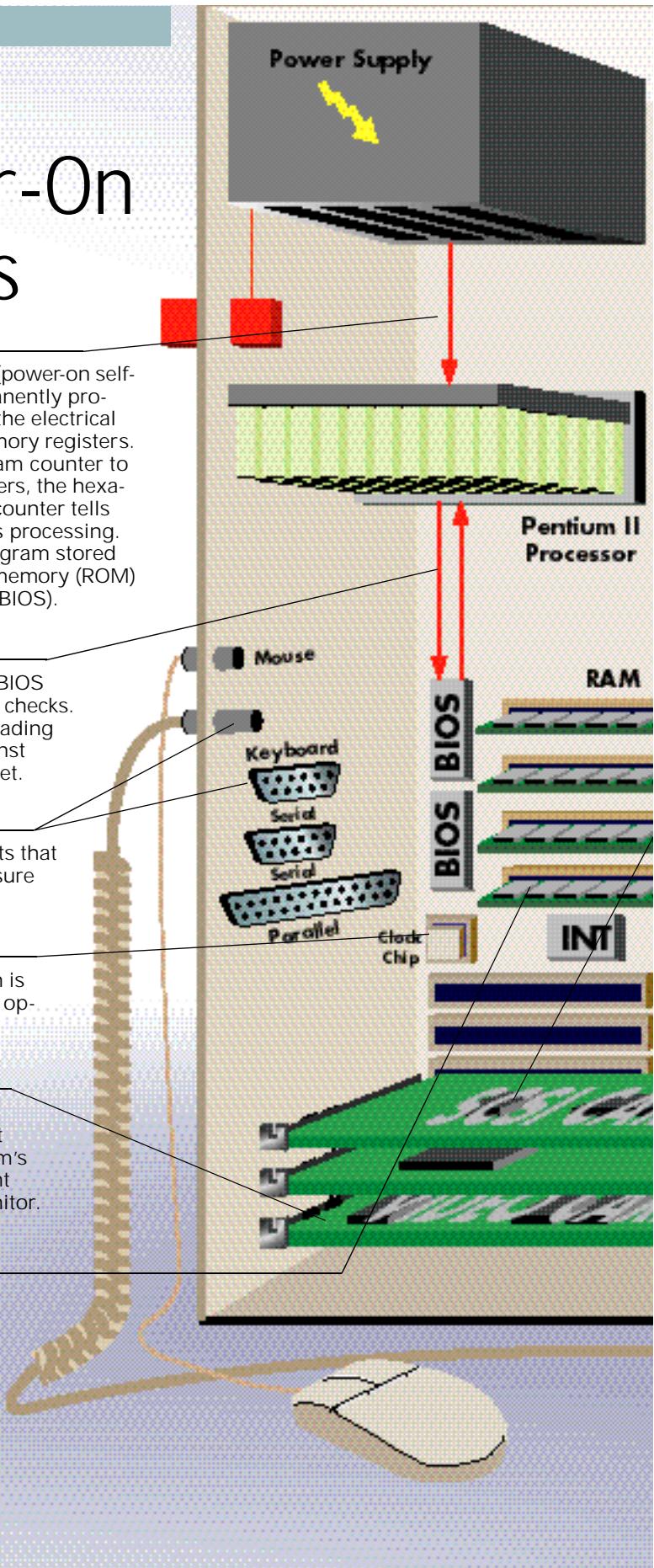
A single beep combined with a display of the normal DOS prompt means that all components have passed the POST. But any other combination of short beeps and long beeps usually means trouble. Even no beep at all indicates a problem.

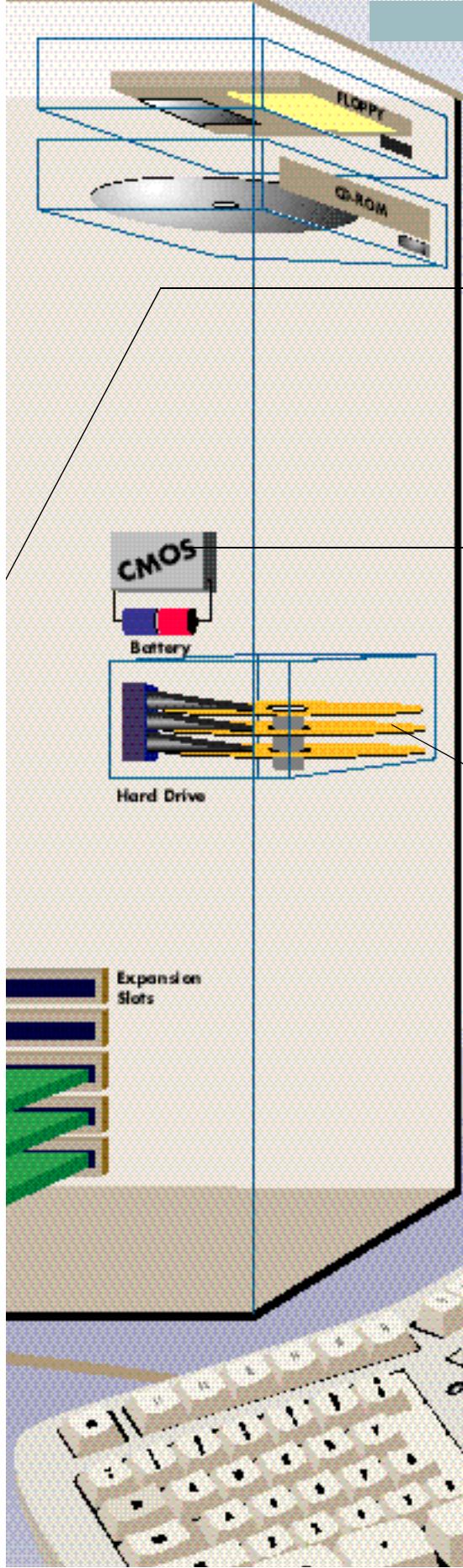
If no error message appears or beeps occur, however, that doesn't mean all the hardware components of your system are functioning as they should. The POST is capable of detecting only the most general types of errors. It can tell if a hard drive that's supposed to be installed isn't there, but it can't tell if there is trouble with the drive's formatting.

All in all, the POST does not appear to be extremely helpful. That's because most PCs function so reliably that only occasionally something triggers a POST alarm. The POST's benefits are subtle but fundamental. Without it, you could never be sure of the PC's ability to carry out its tasks accurately and reliably.

# How the Power-On Self-Test Works

- 1 When you turn on your PC, a process called the POST (power-on self-test) begins with an electrical signal following a permanently programmed path to the CPU, or microprocessor. There, the electrical signal clears leftover data from the chip's internal memory registers. The signal also resets a CPU register called the program counter to a specific number. In the case of ATs and later computers, the hexadeciml number is F000. The number in the program counter tells the CPU the address of the next instruction that needs processing. In this case, the address is the beginning of a boot program stored permanently at the address F000 in a set of read-only memory (ROM) chips that contain the PC's basic input/output system (BIOS).
- 2 The CPU uses the address to find and invoke the ROM BIOS boot program, which in turn invokes a series of system checks. The CPU first checks itself and the POST program by reading code loaded into various locations and checking it against identical records stored permanently in the BIOS chip set.
- 3 The CPU sends signals over the system bus—the circuits that connect all the components with each other—to make sure that they are all functioning.
- 4 The CPU also checks the system's timer, or clock, which is responsible for pacing signals to make sure all the PC's operations function in a synchronized, orderly fashion.
- 5 The POST tests the memory contained on the display adapter and the video signals that control the display. It then makes the adapter's BIOS code a part of the system's overall BIOS and memory configuration. It's at this point that you'll first see something appear on your PC's monitor.
- 6 The POST runs a series of tests to ensure that the RAM chips are functioning properly. The tests write data to each chip, then read it and compare what they read with the data sent to the chips in the first place. On some PCs at this point, you'll see on the monitor a running account of the amount of memory that's been checked.

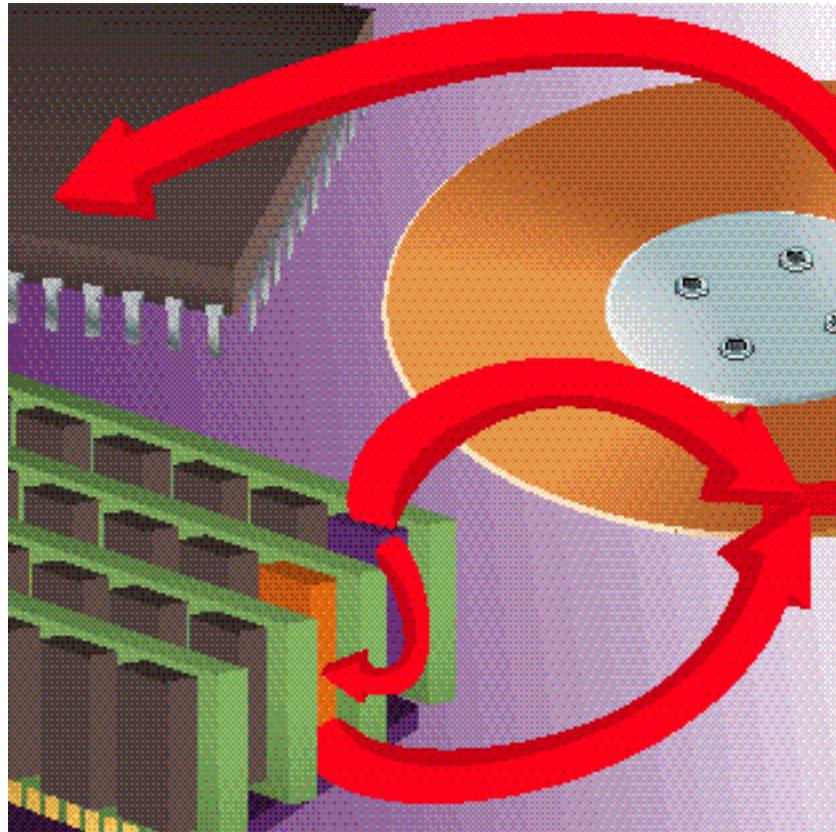




- 10 Some systems' components, such as a SCSI controller card, contain a BIOS which interprets commands from the processor to control that hardware. Those components' BIOS codes are incorporated as part of the system's own overall BIOS. Sometimes these BIOS codes are copied from the slow CMOS BIOS chip to the PC's faster RAM. Newer PCs may also run a Plug and Play operation to distribute system resources among different components. (See Chapter 5.) The PC is now ready to take the next step in the boot process: loading an operating system from disk.
- 9 The results of the POST tests are compared with data in a specific CMOS chip that is the official record of which components are installed. CMOS is a type of memory chip that retains its data when power is turned off, as long as it receives a trickle of electricity from a battery. Any changes to the basic system configuration must be recorded in the CMOS setup. If the tests detect new hardware, you're given a chance to update the configuration on the setup screen.
- 8 The POST sends signals over specific paths on the bus to the floppy and hard disk drives and listens for a response to determine which drives are available.
- 7 The CPU verifies that the keyboard is attached properly and determines whether any keys have been pressed.

CHAPTER

## 2

How a Disk  
Boot Works

A personal computer can't do anything useful unless it's running an operating system—a basic type of software that acts as a supervisor for all the applications, games, or other programs you use. The operating system sets the rules for using memory, drives, and other parts of the computer. But before a PC can run an operating system, it needs some way to load the operating system from disk to random access memory (RAM). The way to do this is with the bootstrap or simply to boot—a small amount of code that's permanently a part of the PC.

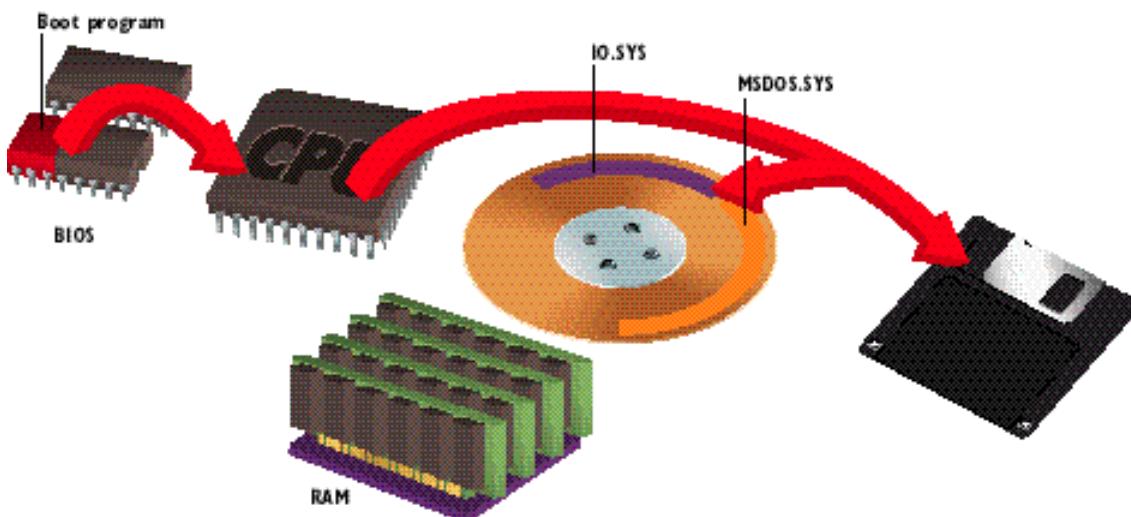
The bootstrap is aptly named because it lets the PC do something entirely on its own, without any outside operating system. Of course, the boot operation doesn't do much. In fact, it has only two functions: one is to run a POST, or power-on self-test described in the preceding chapter, and the other is to search drives for an operating system. When these functions are complete, the boot operation launches the process of reading the system files and copying them to random access memory.

Why do PCs use such a round-about arrangement? Why not simply make the operating system a part of the PC? A few low-end or specialized computers do this. Early computers used primarily for playing games, such as the Atari 400 and 800, and the more recent palm-sized PCs, have a permanent operating system. But in most cases, the operating system is loaded from hard disk for two reasons.

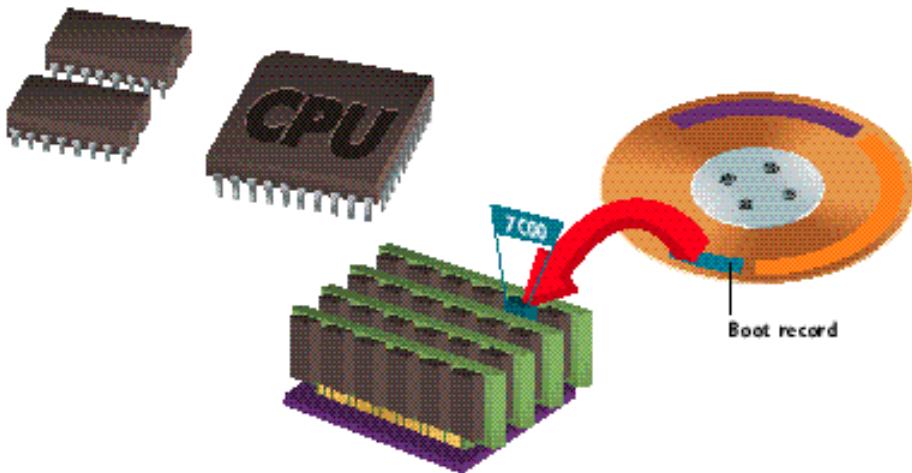
It is simpler to upgrade the operating system when loading from a disk. When a company such as Microsoft—which makes MS-DOS and Windows, the most commonly used PC operating systems—wants to add new features or fix serious bugs, it can simply issue a new set of disks. Sometimes all that's necessary is a single file that patches a flaw in the operating system. It's cheaper for Microsoft to distribute an operating system on disk than to design a microchip that contains the operating system. And it's easier for computer users to install a new operating system from disk than it is to swap chips.

The other reason for loading an operating system from disk is that it gives users a choice of operating systems. Although most PCs based on microprocessors built by Intel use Windows or MS-DOS, there are alternative operating systems, such as Windows NT, OS/2, DR DOS, and UNIX. In some PC setups, you can even choose which of the operating systems to use each time you turn on your computer. We'll use DOS/Windows here because it's typical of all operating systems.

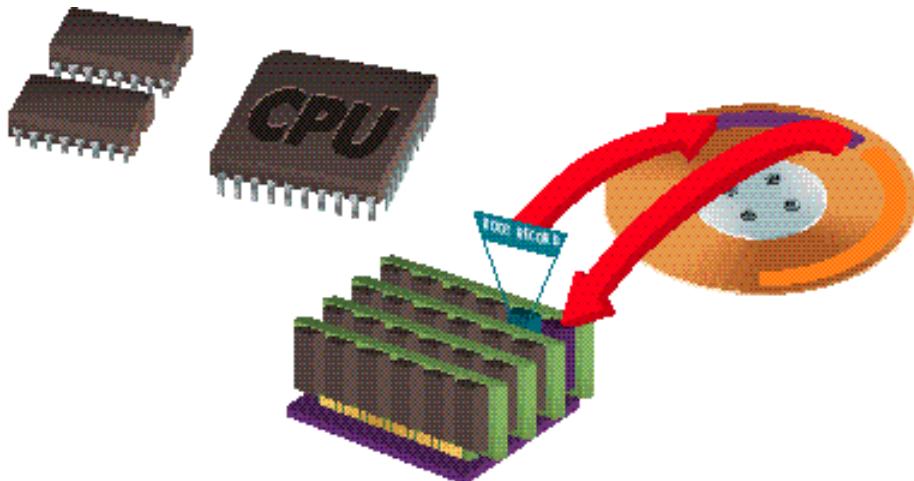
# Disk Boot



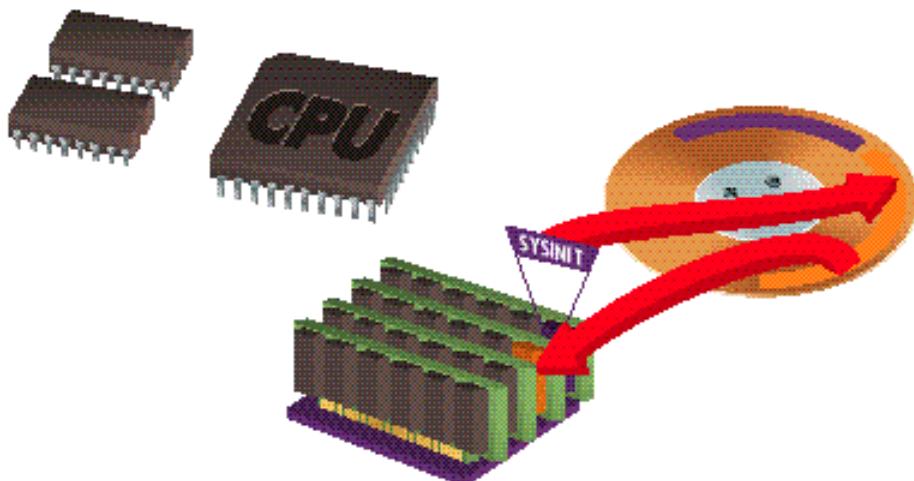
- 1 After conducting a POST check of all the hardware components of a PC, the boot program contained on the computer's ROM BIOS chips checks drive A to see if it contains a formatted floppy disk. If a disk is mounted in the drive, the program searches specific locations on the disk for the files that make up the first two parts of the operating system. You won't ordinarily see these system files because each is marked with a special file attribute that ordinarily hides it from any file listing. For Windows systems, the files are named IO.SYS and MSDOS.SYS. If the floppy drive is empty, the boot program checks the hard drive C for the system files, and on some systems, as a last resort, checks the CD-ROM drive. If a boot disk does not contain the files, the boot program generates an error message.



- 2 After locating a disk with the system files, the boot program reads the data stored on the disk's first sector and copies that data to specific locations in RAM. This information constitutes the boot record. The boot record is found in the same location on every formatted disk. The boot record is only about 512 bytes, just enough code to initiate the loading of the two hidden system files. After the BIOS boot program has loaded the boot record into memory at the hexadecimal address 7C00, the BIOS passes control to the boot record by branching to that address.



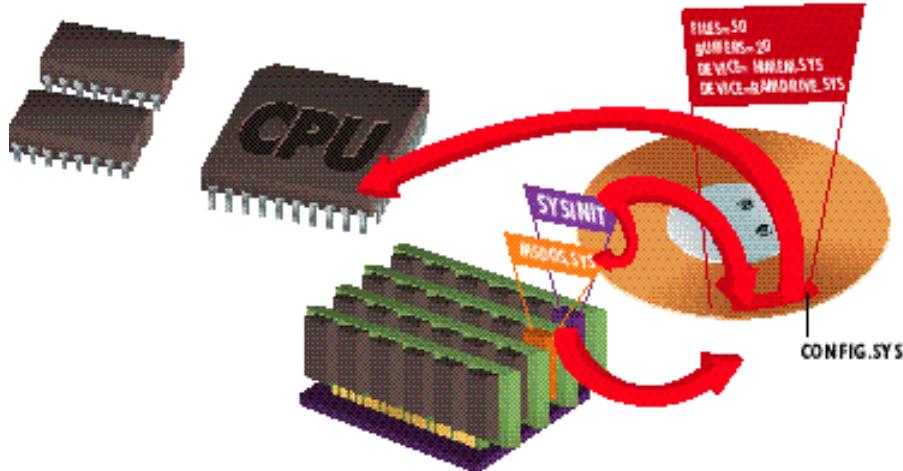
- 3 The boot record takes control of the PC and loads IO.SYS into RAM. The IO.SYS file contains extensions to the ROM BIOS and includes a routine called SYSINIT that manages the rest of the boot up. After loading IO.SYS, the boot record is no longer needed and is replaced in RAM by other code.



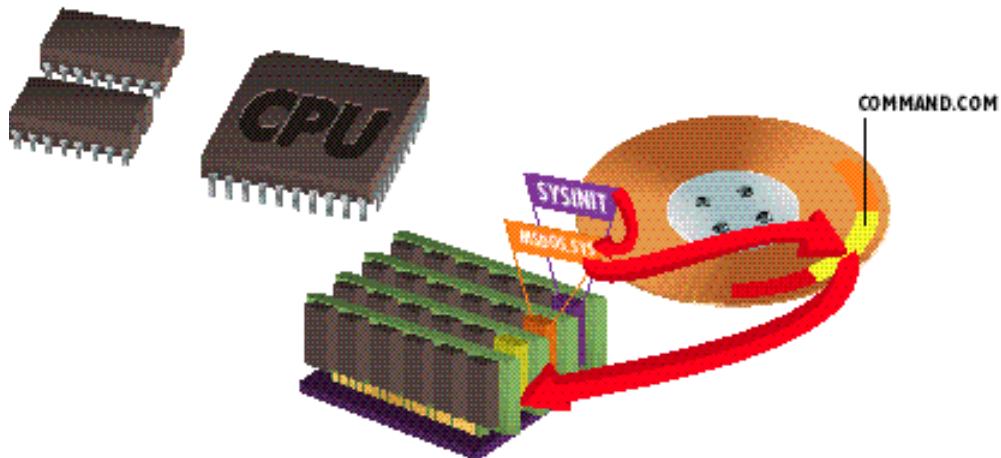
- 4 SYSINIT assumes control of the start-up process and loads MSDOS.SYS into RAM. The MSDOS.SYS file works with the BIOS to manage files, execute programs, and respond to signals from hardware.

[Continued on next page.]

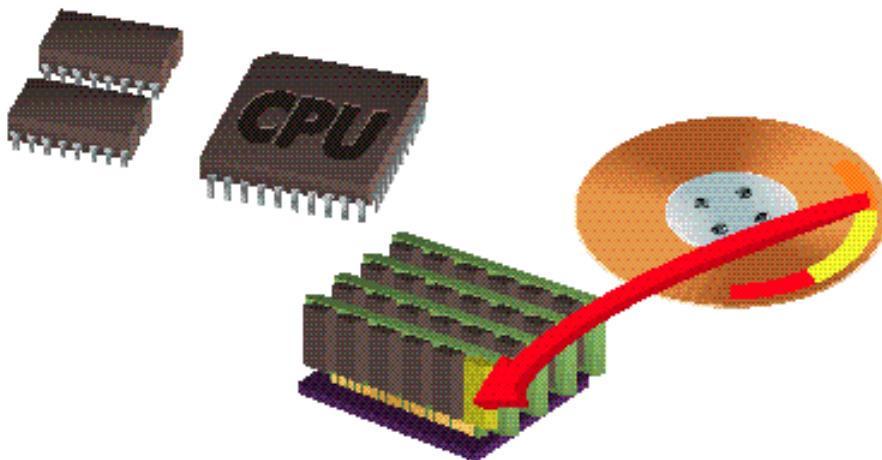
# Disk Boot (Continued)



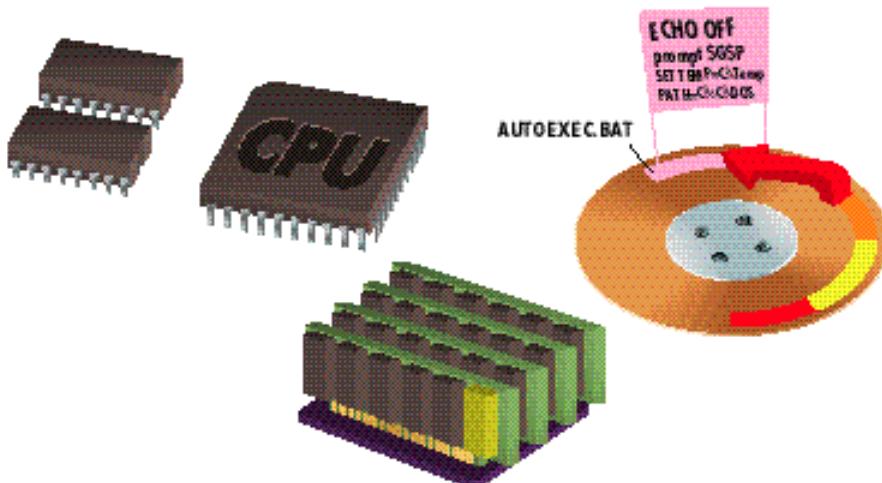
- 5 SYSINIT searches the root directory of the boot disk for a file named CONFIG.SYS. If CONFIG.SYS exists, SYSINIT tells MSDOS.SYS to execute the commands in the file. CONFIG.SYS is a file created by the user. Its commands tell the operating system how to handle certain operations, such as how many files may be opened at one time. CONFIG.SYS may also contain instructions to load device drivers. Device drivers are files containing code that extends the capabilities of the BIOS to control memory or hardware devices. (In Windows, drivers are loaded through records in a file called the Registry.)



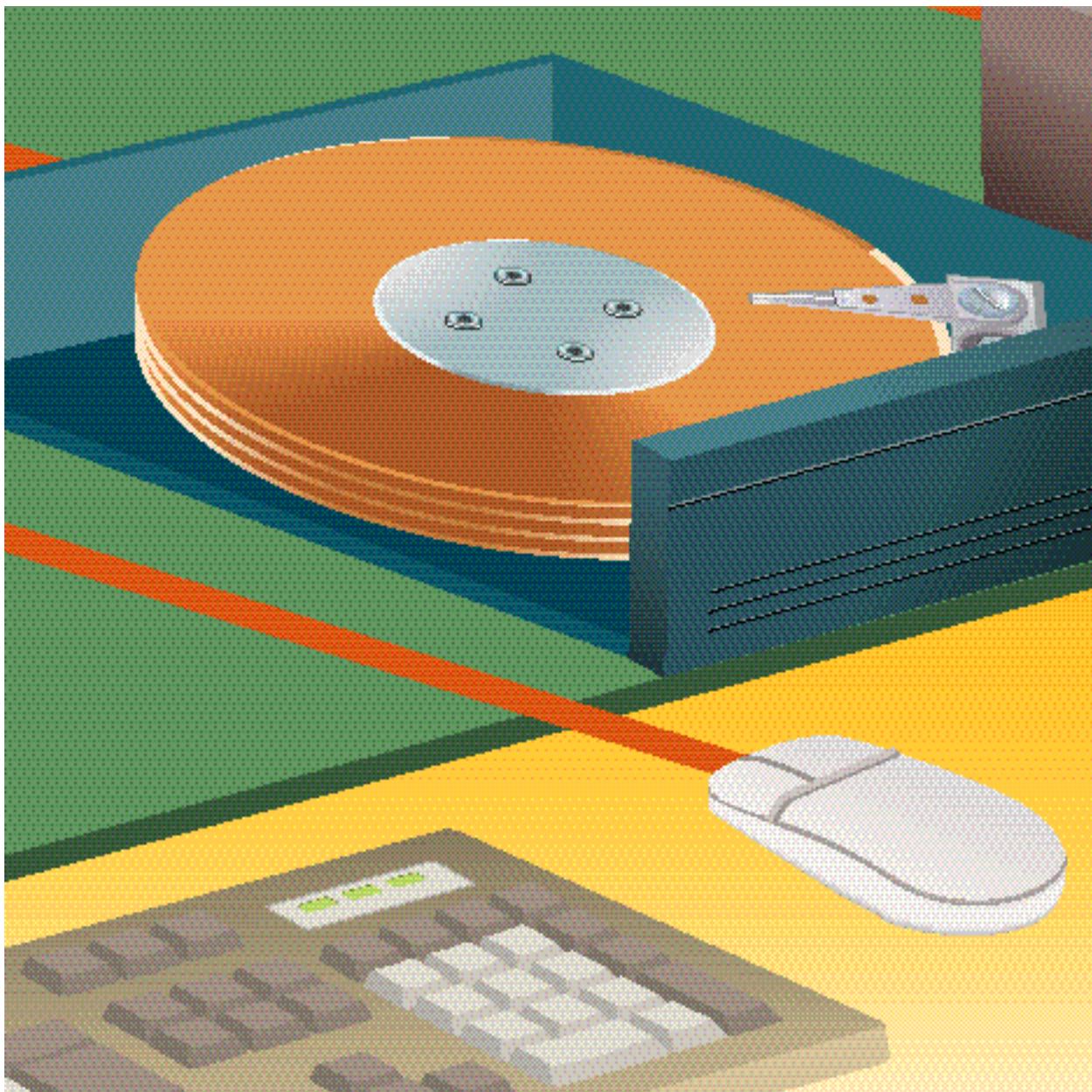
- 6 SYSINIT tells MSDOS.SYS to load the file COMMAND.COM. This operating system file consists of three parts. One is a further extension to the input/output functions. This part is loaded in memory with the BIOS and becomes a part of the operating system.



- 7 The second part of COMMAND.COM contains the internal DOS commands such as DIR, COPY, and TYPE. It is loaded at the high end of conventional RAM, where it can be overwritten by applications programs if they need the memory.



- 8 The third part of COMMAND.COM is used only once and then discarded. This part searches the root directory for a file named AUTOEXEC.BAT. This file is created by the computer's user and contains a series of DOS batch file commands and/or the names of programs that the user wants to run each time the computer is turned on. The PC is now fully booted and ready to be used.



P A R T



# HOW HARDWARE AND WINDOWS 98 WORK TOGETHER

Chapter 3: How Windows 98 Controls Hardware

18

Chapter 4: How the BIOS and Drivers Work

22

Chapter 5: How Plug and Play Works

26

Chapter 6: How Windows 98 Uses Memory

30

**COMPUTERS** are powerful electronic beasts, but they are also terribly ignorant. They know little of the outside world. By itself, your PC doesn't know how to put images on the screen, how to print a document, or even what to do with the keystrokes you peck into it.

There's a good reason for this. To create one piece of hardware that could automatically work with any other piece of hardware would require filling it with information about not only how to communicate with any other hardware that already exists, but also with any hardware someone might come up with in the future. It can't be done.

That's where software comes in. You may already be accustomed to the idea of software as the applications, tools, and games you run on your computer. But specialized software is also needed to tell the PC how to harness its awesome power, how to use new and changing equipment that may be coupled to your PC, and how to make sense of the millions of electrical signals coming in from dozens of different sources.



If you look at the PC as muscle and the software as brains, there is one more element left. A specific chip set called the BIOS is the soul of your PC. The instructions written in the chips' memory to create a basic input/output system are what make your PC an IBM, a Gateway, a Dell, a Compaq, a Hewlett-Packard, or a Packard Bell. Often called firmware to signal its status halfway between hardware and software, the BIOS is the bridge between the rest of the hardware and the operating system. It defines what those electrical signals mean when they arrive at the PC, and it translates the equally baffling signals from the computer into instructions for software and other hardware connected to the computer.

It is this marriage of software and hardware that determines how healthy your PC is. In the early days of the IBM PC, manufacturers of imitators of the IBM machine had to re-create the functions of the IBM BIOS without outright copying the instructions branded into the BIOS chips. And until clone makers got the hang of it, there were a lot of "IBM-compatible" PCs that couldn't display the graphics an IBM could, interpreted keystrokes incorrectly, and sometimes were so confused by the input and output going on that they went into a state of PC catatonia. They crashed.

Today, the relationships between hardware, software, and firmware have been so standardized and purified that compatibility is no longer an issue. But the BIOS and operating system remain the glue between the hard and soft sides of the PC.

## KEY CONCEPTS

**bit, 16-bit, 32-bit** A bit is the smallest chunk of information a computer can work with—either the binary 0 or 1. 16-bit and 32-bit identify processors that can handle binary numbers up to 16 or 32 bits long. The more bits a processor can use, the faster it can make computations and the more memory it can access easily.

**conflict, resource conflict** A problem that occurs when two or more hardware components try to use the same PC resources for memory, interrupts, or direct memory access.

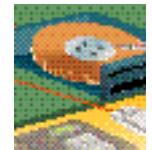
**direct memory access (DMA)** This system recourse allots a section of memory for exclusive use by a component without having to go through the processor.

**driver, device driver** Code written for specific peripherals, such as a video card or a printer, that translates commands from software into the signals the peripheral can recognize.

**expansion cards** Circuit boards designed for specific functions, such as handling sound or video, that plug into the PC's motherboard.

**IBM-compatible** Originally, a PC designed to work with software and hardware the same way an IBM PC would. This meant creating a BIOS that duplicated the functions of the IBM BIOS without violating its copyright. Today the term is archaic because IBM no longer sets the standard for PCs. Instead, PCs conform to standards set by the use of Windows operating systems and Intel processors (Wintel).

**interrupt controller** A microchip that notifies the CPU that some piece of hardware has sent a signal, in the form of a number, that it needs the processor to perform some task.



**interrupt table** A record of memory addresses linked to a specific interrupt number. When an interrupt controller receives an interrupt, it looks up the address associated with that number and instructs the processor to put onto a stack the current address of the data the processors work with and to start executing the code contained at the address found in the table.

**interrupt (IRQ), or interrupt request** A signal from a peripheral requesting some service by the processor.

**interrupt return, or IRET instruction** A signal the processor sends to the interrupt controller telling it that the CPU has completed the task requested by an interrupt.

**memory** Where a computer stores software and the data the software manipulates. Most commonly used to refer to RAM, but also includes data storage on hard drives and other devices.

**memory, extended** That area of RAM that includes all memory addresses above 1 megabyte.

**memory, low** RAM with memory addresses from 0 to 640K.

**memory, upper** RAM with memory addresses from 640K to 1MB.

**memory, virtual** Hard drive space that the operating system tells the processor to treat as if it were RAM.

**memory address** The specific location in RAM or virtual memory that can mark the beginning of a section of code or data.

**motherboard** The main circuit board of a PC, into which the processor and expansion cards are inserted.

**peripheral** Any component, inside or external, that adds new hardware capabilities to the basic design of a computer. For example: hard drives, printers, mouse.

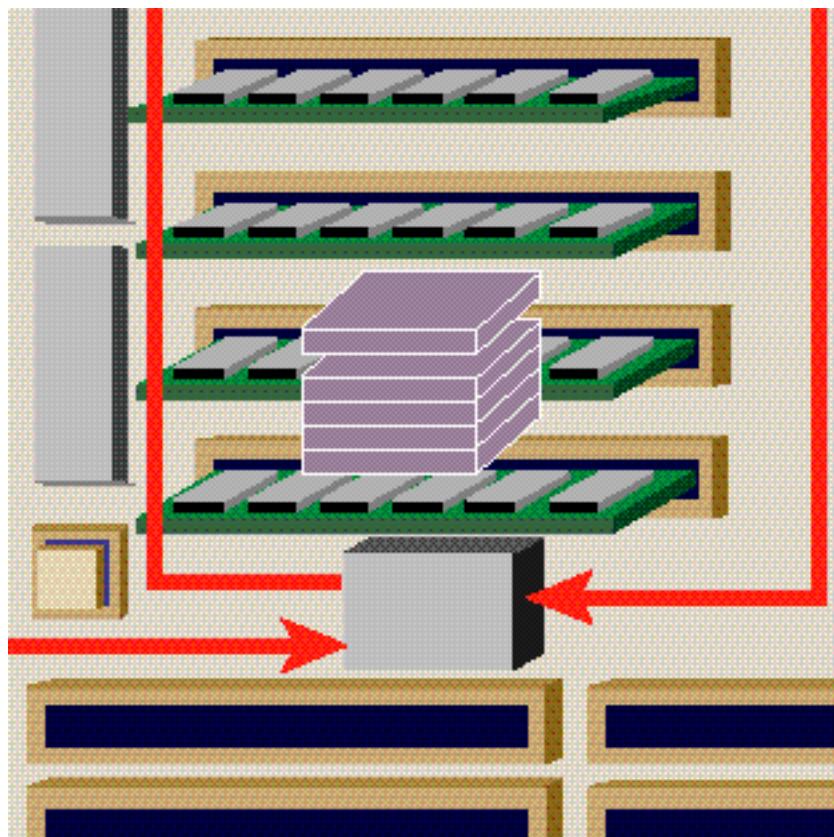
**Plug and Play** A hardware and software design that is supposed to automatically configure system resource settings.

**processor** The main microchip in a PC. It carries out the instructions of software by using other components, such as RAM or disk drives, as needed.

CHAPTER

## 3

# How Windows 98 Controls Hardware



**OPERATING** systems were originally developed to handle one of the most complex input/output operations: communicating with a variety of disk drives. This is evidenced by the names given to early operating systems, which often contained the acronym DOS for disk operating system. Eventually, the operating system quickly evolved into an all-encompassing bridge between your PC and the software you run on it.

Without an operating system, such as Windows 98, each programmer would have to invent from scratch the way a program displays text or graphics onscreen, how it sends data to the printer, how it reads or writes disk files, and how it implements other functions that mesh software with hardware. An operating system, however, is more than a way to make life easier for programmers.

An operating system creates a common platform for all the software you use. Without an operating system, you might not be able to save files created by two different programs to the same disk because each might have its own storage format. An operating system also gives you a tool for all the tasks you want to perform outside of an application program: deleting and copying files to disk, printing, and running a collection of commands in a batch file.

The operating system does not work alone. It depends not only on the cooperation of other programs, but also on meshing smoothly with the BIOS. The BIOS—or basic input/output system—is made of code contained on chips in a PC. It acts as the intermediary among the hardware, processor, and operating system. Device drivers are like a specialized BIOS. Drivers translate commands from the operating system and BIOS into instructions for a specific piece of hardware, such as a printer, scanner, or CD-ROM drive. When some parts of the operating system are loaded from disk, they are added to the BIOS and then joined by device drivers, and all of them carry out routine hardware functions. The operating system is really composed of all three of these components. It is simplistic to think of an operating system as being only the files contained on your Windows 98 CD-ROM.

Together, the BIOS, device drivers, and Windows 98 perform so many functions that it's impossible to depict their complexity with a couple of pages of illustrations. Here we'll show how the operating system works with BIOS to perform a simple task.

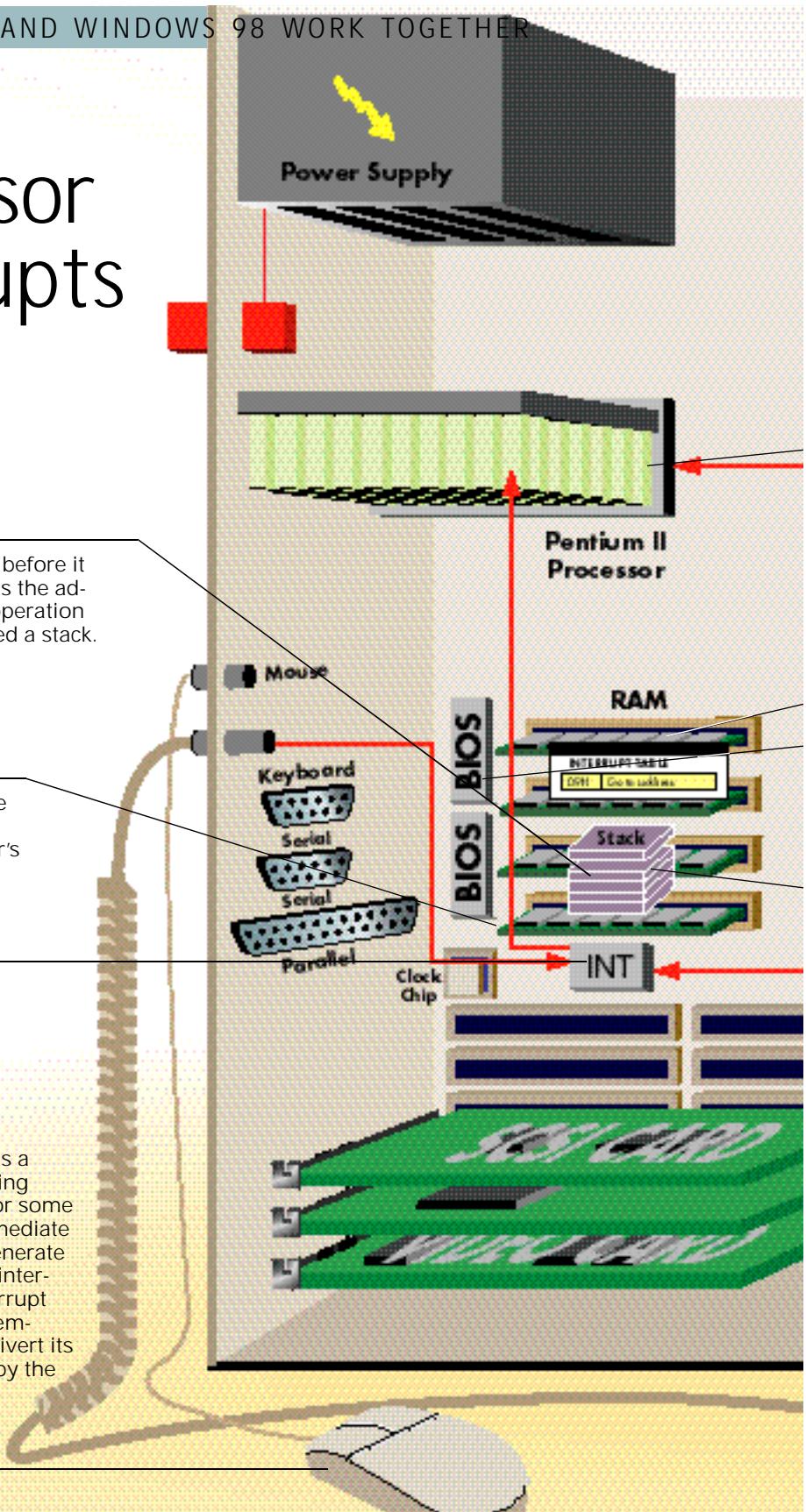
# The Processor and Interrupts

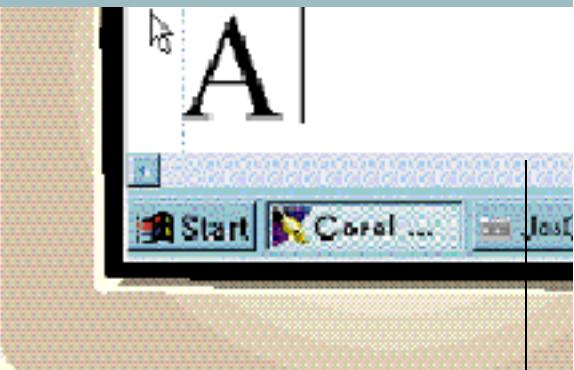
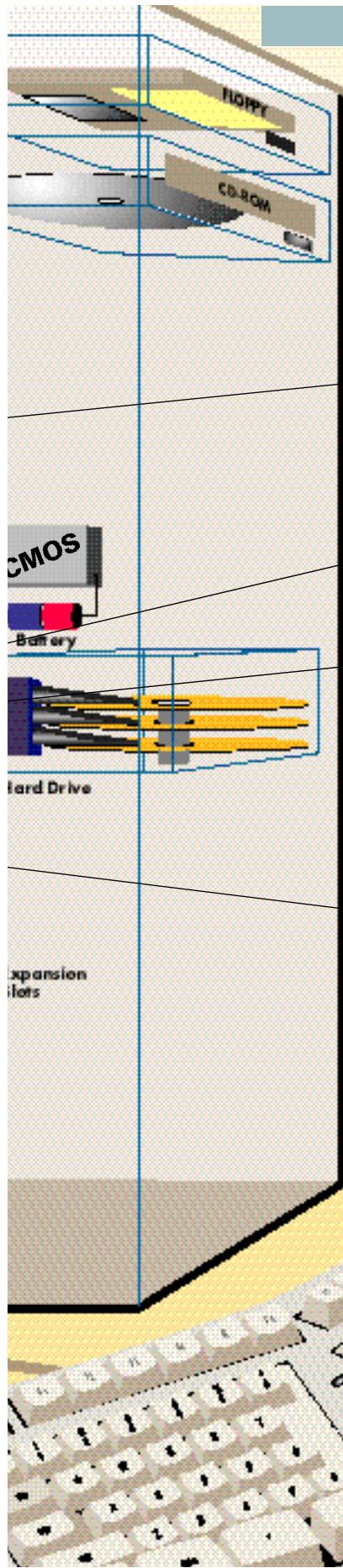
- 4 To keep track of what it was doing before it was interrupted, the processor puts the address of the application's current operation into a special location in RAM called a stack.

- 3 The interrupt controller notifies the processor that an interrupt has occurred and demands the processor's immediate attention.

- 2 A special chip called the interrupt controller receives the interrupt signal.

1 Various hardware events—such as a keystroke, mouse click, data coming through a serial or parallel port, or some software events that need an immediate response from the processor—generate a special type of signal called an interrupt. As its name implies, an interrupt causes the operating system to temporarily stop what it is doing to divert its attention to the service required by the signal.



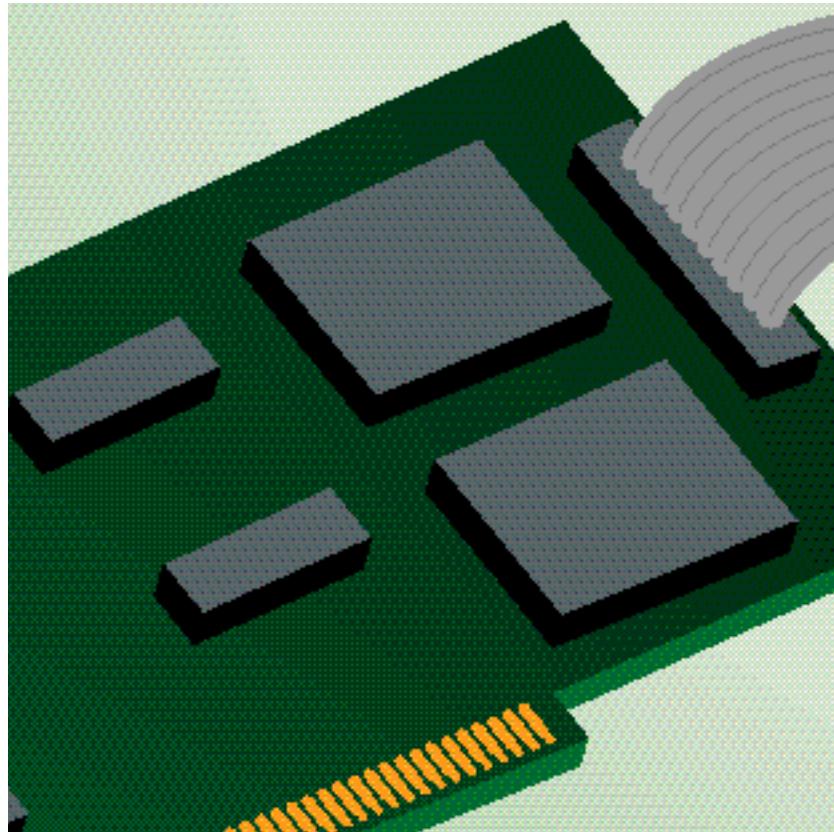


- 5 The processor retrieves the number of the interrupt from the interrupt controller. Each interrupt is associated with a particular number and, in some cases, a second number denoting a particular service—a specific hardware function that is being requested by the interrupt.
- 6 The processor looks in the interrupt table, the section of RAM used to hold interrupt vectors to find the memory address associated with the particular interrupt.
- 7 The processor reads in the instructions that begin at the address it found in the interrupt table. In this example, the memory address is in the range occupied by the system's primary BIOS code.
- 8 Executing the BIOS instructions causes the processor to retrieve a special code that represents a specific keystroke, to give the keystroke to the application program, and to display the character on the screen.
- 9 If the BIOS routine is completed successfully, the BIOS generates an interrupt return, or IRET instruction. The IRET tells the processor to retrieve the address it had placed on the stack, so it can continue to execute those instructions where it left off.

CHAPTER

# 4

# How the BIOS and Drivers Work



**IMAGINE** sitting down in the driver's seat of a car only to discover that the steering wheel has been replaced by an airplane joystick. Luckily, you don't ever have to face that problem. Despite how car mechanisms may differ, you only need to know how to turn the steering wheel and press the gas and brake pedals. An intervening layer—the levers, gears, and hydraulic systems—separates you from the parts that do the real work. These mechanisms translate your actions into car movements.

In the same way, your personal computer, too, has a layer that separates you and your application software from the down-and-dirty workings of your hardware. In fact, there are three layers: the operating system; an even more fundamental layer that lies between the operating system and the hardware called the BIOS, or basic input/output system; and device drivers, which supplement the BIOS's knowledge of how to operate peripherals.

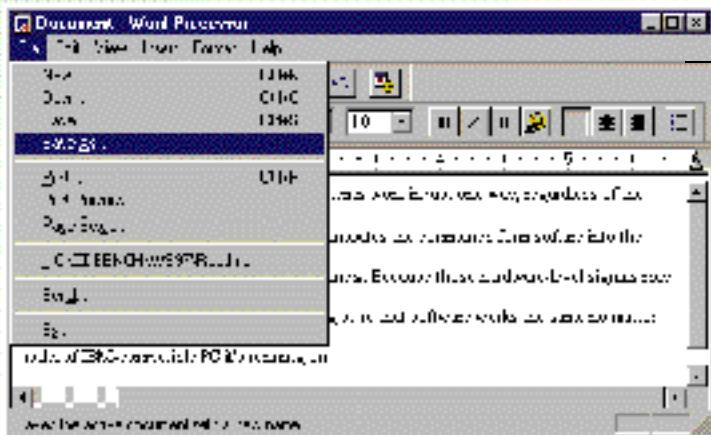
A good way to think of the operating system, BIOS, drivers, and hardware is in terms of the human body. The operating system is the brain—more exactly, the cerebrum. It directs all the voluntary, “conscious” operations of the body hardware. The BIOS is the medulla oblongata, which controls involuntary body functions so that the cerebrum doesn't have to think about them. As for device drivers, imagine that you've received an artificial limb. Neither the cerebrum nor the medulla know exactly how to control that limb, but it comes with its own knowledge of how to react to certain signals from the brain. The brain can send neuron impulses that say “move up” and the limb translates those impulses into the electrical currents and motor movements needed to move the limb up. The BIOS and drivers let your applications and various operating systems get the same results, regardless of the type of IBM-compatible computer they're running on and regardless of the type of printer, hard drive, or any other peripheral your PC is working with.

The BIOS handles the grunt work of sending and recognizing the correct electrical signals associated with various hardware components. Instead of having to know how a certain driver works, an application program needs to know only how to communicate through the operating system to the BIOS. It's like having a highly efficient assistant who knows how the office filing system works. All you have to do is leave a file in the out basket, and the assistant makes sure it gets into the right drawer in the right cabinet.

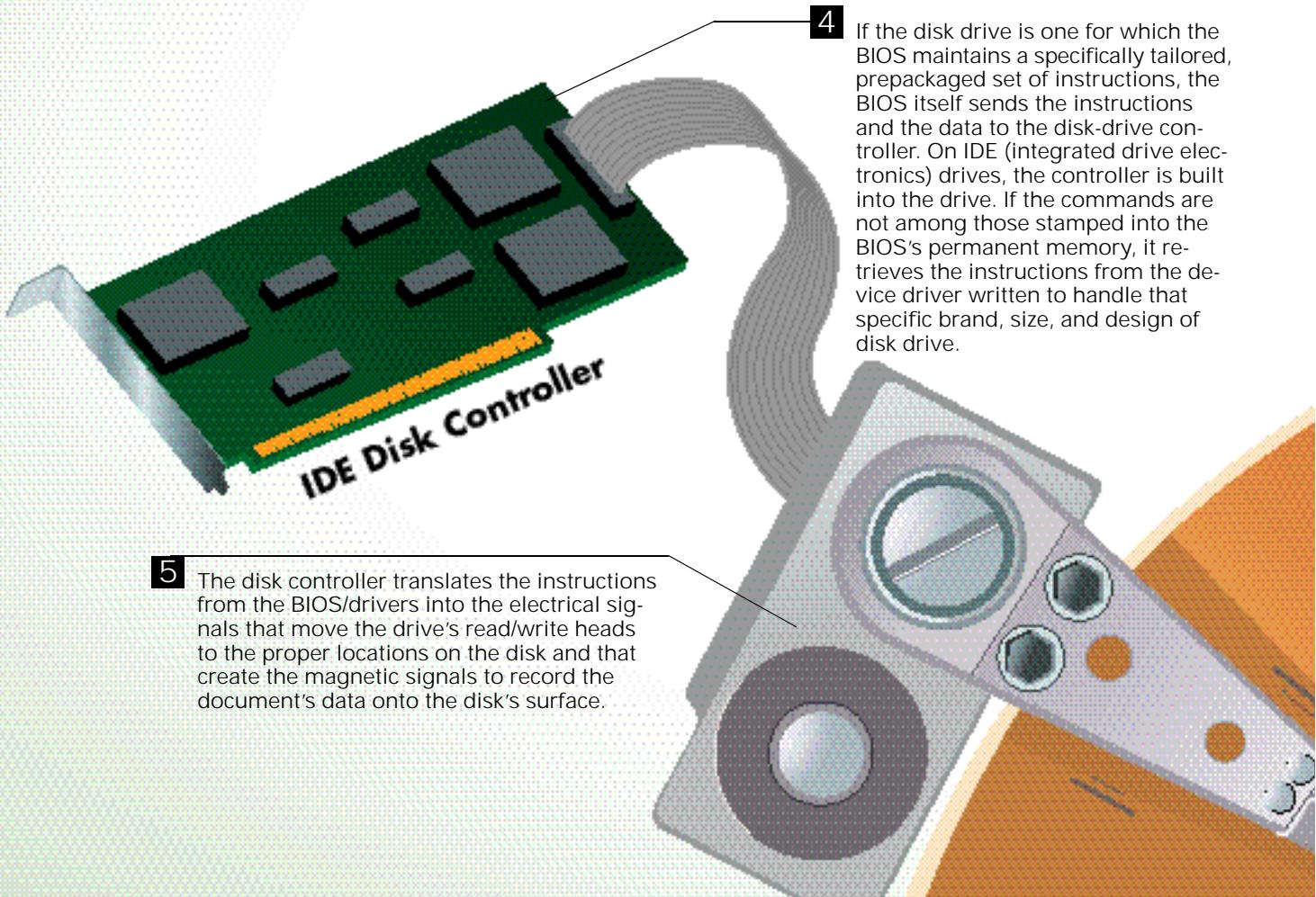
The working part of the BIOS is the code located in one or more read-only memory (ROM) chips—a type of memory that cannot be modified. These chips normally are located on your PC's main circuit board, or motherboard. It's one part of your PC that, ordinarily, you don't replace.

Except for a fleeting message that appears onscreen as you boot your computer and it loads the BIOS code, you will rarely be aware of your PC's BIOS codes or drivers and what they're doing, which is exactly the way it should be. A good BIOS is a vital but silent partner.

# How the BIOS and Drivers Work

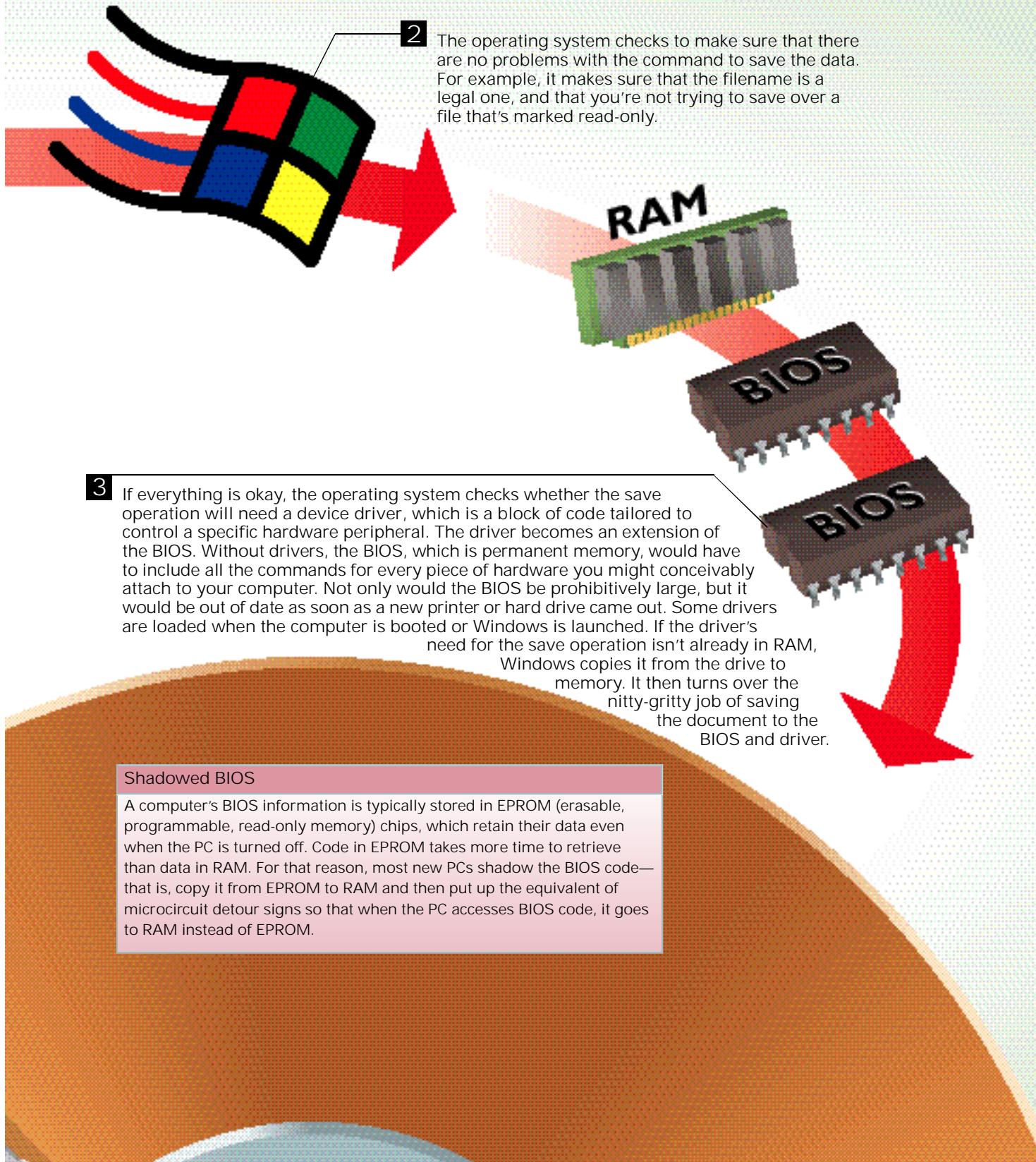


1 When you choose the command, for example, to have your word processor save a document, the application doesn't need to know how to control the hard drive. Instead, the word processor sends the command and the data to be saved to Windows 98.



5 The disk controller translates the instructions from the BIOS/drivers into the electrical signals that move the drive's read/write heads to the proper locations on the disk and that create the magnetic signals to record the document's data onto the disk's surface.

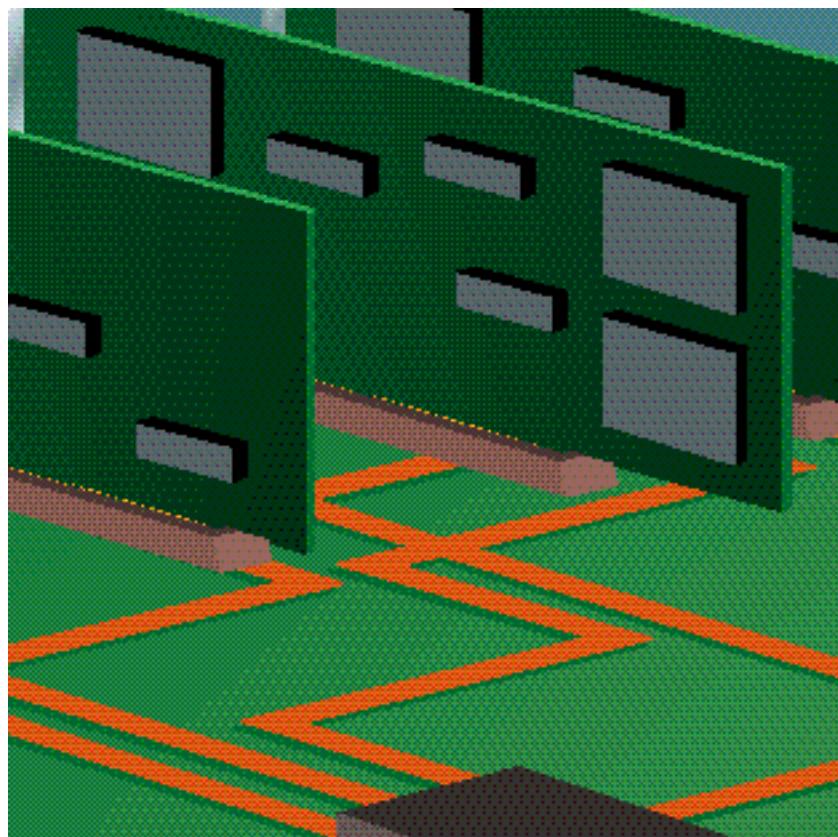
4 If the disk drive is one for which the BIOS maintains a specifically tailored, prepackaged set of instructions, the BIOS itself sends the instructions and the data to the disk-drive controller. On IDE (integrated drive electronics) drives, the controller is built into the drive. If the commands are not among those stamped into the BIOS's permanent memory, it retrieves the instructions from the device driver written to handle that specific brand, size, and design of disk drive.



CHAPTER

# 5

# How Plug and Play Works



**UNTIL** recently, when you bought a new expansion card for your computer, it was a nightmare trying to ensure that it got along peacefully with the other components already in your system. Problems arose because each component needed to communicate with the processor and other peripherals, and there were only a few channels for that communication. These channels are usually referred to as **system resources**. One resource is an interrupt, or IRQ. Another system resource is a direct line to memory called a DMA (direct memory access).

As its name suggests, an interrupt makes the processor interrupt whatever it's doing to take a look at the request a component is making for the processor's time. If two devices use the same interrupt, the processor can't tell which device is asking for its attention. If two devices use the same DMA, one device will overwrite data the other has already stored in memory. When anything like this happens, it's called a conflict, and it's not a pretty sight.

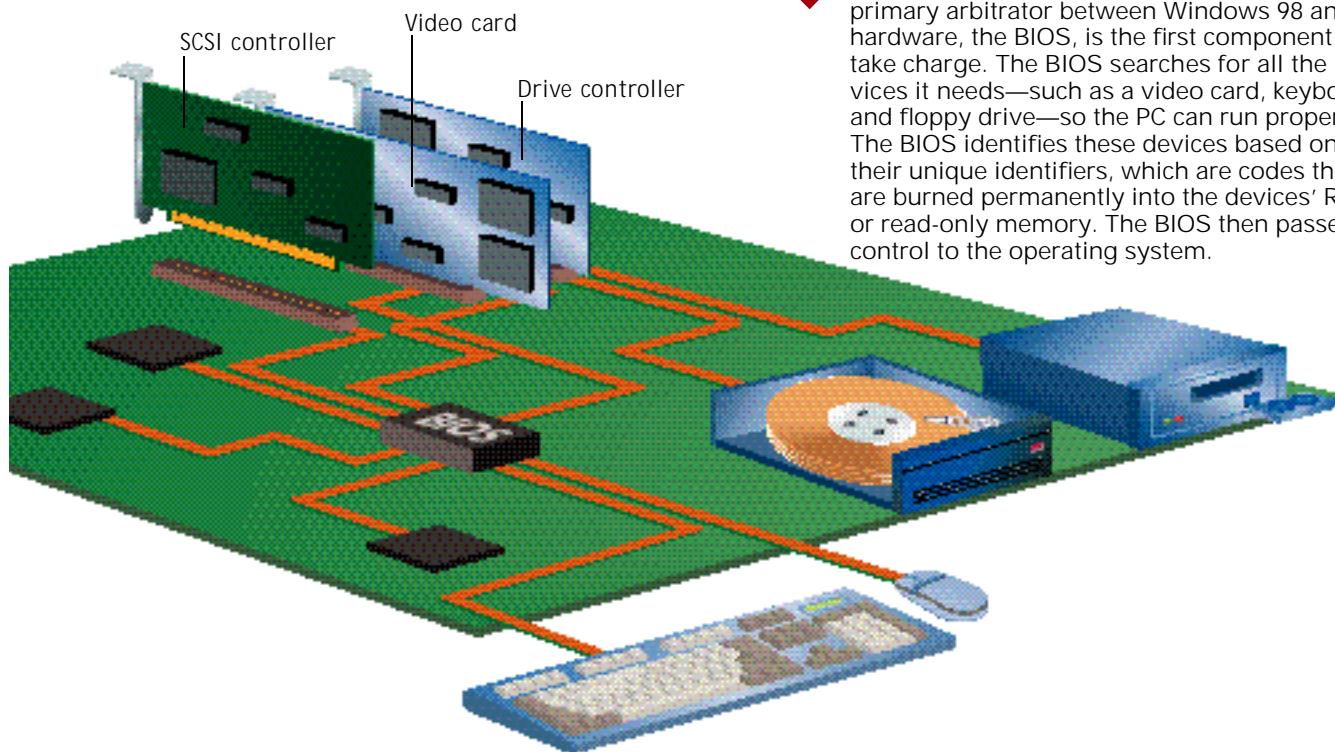
In the dark ages of PCs—the decade of the eighties and half the nineties—there were two solutions to avoid conflicts. One way was to be so anal-retentive that you had a complete record of every resource used by every device in your PC. Of course, no one had such records. So most people resolved conflicts by plugging in a new expansion card or drive, seeing if everything worked, and when it didn't—as was often the case the first time—folks pulled out the new device and started over. That involved changing some nearly microscopic switches to change the resources the device used, plugging the device back in, rebooting, seeing if everything worked, and so on until finally you stumbled upon a combination that appeared to work.

There had to be a better way, and now there is. Most PC companies, including the influential Microsoft and Intel, agreed to a system called (optimistically) Plug and Play. In theory, if every device in your PC conforms to the Plug and Play standard, the PC's BIOS, Windows, and the devices themselves work together to automatically make sure no two of them compete for the same resources. Not every component uses Plug and Play, however. Look for it when buying components.

Before Plug and Play, if you wanted to add a piece of hardware to your system, you had to turn off the PC before installing the component. But Plug and Play lets you change devices on-the-fly without turning off the system—a process called hot swapping. It's most likely to crop up on notebooks or other PCs that use PCMCIA cards (PC cards).

The catch is that your PC, its BIOS, the peripherals, and your operating system all have to support Plug and Play. With the laissez-faire attitude many PC and component makers take toward standards, Plug and Play is not perfect. Windows 98 supplies many Plug and Play software drivers that other companies can use, but there's no way to make component manufacturers conform to the Plug and Play standard. Still, it's a big step toward hassle-free upgrading.

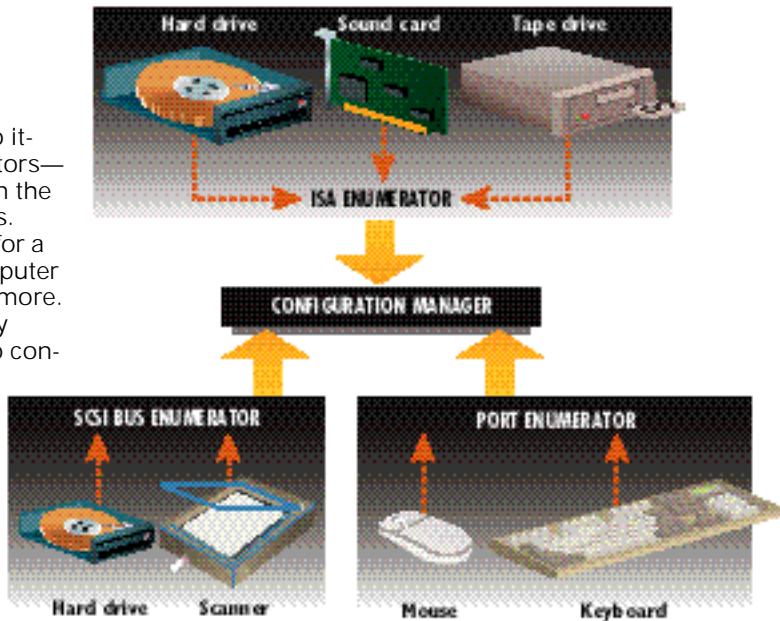
# How Plug and Play Works



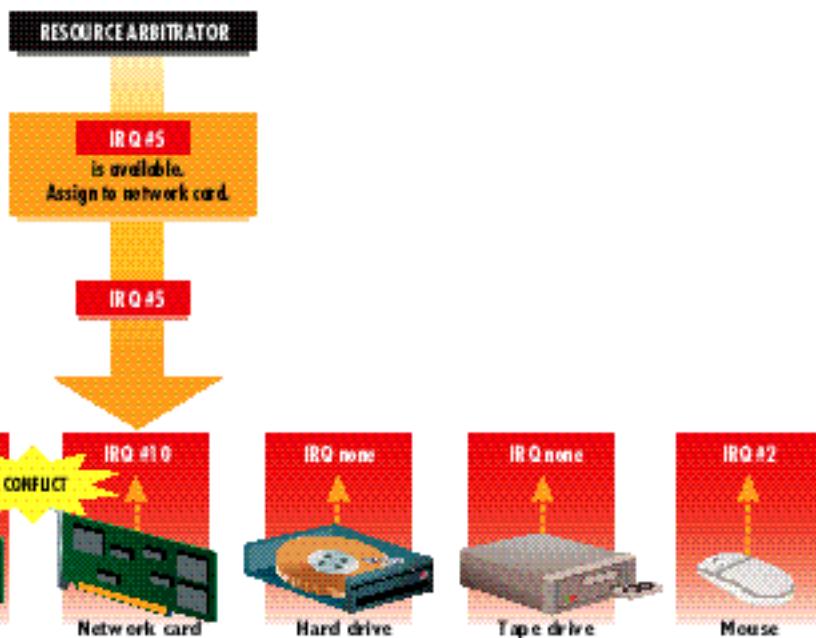
1

When you turn on a Plug and Play system, the primary arbitrator between Windows 98 and hardware, the BIOS, is the first component to take charge. The BIOS searches for all the devices it needs—such as a video card, keyboard, and floppy drive—so the PC can run properly. The BIOS identifies these devices based on their unique identifiers, which are codes that are burned permanently into the devices' ROM, or read-only memory. The BIOS then passes control to the operating system.

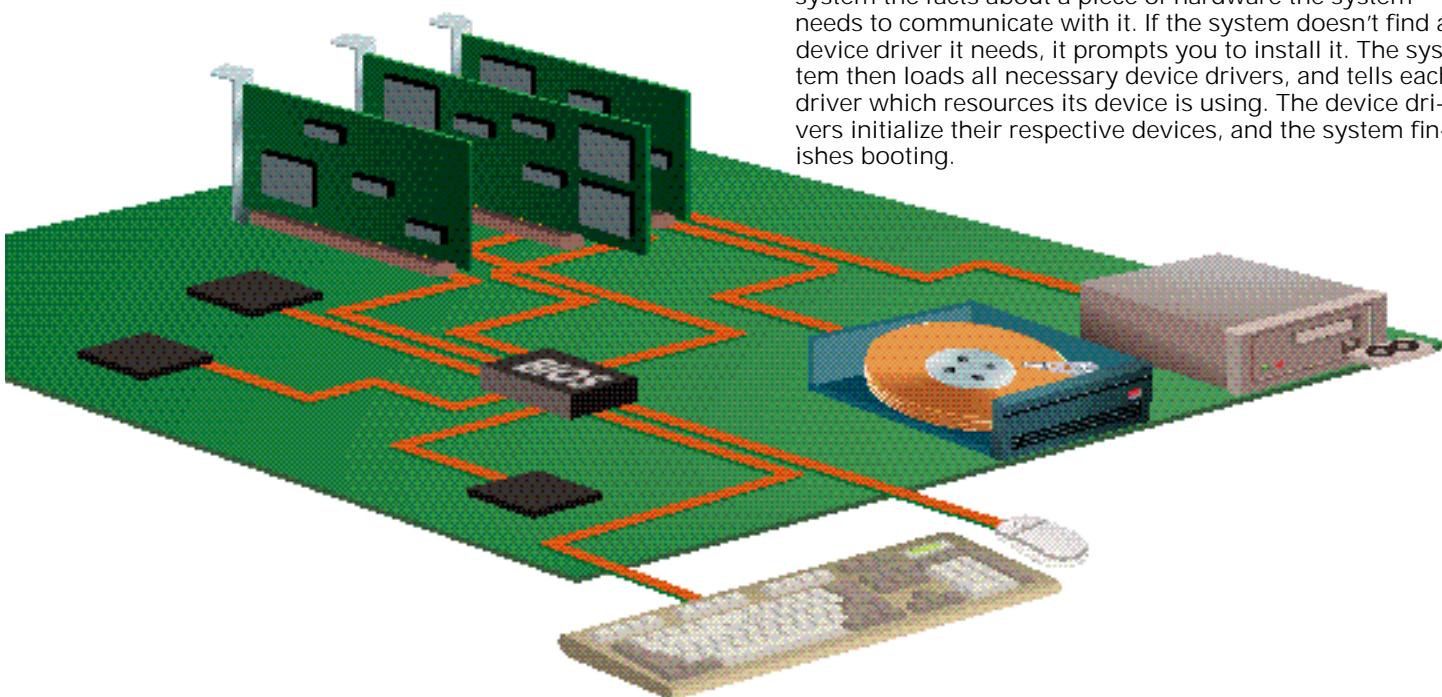
- 2** Windows's configuration manager adds to itself special device drivers called enumerators—programs that act as the interface between the operating system and the different devices. There are bus enumerators, enumerators for a special type of bus called SCSI (small computer system interface), port enumerators, and more. Windows asks each enumerator to identify which devices the enumerator is going to control and what resources it needs.



- 3 Windows 98 takes the information from the enumerators and stores it in the hardware tree, which is a database stored in RAM. The operating system then examines the hardware tree for resource arbitration. In other words, after storing the information in a database, the operating system decides what resources—interrupts (IRQs), for example—to allocate to each device. The system then tells the enumerators what resources it allocated to their respective devices. The enumerators save the resource allocation information in the peripherals' microscopic programmable registers, which are sort of digital scratch pads located in some chips.



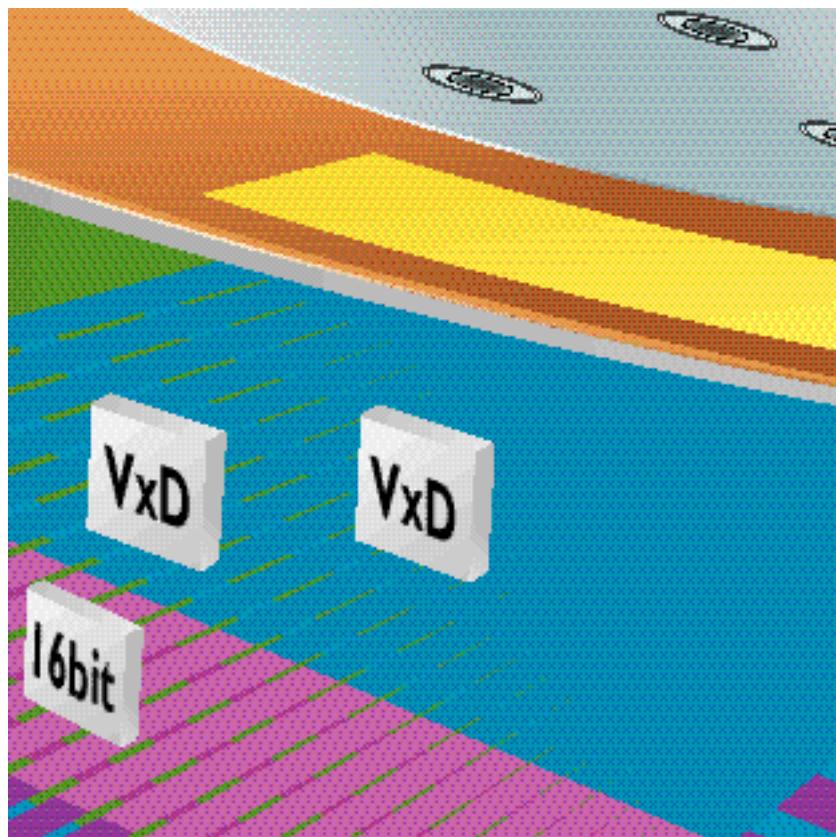
- 4 Finally, the operating system searches for the appropriate device driver for each device. A device driver is a small piece of add-on code for Windows that tells the operating system the facts about a piece of hardware the system needs to communicate with it. If the system doesn't find a device driver it needs, it prompts you to install it. The system then loads all necessary device drivers, and tells each driver which resources its device is using. The device drivers initialize their respective devices, and the system finishes booting.



CHAPTER

# 6

# How Windows 98 Uses Memory



**MEMORY** is the staging area for the processor, the place where the processor receives the instructions and data it needs to do its job. It is Windows's task to organize this staging area in an efficient manner so that the processor can get what it needs, when it needs it. In fact, some of the biggest improvements of Windows 95, Windows 98, and Windows NT over Windows 3.1 and DOS are largely due to the methods Windows uses to organize and manage memory.

Intel's 80386, 80486, and Pentium microprocessors are 32-bit processors, which means that they can manipulate binary numbers as large as 11111111111111111111111111111111, which translates to the decimal number 4,294,967,296, or 4GB (gigabytes). This number represents the upper limit to the amount of memory that the processor—and therefore Windows—can address.

Most of this addressable memory is contained in the physical RAM chips in your PC. However, Windows can also make use of virtual memory—hard disk space that acts as an extension of RAM. The memory management features of Windows use real and virtual memory to let you open multiple applications at the same time, even if there isn't enough physical memory to open them all. With the improved memory management of Windows 95 and Windows 98, you can run more applications before getting the out-of-memory message than you could in Windows 3.1.

Windows 95 and 98 use a flat memory scheme that allows them to address directly the full range of memory that 32-bit processors are capable of using. Older versions of Windows, which relied on DOS, used a segmented memory scheme. Instead of accessing memory 32 bits at a time, DOS accesses RAM with two 16-bit segments.

Windows also assigns portions of memory to each application. When you run multiple Windows 3.1 (16-bit) applications in Windows 95, the applications must share a segment of memory in a cooperative manner, as they do under Windows 3.1. If the applications are not well behaved, they can fight over RAM, and cause a system crash. Newer 32-bit applications each run in their own protected memory block, eliminating some disputes over memory and reducing the number of crashes. In fact, in Windows NT, it is difficult to crash the system because every task runs in its own piece of protected memory. This is not so for Windows 95/98.

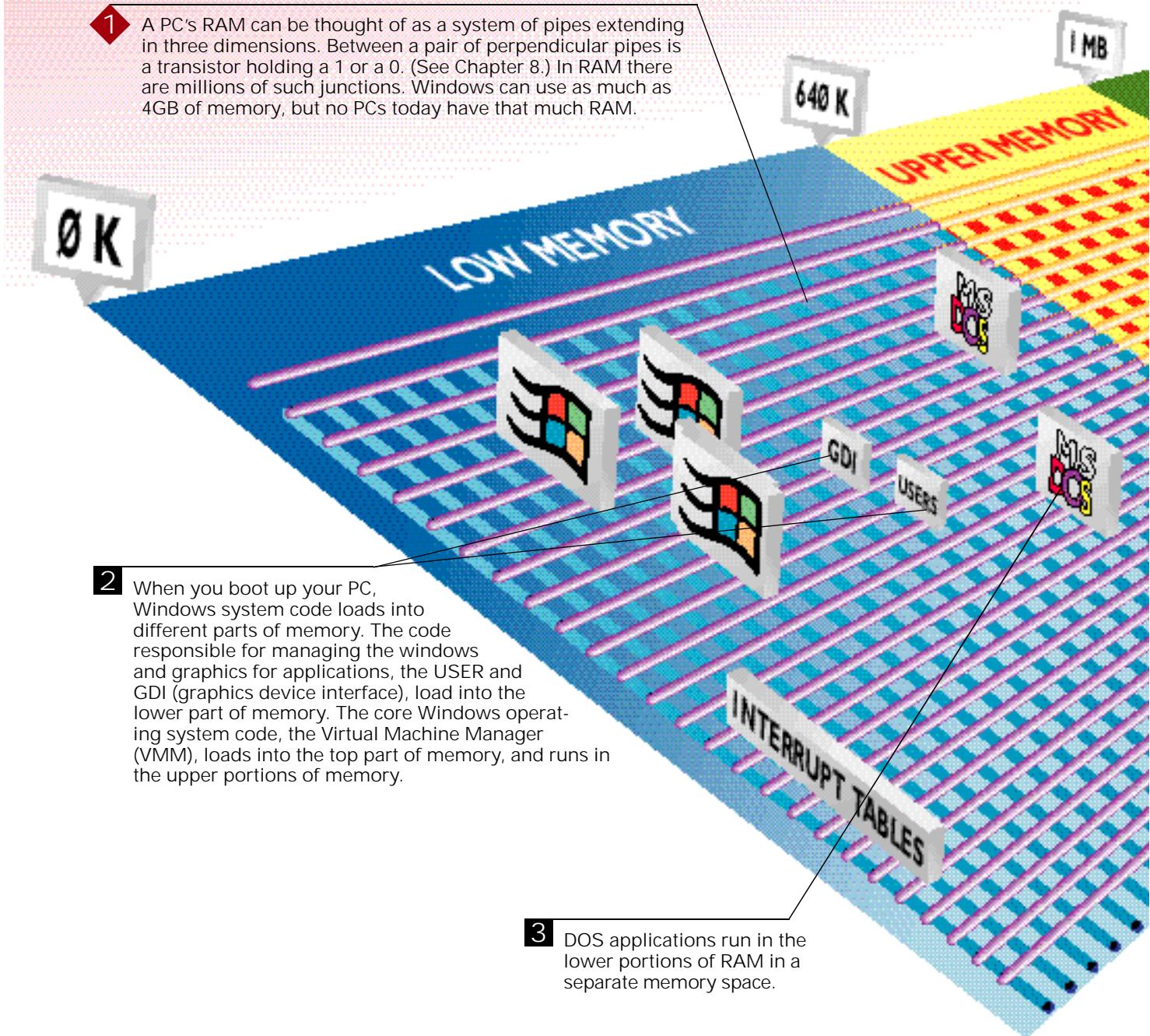
To remain compatible with Windows 3.1 software, Windows 95/98 contains some 16-bit code, which both older 16-bit and newer 32-bit applications must access. This is the windows code that provides window management and graphics services to applications. A problem with this code can bring down the older Windows system. Eventually, the time will come when compatibility with old 16-bit applications won't be important, and Windows 98 can become as robust as Windows NT is today.

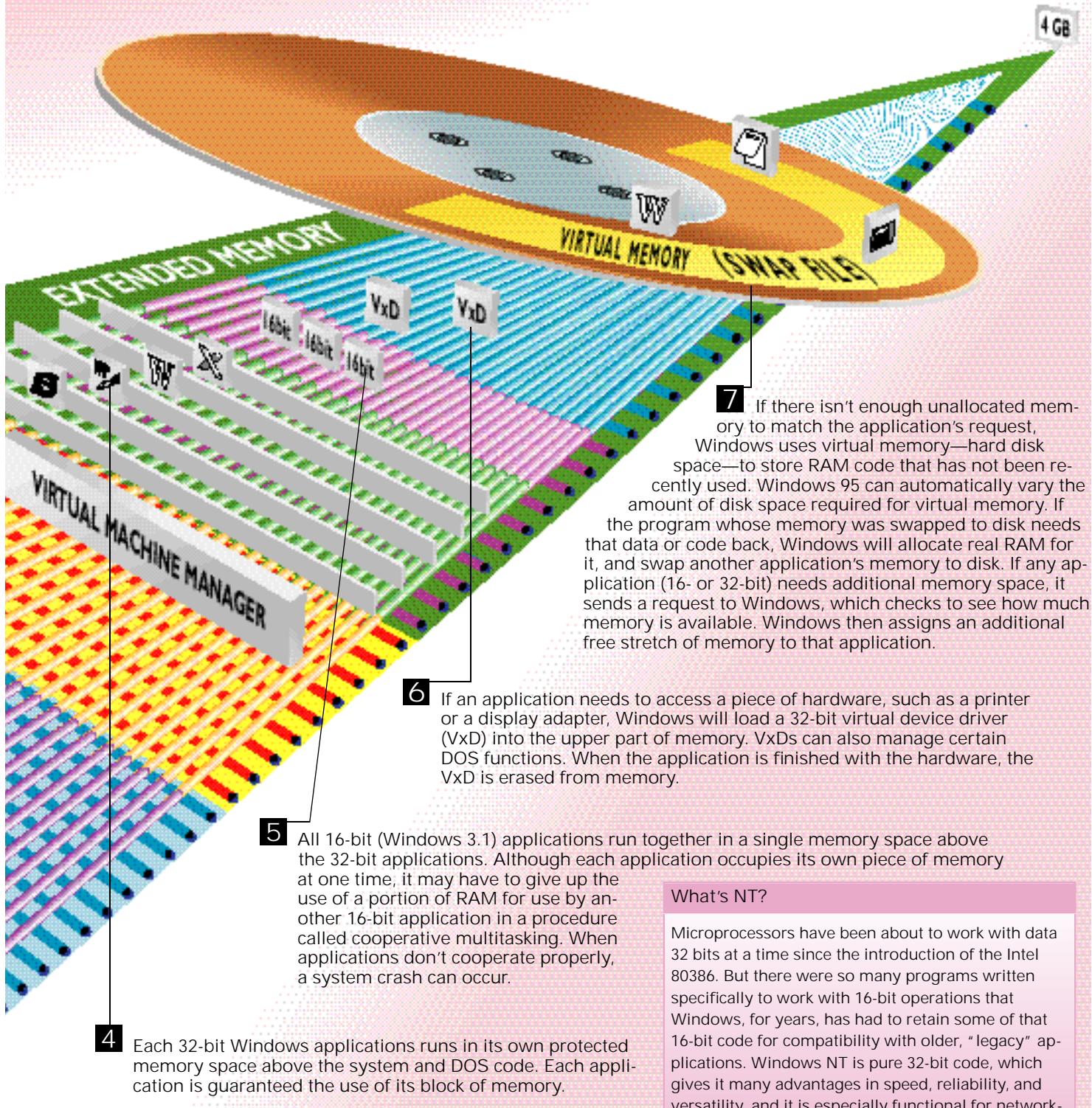
# How Windows Uses Memory

- 1 A PC's RAM can be thought of as a system of pipes extending in three dimensions. Between a pair of perpendicular pipes is a transistor holding a 1 or a 0. (See Chapter 8.) In RAM there are millions of such junctions. Windows can use as much as 4GB of memory, but no PCs today have that much RAM.

- 2 When you boot up your PC, Windows system code loads into different parts of memory. The code responsible for managing the windows and graphics for applications, the USER and GDI (graphics device interface), load into the lower part of memory. The core Windows operating system code, the Virtual Machine Manager (VMM), loads into the top part of memory, and runs in the upper portions of memory.

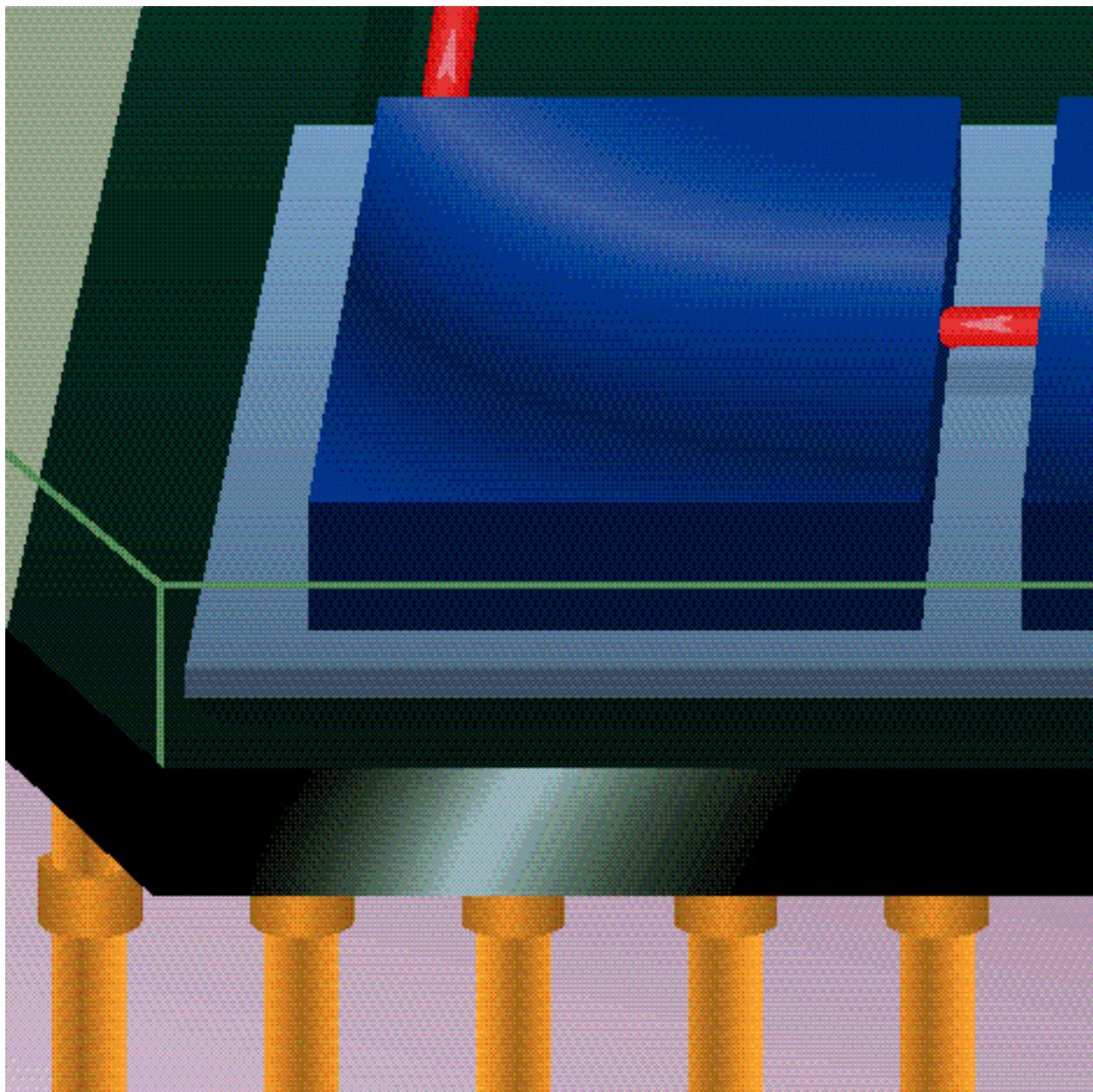
- 3 DOS applications run in the lower portions of RAM in a separate memory space.





#### What's NT?

Microprocessors have been able to work with data 32 bits at a time since the introduction of the Intel 80386. But there were so many programs written specifically to work with 16-bit operations that Windows, for years, has had to retain some of that 16-bit code for compatibility with older, "legacy" applications. Windows NT is pure 32-bit code, which gives it many advantages in speed, reliability, and versatility, and it is especially functional for networking. But it cannot run old, 16-bit Windows programs. The next version of Windows following Windows 98 is expected to similarly eliminate the last vestiges of 16-bit code.



P A R T

# 3

## MICROCHIPS

Chapter 7: How a Transistor Works

38

Chapter 8: How RAM Works

42

Chapter 9: How a Computer Performs Math

50

Chapter 10: How CISC and RISC Processors Work

54

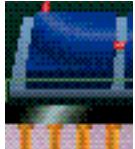
Chapter 11: How a Microprocessor Works

58

**THE** first computers used components called vacuum tubes. If you're not at least old enough to be part of the baby boom generation, you may never have seen more than one type of vacuum tube—the cathode ray tube that displays images on your PC monitor and TV screen. Except for CRTs, vacuum tubes are rarely used in modern electronics.

Vacuum tubes in early computers functioned as electronic switches. When current flowed through one part of the tube, it made that component so hot, electrons boiled off and were attracted to another part of the tube that had a positive charge. A partial vacuum was needed inside the tube so that the electrons would encounter little resistance from molecules in the air. When the electrons were flowing, the switch was on. When they weren't flowing, the switch was off.

Essentially, a computer is just a collection of On/Off switches, which at first doesn't seem very useful. But imagine a large array of lightbulbs—say, ten rows that each has 50 lightbulbs in it. Each bulb is connected to a light switch. If you turn on the right combination of switches, you can put your name in lights.



Computers are similar to that bank of lights, with one important difference: A computer can sense which lightbulbs are on and use that information to turn on other switches. If the pattern of On switches spells Tom, the computer could be programmed to associate the Tom pattern with instructions to turn on another group of switches to spell boy. If the pattern spells Mary, the computer could turn on a different group of switches to spell girl.

The two-pronged concept of On and Off maps perfectly with the binary number system, which uses only 0 and 1 to represent all numbers. By manipulating a roomful of vacuum tubes, early computer engineers could perform binary mathematical calculations, and by assigning alphanumeric characters to certain numbers, they could manipulate text.

The problem with those first computers, however, was that the intense heat generated by the hundreds of vacuum tubes made them notoriously unreliable. The heat caused many components to deteriorate and consumed enormous amounts of power. But for vacuum tubes to be on, the tubes didn't really need to generate the immense flow of electrons that they created. A small flow would do quite nicely, but vacuum tubes were big. They worked on a human scale in which each part could be seen with the naked eye. They were simply too crude to produce more subtle flows of electrons. Transistors changed the way computers could be built.

A transistor is essentially a vacuum tube built, not to human size, but on a microscopic scale. Because it's small, a transistor requires less power to generate a flow of electrons. Because it uses less power, a transistor generates less heat, making computers more dependable. And the microscopic scale of transistors means that a computer that once took up an entire room now fits neatly on your lap.

All microchips, whether they're microprocessors, a memory chip, or a special-purpose integrated circuit, are basically vast collections of transistors arranged in different patterns so that they accomplish different tasks. Currently, the number of transistors that can be created on a single chip is about 16 million. The physical limitation is caused by how narrowly manufacturers can

focus the beams of light used to etch away transistor components that are made of light-sensitive materials. Chip makers are experimenting with X-rays instead of ordinary light because X-rays are much narrower. Someday, transistors may be taken to their logical extreme—the molecular level, in which the presence or absence of just one electron signals an On or Off state.

## KEY CONCEPTS

adder, half-adder, full-adder Differing combinations of transistors that perform mathematical and logical operations on data being processed.

address line An electrical line, or circuit, associated with a specific location in RAM.

arithmetic logic unit (ALU) The central part of a microprocessor that manipulates the data received by the processor.

ASCII The acronym for American Standard Code for Information Interchange.

binary Consists of only two integers—0 and 1. Binary math is the basis for manipulating all data in computers.

Boolean operations Logical operations, based on whether a statement is true or false, that are the equivalent of mathematical operations with numbers.

cache A block of high-speed memory where data is copied when it is retrieved from RAM. Then, if the data is needed again, it can be retrieved from the cache faster than from RAM. Level 1 cache is located on the motherboard separate from the processor. Level 2 is a part of the processor, making it even faster to retrieve.



capacitor A component that stores an electrical charge.

complex instruction set computing (CISC) A processor architecture design in which large, complicated instructions are broken down into smaller tasks before the processor executes them. See reduced instruction set computing.

data line An electrical line, or circuit, that carries data; specifically in RAM chips, a circuit that determines whether a bit represents a 0 or 1.

drain The part of a transistor where electrical current flows out of the transistor when it is closed.

gate A microcircuit design in which transistors are arranged so that the value of a bit of data can be changed.

megahertz (MHz) A measurement, in millions, of the number of times something oscillates or vibrates. Processor speeds are normally measured in megahertz.

microchip A sheet of silicon dioxide on which microscopic electrical circuits have been etched using a system of light, light-sensitive films, and acid baths.

microprocessor, processor The “brains” of a computer; a component that contains circuitry that can manipulate data in the form of binary bits. A microprocessor is contained on a single microchip.

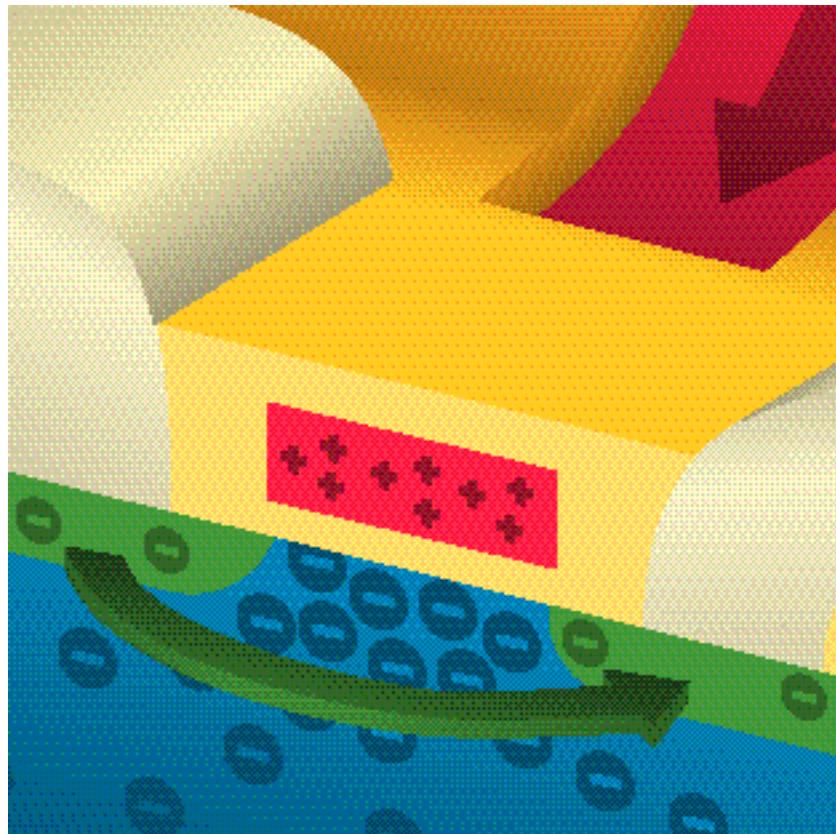
pipelining A computer architecture designed so that all parts of a circuit are always working, so that no part of the circuit is stalled waiting for data from another part.

reduced instruction set computing (RISC) A processor design in which only small, quickly executing instructions are used.

CHAPTER

## 7

# How a Transistor Works



THE transistor is the basic building block from which all microchips are built. The transistor can only create binary information: a 1 if current passes through or a 0 if current doesn't. From these 1s and 0s, called bits, a computer can create any number, provided it has enough transistors grouped together to hold all the 1s and 0s.

Binary notation starts off simply enough:

Decimal Number	Binary Number	Decimal Number	Binary Number
0	0	6	110
1	1	7	111
2	10	8	1000
3	11	9	1001
4	100	10	1010
5	101		

Personal computers such as the original IBM PC and AT systems based on the Intel 8088 and 80286 microprocessors are 16-bit PCs. That means they can work directly with binary numbers of up to 16 places or bits. That translates to the decimal number 65,536. If an operation requires numbers larger than that, the PC must first break those numbers into smaller components, perform the operation on each of the components, and then recombine the results into a single answer. More powerful PCs, such as those based on the Intel 80386, 80486, and Pentium, are 32-bit computers, which means they can manipulate binary numbers up to 32 bits wide—the equivalent in decimal notation of 4,294,967,296. The ability to work with 32 bits at a time helps make these PCs work much faster and be able to directly use more memory.

Transistors are not used simply to record and manipulate numbers. The bits can just as easily stand for true (1) or not true (0), which allows computers to deal with Boolean logic. (“Select this AND this but NOT this.”) Combinations of transistors in various configurations are called logic gates, which are combined into arrays called half adders, which in turn are combined into full adders. More than 260 transistors are needed to create a full adder that can handle mathematical operations for 16-bit numbers.

In addition, transistors make it possible for a small amount of electrical current to control a second, much stronger current—just as the small amount of energy needed to throw a wall switch can control the more powerful energy surging through the wires to give life to a spotlight.

# Transistor

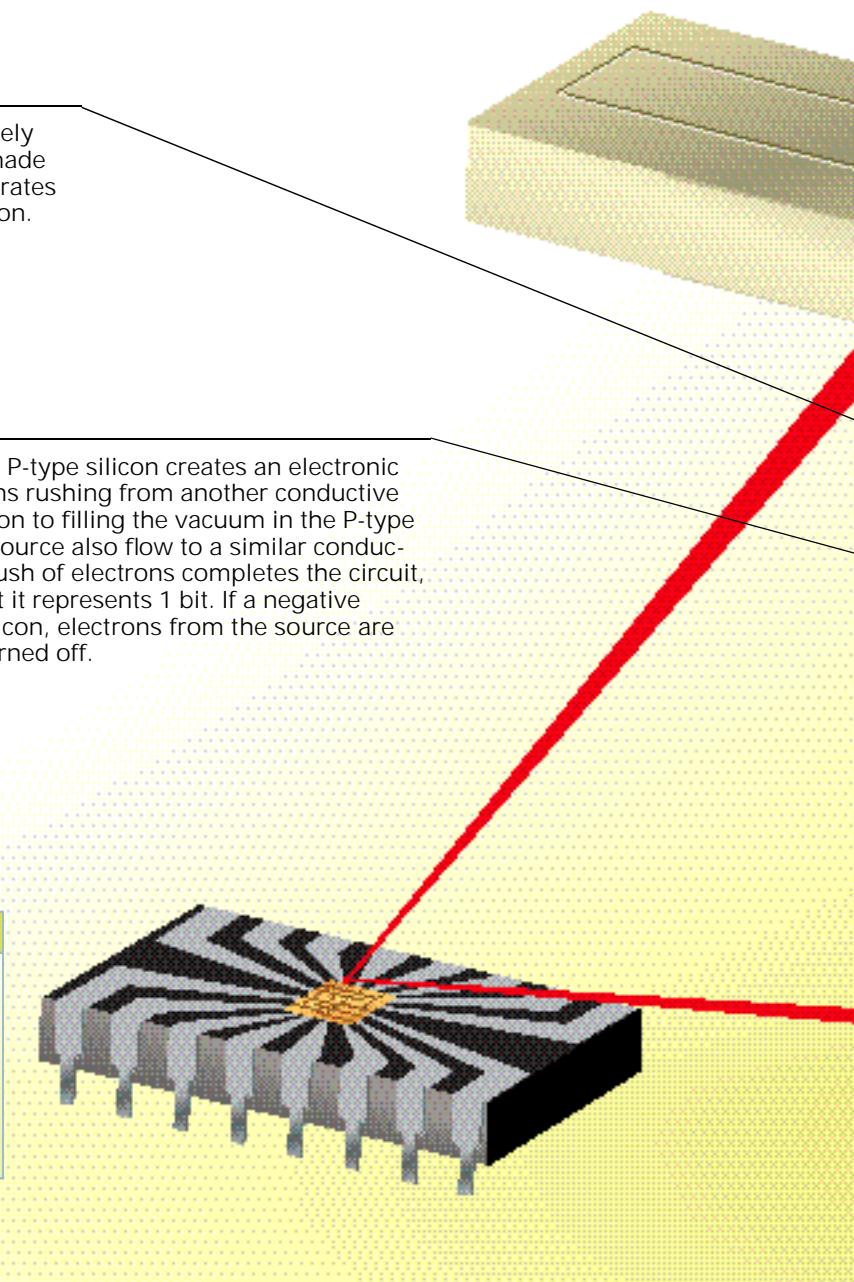
1 A small positive electrical charge is sent down one aluminum lead that runs into the transistor. The positive charge spreads to a layer of electrically conductive polysilicon buried in the middle of nonconductive silicon dioxide. Silicon dioxide is the main component of sand and the material that gave Silicon Valley its name.

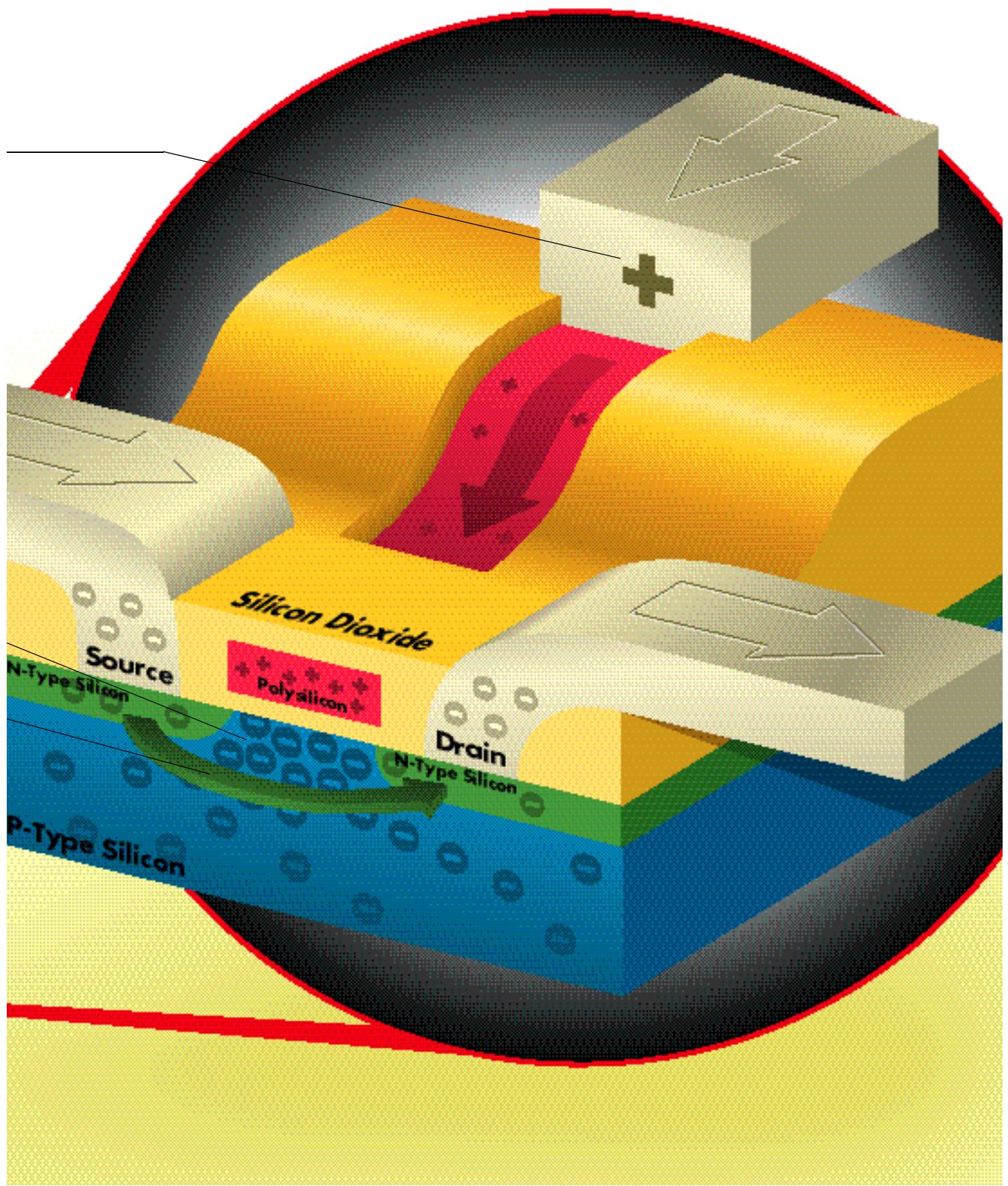
2 The positive charge attracts negatively charged electrons out of the base made of P-type (positive) silicon that separates two layers of N-type (negative) silicon.

3 The rush of electrons out of the P-type silicon creates an electronic vacuum that is filled by electrons rushing from another conductive lead called the source. In addition to filling the vacuum in the P-type silicon, the electrons from the source also flow to a similar conductive lead called the drain. The rush of electrons completes the circuit, turning on the transistor so that it represents 1 bit. If a negative charge is applied to the polysilicon, electrons from the source are repelled and the transistor is turned off.

## Creating a Chip from Transistors

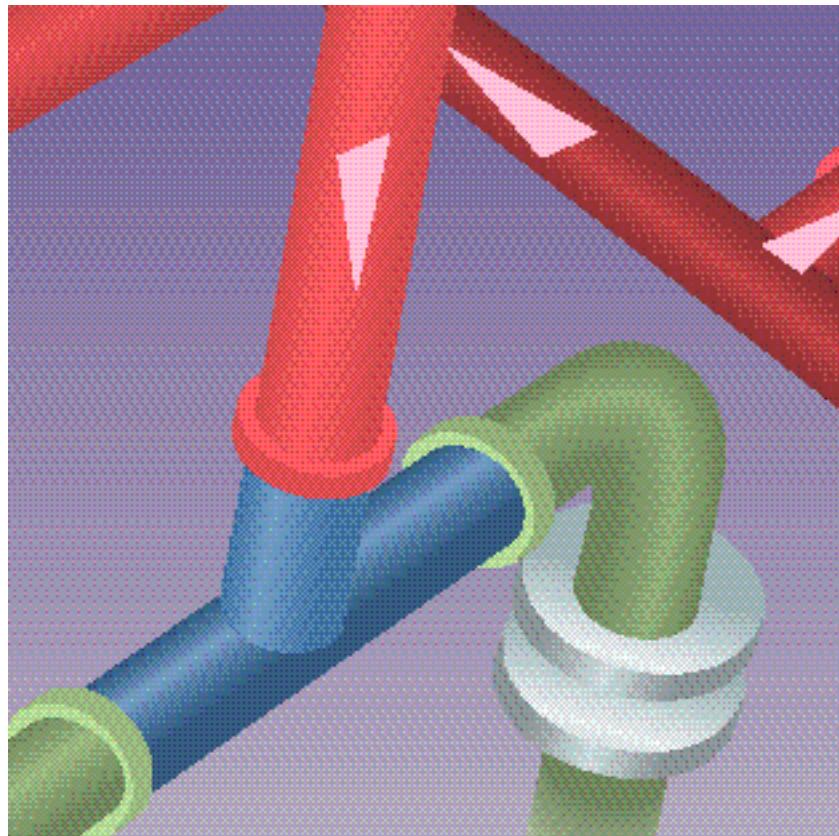
Thousands of transistors are connected on a single slice of silicon. The slice is embedded in a piece of plastic or ceramic material and the ends of the circuitry are attached to metal leads that expand in order to connect the chip to other parts of a computer circuit board. The leads carry signals into the chip and send signals from the chip to other computer components.





CHAPTER

## 8

How RAM  
Works

**RANDOM** access memory (RAM) chips are to your computer what a blank canvas is to an artist. Before a PC can do anything useful, it must move programs from disk to RAM. The data contained in documents, spreadsheets, graphics, databases, or any type of file must also be stored in RAM, if only momentarily, before the software can use the processor to manipulate that data.

Regardless of what kind of data a personal computer is using, and regardless of how complex that data may appear to us, to the PC that data exists only as 0s and 1s. Binary numbers are the native tongue of computers because even the biggest, most powerful computer essentially is no more than a collection of switches: An open switch represents a 0; a closed switch represents a 1. This is sometimes referred to as a computer's machine language. From this simplest of all numeric systems, your computer can construct representations of millions of numbers, any word in any language, and hundreds of thousands of colors and shapes.

Because humans are not nearly as fluent at binary notation as computers are, all those binary numbers are converted onscreen to some understandable notation—usually the alphabet or decimal numbers. For example, when you type an uppercase A, the operating system and software use a convention known as ASCII, in which certain numbers represent specific letters. The computer is essentially a number manipulator, which is why it's easier at the machine level for computers to deal with binary numbers. But it's easier for programmers and other humans to use decimal numbers. In ASCII, the capital A is the decimal number 65; B is 66; C is 67; and so on. Still, in the heart of a computer, the numbers are stored in their binary equivalents.

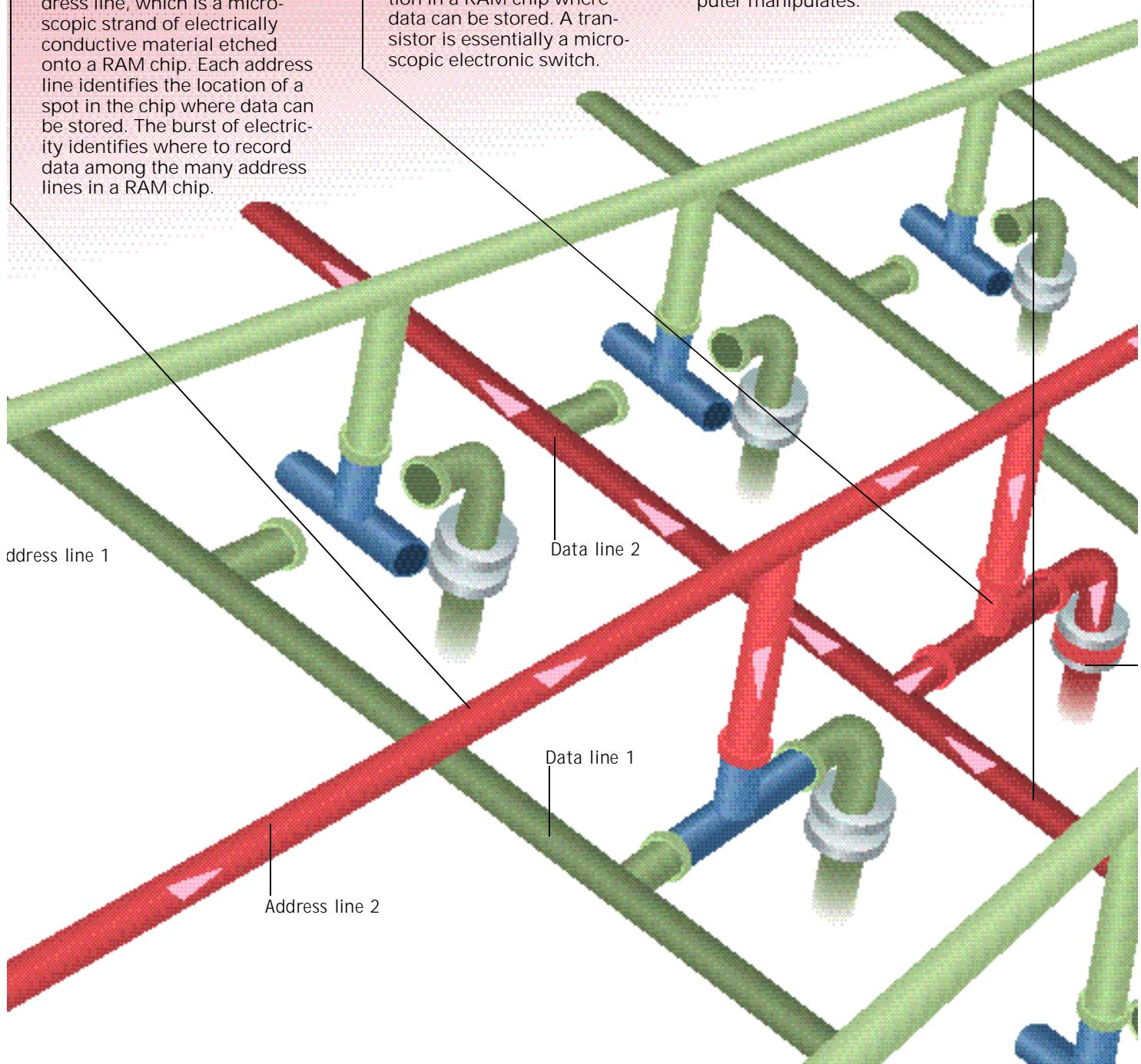
It is these binary notations that fill your disks and the PC's memory. When you first turn on your computer, its RAM is a blank slate. But the memory is quickly filled with 0s and 1s that are read from disk or created by the work you do with the computer. When you turn off your PC, anything that's contained in RAM disappears. Some forms of RAM, called flash memory, retain their electrical charges when a computer is turned off. But most memory chips work only if there is a source of electricity to constantly refresh the thousands or millions of individual electrical charges that make up the programs and data stored in RAM.

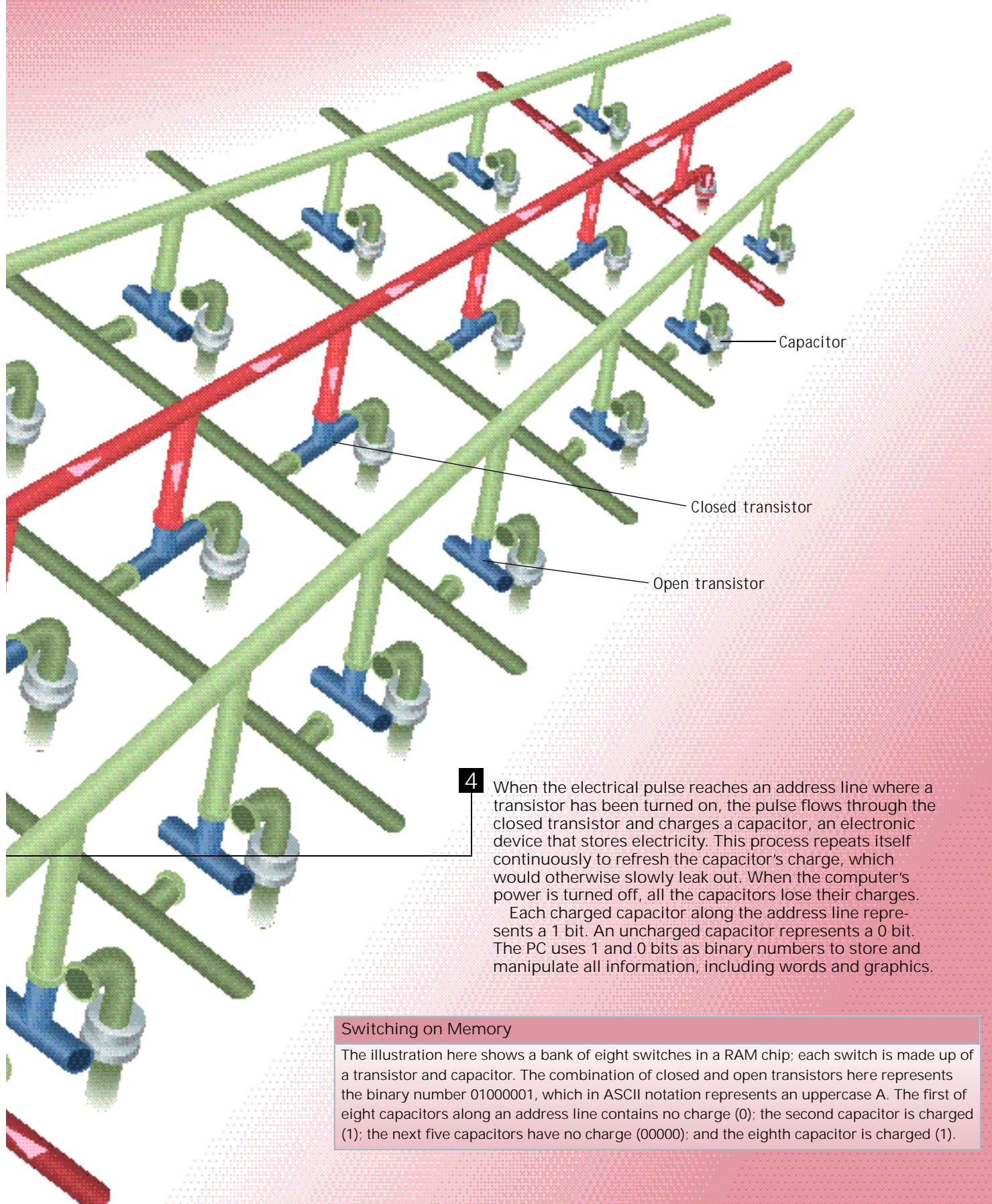
# Writing Data to RAM

1 Software, in combination with the operating system, sends a burst of electricity along an address line, which is a microscopic strand of electrically conductive material etched onto a RAM chip. Each address line identifies the location of a spot in the chip where data can be stored. The burst of electricity identifies where to record data among the many address lines in a RAM chip.

2 The electrical pulse turns on (closes) a transistor that's connected to a data line at each memory location in a RAM chip where data can be stored. A transistor is essentially a microscopic electronic switch.

3 While the transistors are turned on, the software sends bursts of electricity along selected data lines. Each burst represents a 1 bit, in the native language of processors—the ultimate unit of information that a computer manipulates.





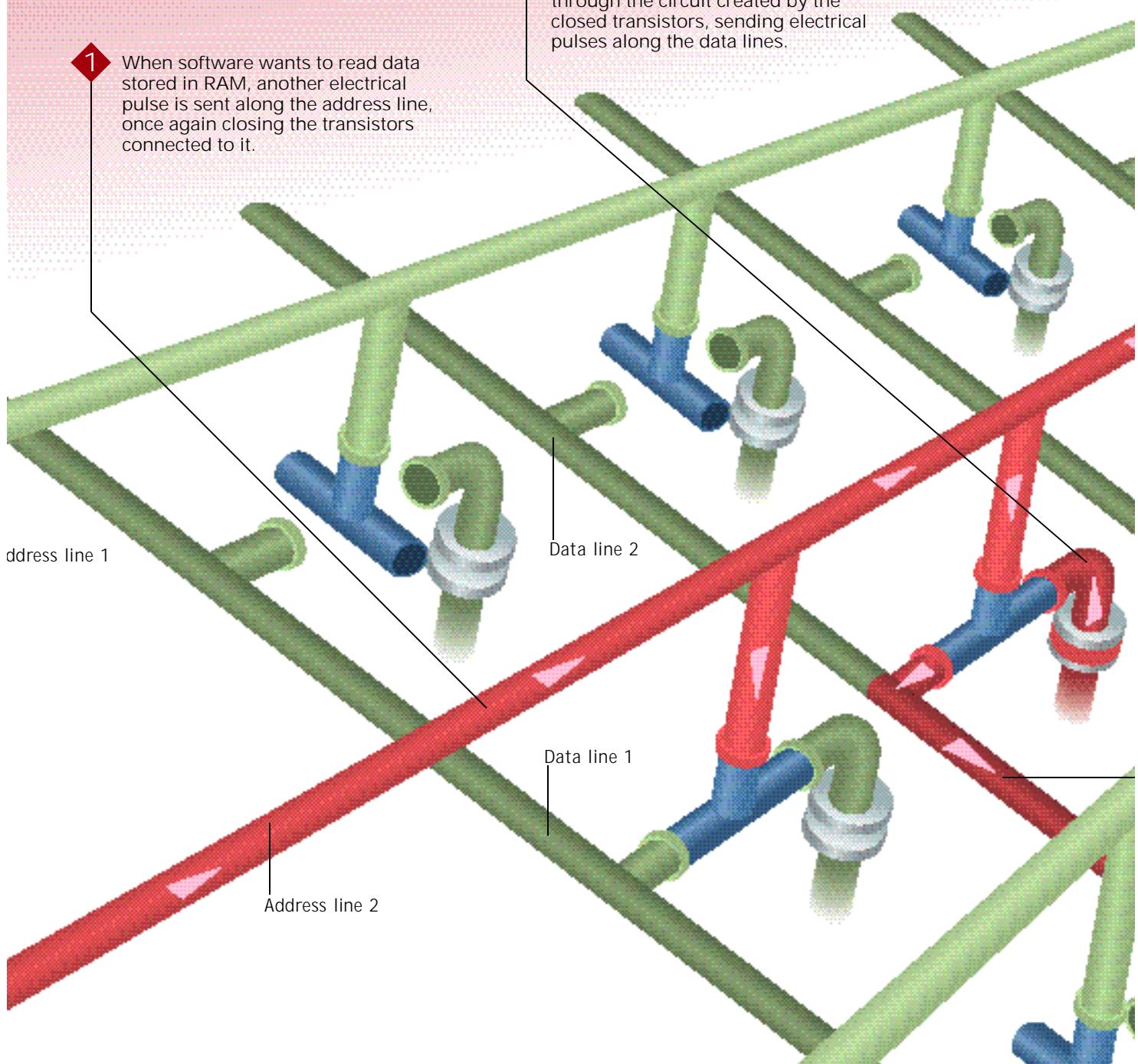
# Reading Data from RAM

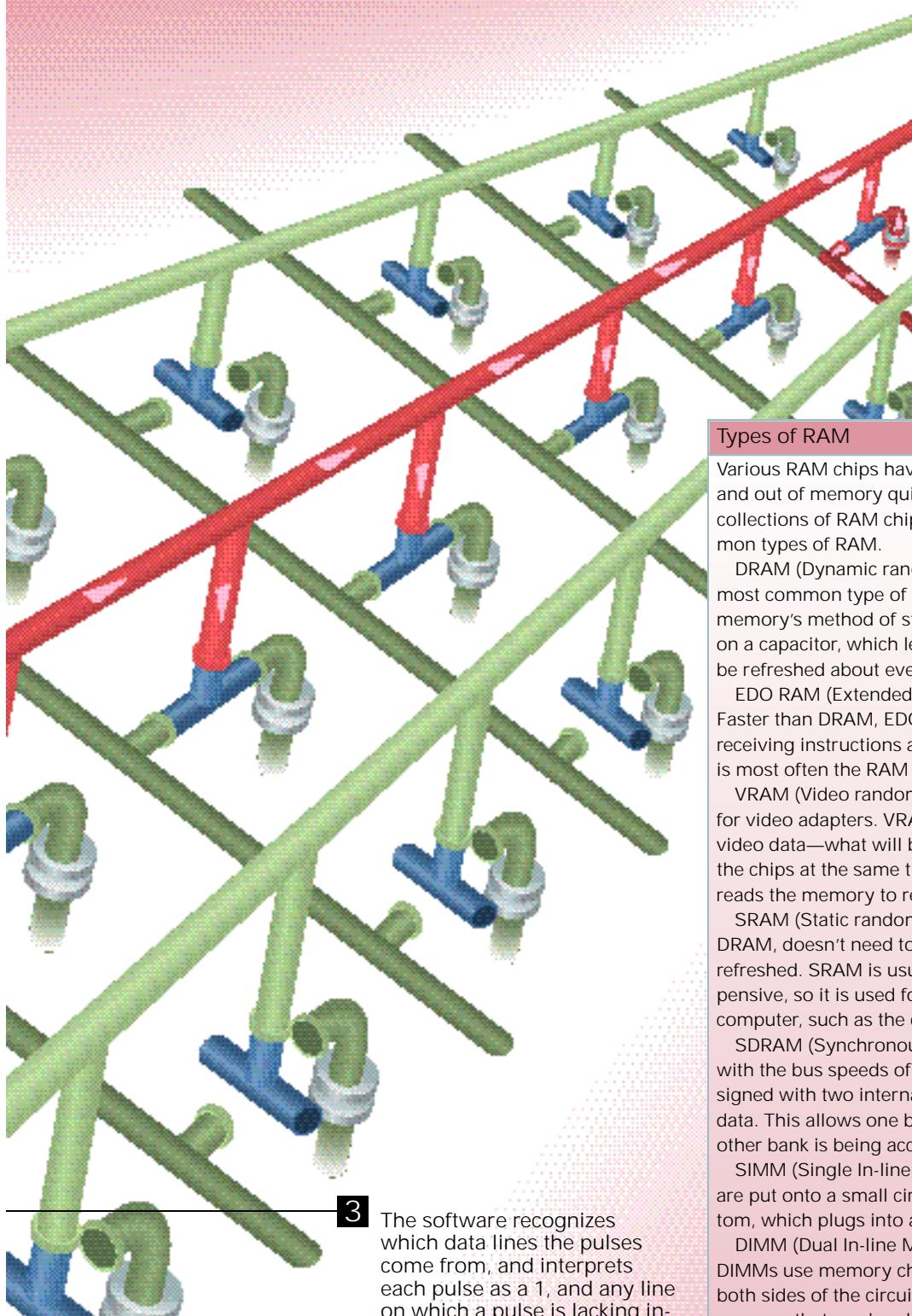
1

When software wants to read data stored in RAM, another electrical pulse is sent along the address line, once again closing the transistors connected to it.

2

Everywhere along the address line that there is a capacitor holding a charge, the capacitor will discharge through the circuit created by the closed transistors, sending electrical pulses along the data lines.





**3** The software recognizes which data lines the pulses come from, and interprets each pulse as a 1, and any line on which a pulse is lacking indicates a 0. The combination of 1s and 0s from eight data lines forms a single byte of data.

#### Types of RAM

Various RAM chips have been developed to move data in and out of memory quicker, to avoid errors, and to make collections of RAM chips smaller. Here are the most common types of RAM.

**DRAM (Dynamic random access memory)** For years, the most common type of main RAM. "Dynamic" refers to the memory's method of storage—basically storing the charge on a capacitor, which leaks the charge over time and must be refreshed about every thousandth of a second.

**EDO RAM (Extended Data Out random access memory)** Faster than DRAM, EDO memory can send data even while receiving instructions about what data to access next. This is most often the RAM of choice for Pentium-class PCs.

**VRAM (Video random access memory)** RAM optimized for video adapters. VRAM chips have two ports so that video data—what will be displayed next—can be written to the chips at the same time the video adapter continuously reads the memory to refresh the monitor's display.

**SRAM (Static random access memory)** RAM that, unlike DRAM, doesn't need to have its electrical charges constantly refreshed. SRAM is usually faster than DRAM but more expensive, so it is used for the most speed-critical parts of a computer, such as the cache.

**SDRAM (Synchronous DRAM)** RAM designed to keep up with the bus speeds of fast processors, SDRAMs are designed with two internal banks of transistors for storing data. This allows one bank to get ready for access while the other bank is being accessed.

**SIMM (Single In-line Memory Modules)** Memory chips are put onto a small circuit board with pins along the bottom, which plugs into a connector on the motherboard.

**DIMM (Dual In-line Memory Modules)** Similar to SIMMs, DIMMs use memory chips and separate connector pins on both sides of the circuit board to increase the amount of memory that can be plugged into a single connector and to increase the size of the data path for faster data transfers.

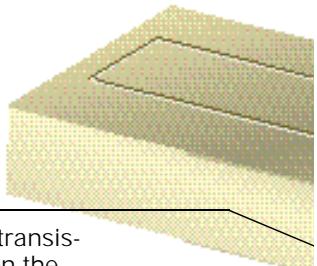
**ECC (Error-Correcting Code)** RAM that uses extra bits to detect errors. Types of memory such as DRAM, SDRAM, and DIMM may also be ECC chips.

# How Flash Memory Works



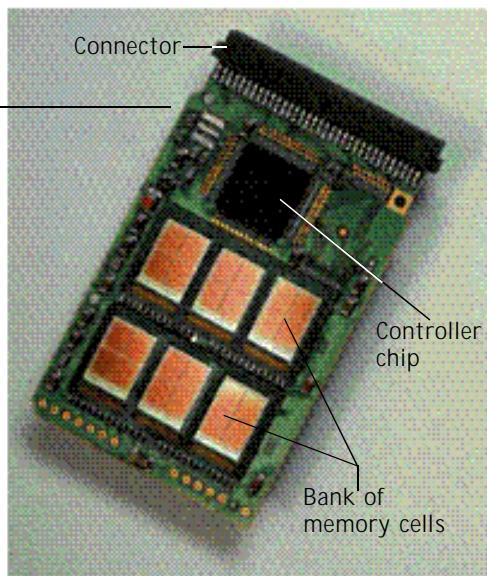
1

Nonvolatile flash memory does not lose its data even if all power, including battery sources, are turned off. Typically found in PC Cards or "compact flash" cards the size of matchbooks, flash memory is most often used in portable devices where its storage capacity of up to 80MB is adequate, such as digital cameras (see Chapter 27), voice recorders, and palm computers. It's also found in PCs as a replacement for EPROM memory, which holds BIOS information. It has not replaced hard drives because flash memory is more expensive and slower.



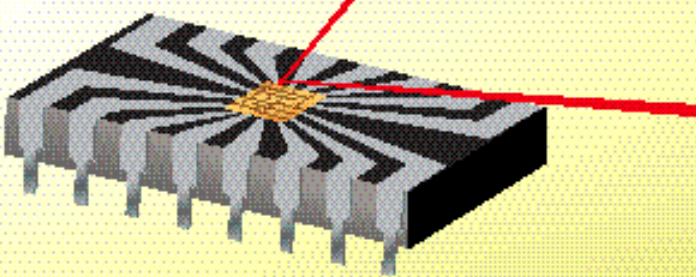
3

The significant difference from an ordinary transistor is the inclusion of a floating gate between the source and the drain. The floating gate is isolated from the other components by thin layers of oxide.



4

When a voltage is applied to the source, the positive charge in the polysilicon control gate attracts electrons, as in a transistor. (See Chapter 7.) Along the way, some of the electrons are pulled through the oxide layers and become trapped inside the floating gate. This process is called channel hot electron injection or tunneling. The extra negative charge (electrons) stays in the polysilicon-floating gate even when power is turned off. The presence of the electrons increases the amount of current needed to turn on the cell. That higher voltage requirement marks the cell as a binary zero.

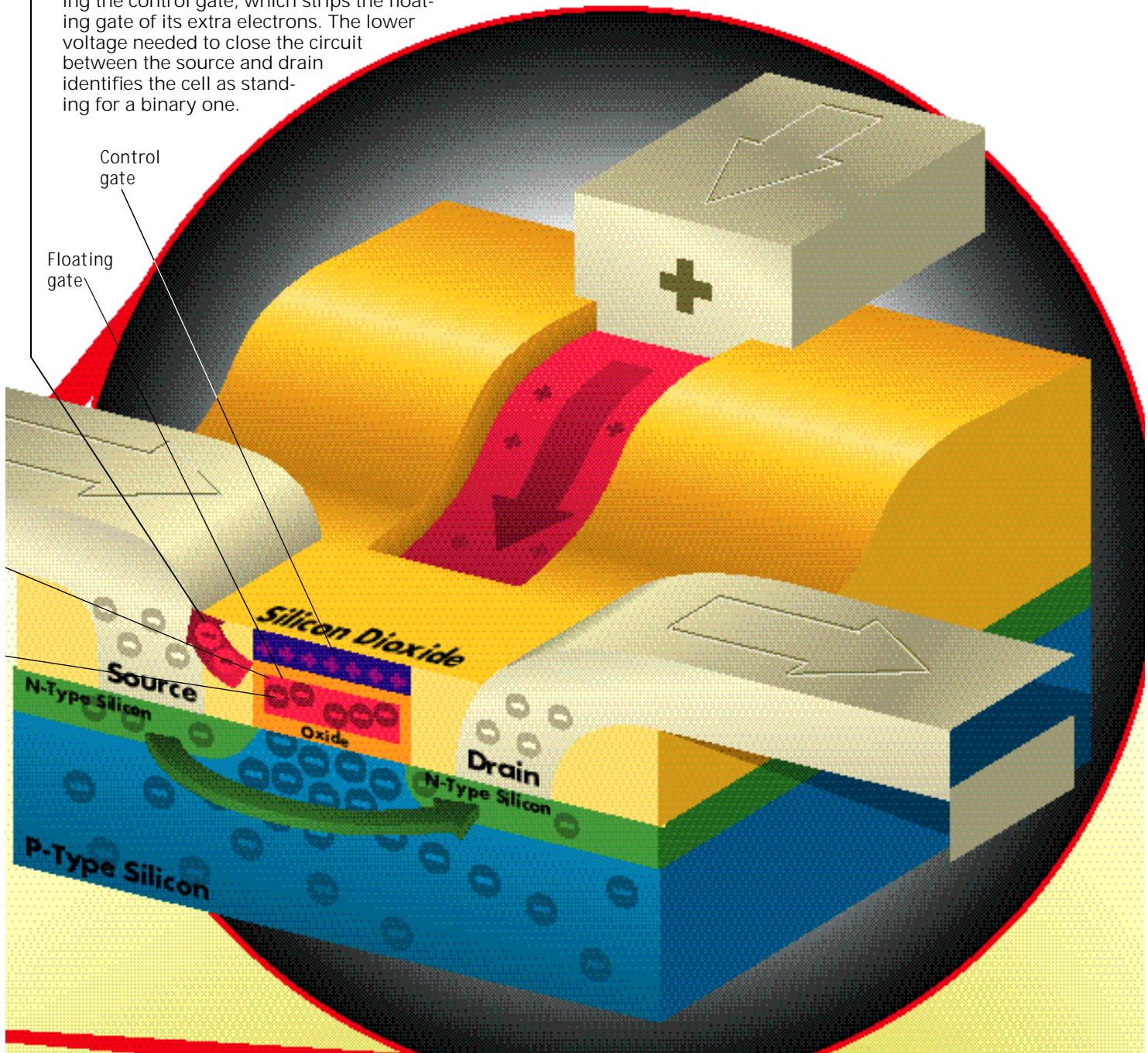


2

Flash memory cards are made up of millions of cells—a modified form of a transistor—where data or programs are stored. The card includes a controller chip, either built into the card or part of a connector attachment, or uses software drivers that make the card look like a hard drive to devices it's plugged into.

5

The cell is erased by applying a relatively high voltage to the source while grounding the control gate, which strips the floating gate of its extra electrons. The lower voltage needed to close the circuit between the source and drain identifies the cell as standing for a binary one.

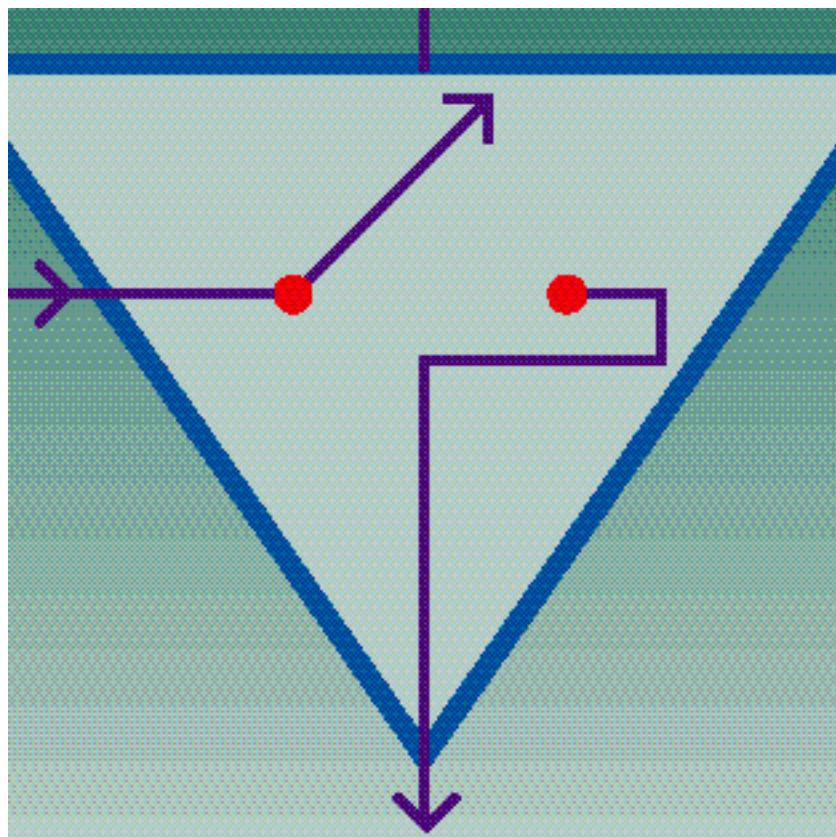


#### When Memories Fade

After a flash-memory cell has been written to and erased anywhere from 100,000 to 1 million times, it begins to malfunction. The oxide layer may break down or more electrons build up in the floating gate than can be drawn out when erasing the cell. To compensate for the lost cells, the flash-memory's controller marks the worn-out cell as bad, so that it knows not to write to it in the future, and calls into service extra cells included just for this purpose.

CHAPTER

## 9

How a Computer  
Performs Math

THE easiest way to visualize how computers work is to think of them as enormous collections of switches, which is really what they are—switches in the form of microscopic transistors etched into a slice of silicon. But for the moment, think of a computer as being a giant billboard made up of columns and rows of lights—thousands of them. Then imagine a control room behind that billboard in which there is a switch for each one of the lightbulbs on the sign. By turning on the correct switches, you can spell your name or draw a picture.

But suppose there are “master switches” that control dozens of other switches. Instead of having to flip each switch individually for every lightbulb that goes into spelling your name, you can throw one switch that lights up a combination of lights to create a B, then another master switch that turns on all the lights for an O, then another switch to light up another B.

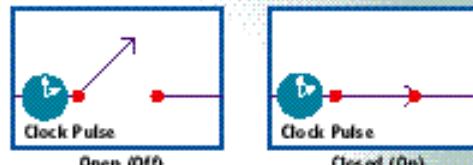
Now you’re very close to understanding how a computer works. In fact, substitute a computer display for the billboard and substitute RAM—which is a collection of transistorized switches—for the control room, and a keyboard for the master switches, and you have a computer performing one of its most basic functions, displaying what you type on screen.

Of course, a computer has to do a lot more than display words to be helpful. But the off and on positions of the same switches used to control a display also can add numbers by representing the 0 and 1 in the binary number system. And once you can add numbers, you can perform any kind of math, because multiplication is simply repeated addition, subtraction is adding a negative number, and division is repeated subtraction. And to a computer, everything—math, words, numbers, and software instructions—is numbers. This fact lets all those switches (transistors) do all types of data manipulation.

Actually, the first computers were more like our billboard in how they were used. They didn’t have keyboards or displays. The first computer users did actually throw a series of switches in a specific order to represent both data and the instructions for handling that data. Instead of transistors, the early computers used vacuum tubes, which were bulky and generated an enormous amount of heat. To get the computer’s answer, the people using it had to decipher what looked like a random display of lights. Even with the most underpowered PC you can buy today, you’ve still got it a lot better than the earliest computer pioneers.

# How a Computer Performs Addition

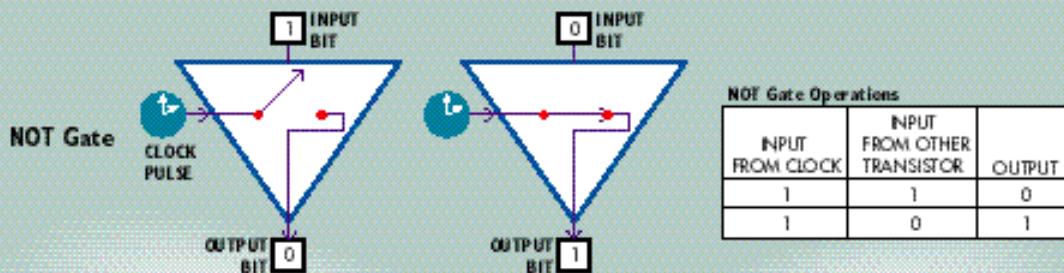
Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010



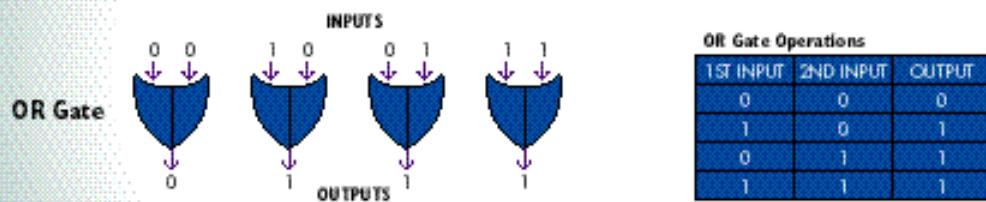
1 All information—words and graphics as well as numbers—is stored in and manipulated by a PC in the form of binary numbers. In the binary numerical system there are only two digits—0 and 1. All numbers, words, and graphics are formed from different combinations of those two digits.

2 Transistor switches are used to manipulate binary numbers because there are two possible states of a switch, open (off) or closed (on), which nicely matches the two binary digits. An open transistor, through which no current is flowing, represents a 0. A closed transistor, which allows a pulse of electricity regulated by the PC's clock to pass through, represents a 1. (The computer's clock regulates how fast the computer works. The faster a clock ticks, causing pulses of electricity, the faster the computer works. Clock speeds are measured in megahertz, or millions of ticks per second.) Current passing through one transistor can be used to control another transistor, in effect turning the switch on and off to change what the second transistor represents. Such an arrangement is called a gate because, like a fence gate, the transistor can be open or closed, allowing or stopping current flowing through it.

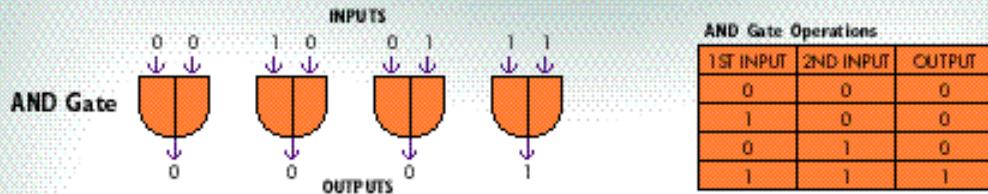
3 The simplest operation that can be performed with a transistor is called a NOT logic gate, made up of only a single transistor. The NOT gate is designed to take one input from the clock and one from another transistor. The NOT gate produces a single output—one that's always the opposite of the input from the other transistor. When current from another transistor representing a 1 is sent to a NOT gate, the gate's own transistor switch opens so that a pulse, or current, from the clock can't flow through it, which makes the NOT gate's output 0. A 0 input closes the NOT gate's transistor so that the clock pulse passes through it to produce an output of 1.



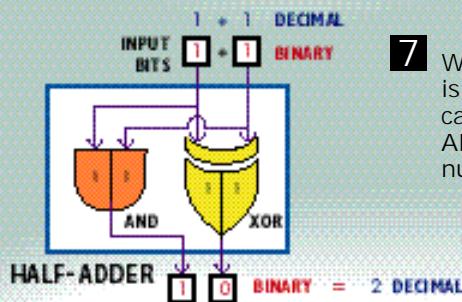
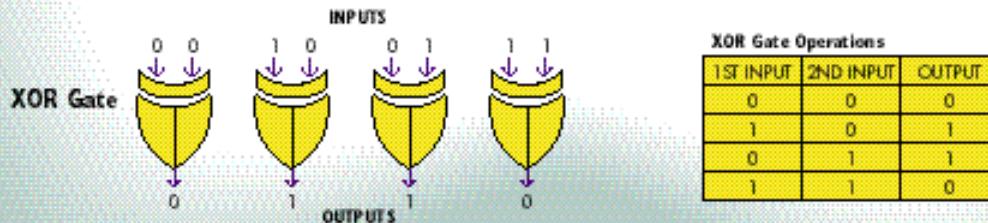
4 NOT gates strung together in different combinations create other logic gates, all of which have a line to receive pulses from the clock and two other input lines for pulses from other logic gates. The OR gate creates a 1 if either the first or the second input is a 1.



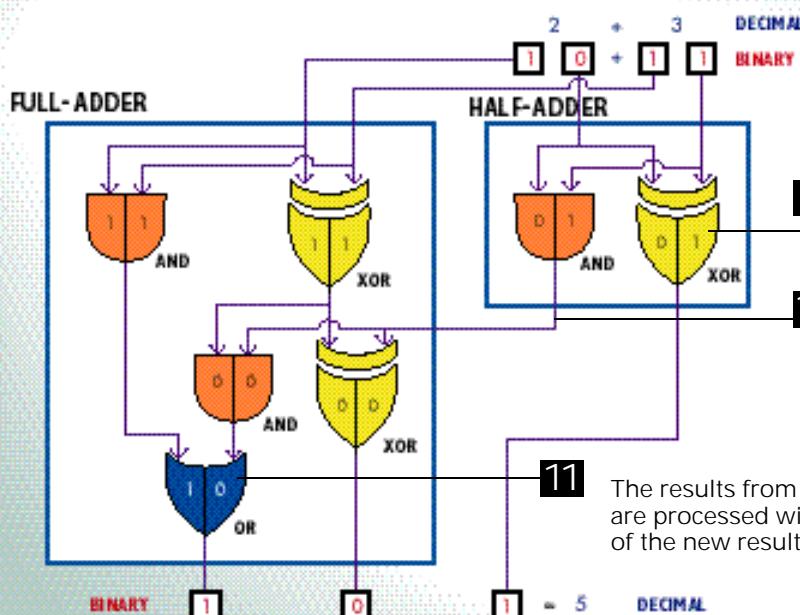
- 5 An AND gate outputs a 1 only if both the first input and the second input are 1s.



- 6 An XOR gate puts out a 0 if both the inputs are 0 or if both are 1. It generates a 1 only if one of the inputs is 1 and the other is 0.



- 7 With different combinations of logic gates, a computer performs the math that is the foundation of all its operations. This is accomplished with gate designs called half-adders and full-adders. A half-adder consists of an XOR gate and an AND gate, both of which receive the same input representing a one-digit binary number. A full-adder consists of half-adders and other switches.



- 8 A combination of a half-adder and a full-adder handles larger binary numbers and can generate results that involve carrying over numbers. To add the decimal numbers 2 and 3 (10 and 11 in the binary system), first the half-adder processes the digits on the right side through both XOR and AND gates.

- 9 The result of the XOR operation—1—becomes the rightmost digit of the result.

- 10 The result of the half-adder's AND operation—0—is sent to XOR and AND gates in the full-adder. The full-adder also processes the left-hand digits from 11 and 10, sending the results of both of the operations to another XOR gate and another AND gate.

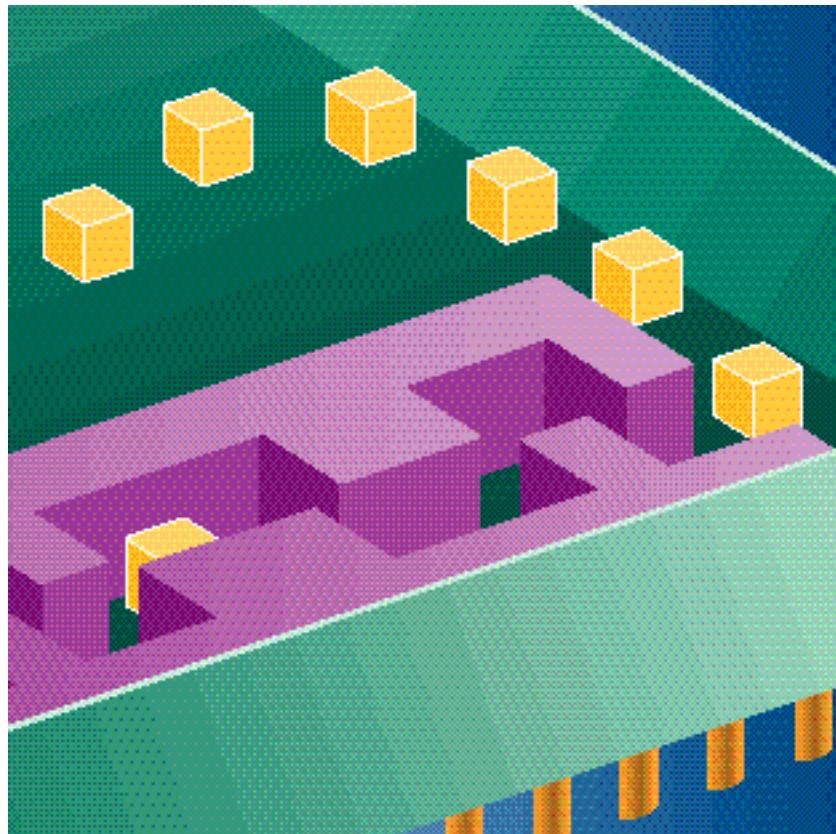
- 11 The results from XORing and ANDing the left-hand digits are processed with the results from the half-adder. One of the new results is passed through an OR gate.

- 12 The result of all the calculations is 101 in binary, which is 5 in decimal. For larger numbers, more full-adders are used—one for each digit in the binary numbers. An 80386 or later processor uses 32 full-adders.

CHAPTER

# 10

# How CISC and RISC Processors Work



THROUGH most of the history of personal computers, the predominant form of microprocessor has been from Intel Corp. The first processor in an IBM PC was Intel's 8088. The generations of Intel processors that followed it were in the 80X86 family—the 8086, 80286, 80386, and 80486. All were more elaborate versions of the original 8088, but improved on its performance in one of two ways—operating faster or handling more data simultaneously. The 8088, for example, operated at 4.7MHz—or 4.7 million frequency waves a second—and some 80486 chips go as fast as 133MHz. The 8088 could handle 8 bits of data at a time, and the 80486 handles 32 bits internally.

But despite the changes, the Intel processors through the 80486 were based on a design philosophy called CISC, for complex instruction set computing. CISC uses commands that incorporate many small instructions to carry out a single operation. It is a broadsword in the way it chops and slices data and code. An alternative design, by comparison, works more like a scalpel, cutting up smaller, more delicate pieces of data and code. The scalpel is called reduced instruction set computing (RISC). RISC designs are found in newer processors such as the Alpha, IBM's RISC 6000, the Power PC processor, and, although Intel doesn't call the Pentium processors RISC chips, they have much in common with how RISC executes code.

RISC is a less complicated design that uses several simpler instructions to execute a comparable operation in less time than a single CISC processor executes a large, complicated command. RISC chips can be physically smaller than CISC chips. And because they use fewer transistors, they're generally cheaper to produce and are less prone to overheating.

There have been many predictions that the future of processors is in RISC design, and that's probably correct. But there has not been a wholesale movement to RISC for two reasons, the most important of which is to maintain compatibility with the vast number of software applications that have been written to work with Intel's earlier CISC processors. The second reason is that you don't really receive the full benefit of RISC architecture unless you're using an operating system and programs that have been written and compiled specifically to take advantage of RISC operations. It's a classic chicken-and-egg situation. Some computer manufacturers are offering RISC processors as a way of projecting themselves to the leading edge of technology. They run old CISC programs only by emulating a CISC processor, which negates the RISC advantages. At the same time, software creators are reluctant to convert their programs to RISC-compiled versions when there aren't that many people who have RISC-based PCs.

But the Pentium chip has changed that. Techies may argue whether the Pentium is true RISC. It's a handy compromise. It runs old applications and operating systems, and at the same time it offers speed advantages for programs written specifically to tap Pentium capabilities. And with an advanced operating system such as Windows NT, you can put more than one Pentium microprocessor in a computer to double the processing power.

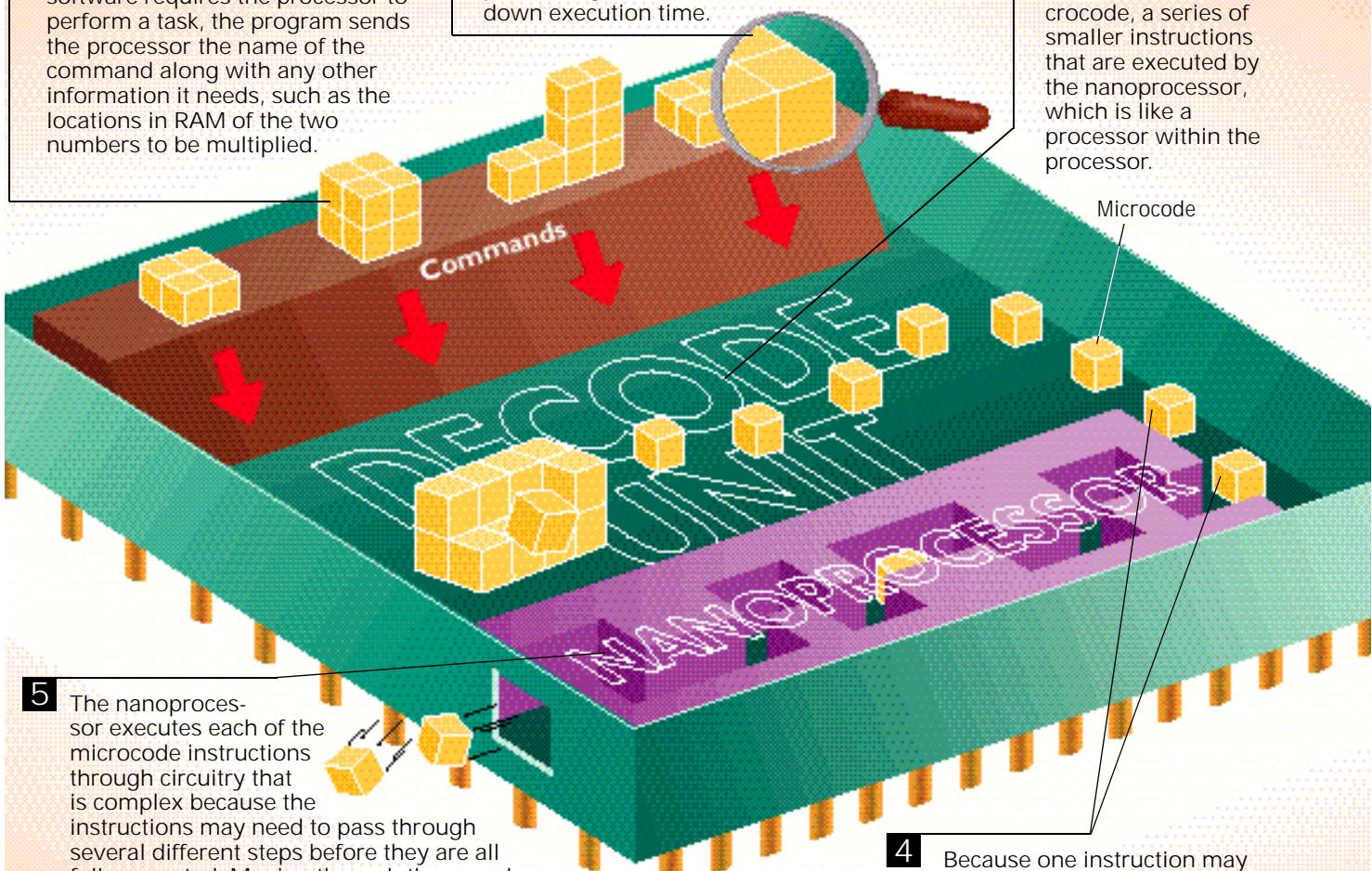
# CISC and RISC Processors

## Complex Instruction Set Computing (CISC)

1 Built into a CISC microprocessor's read-only memory is a large set of commands containing several sub-commands that must be carried out to complete a single operation, such as multiplying two numbers or moving a string of text to another location in memory. Whenever the operating system or application software requires the processor to perform a task, the program sends the processor the name of the command along with any other information it needs, such as the locations in RAM of the two numbers to be multiplied.

2 Because CISC commands are not all the same size, the microprocessor examines the command to determine how many bytes of processing room the command requires and then sets aside that much internal memory. There are also several different ways the commands can be loaded and stored, and the processor must determine the correct way to load and store each of the commands. Both of these preliminary tasks slow down execution time.

3 The processor sends the command requested by the software to a decode unit, which translates the complex command into microcode, a series of smaller instructions that are executed by the nanoprocessor, which is like a processor within the processor.

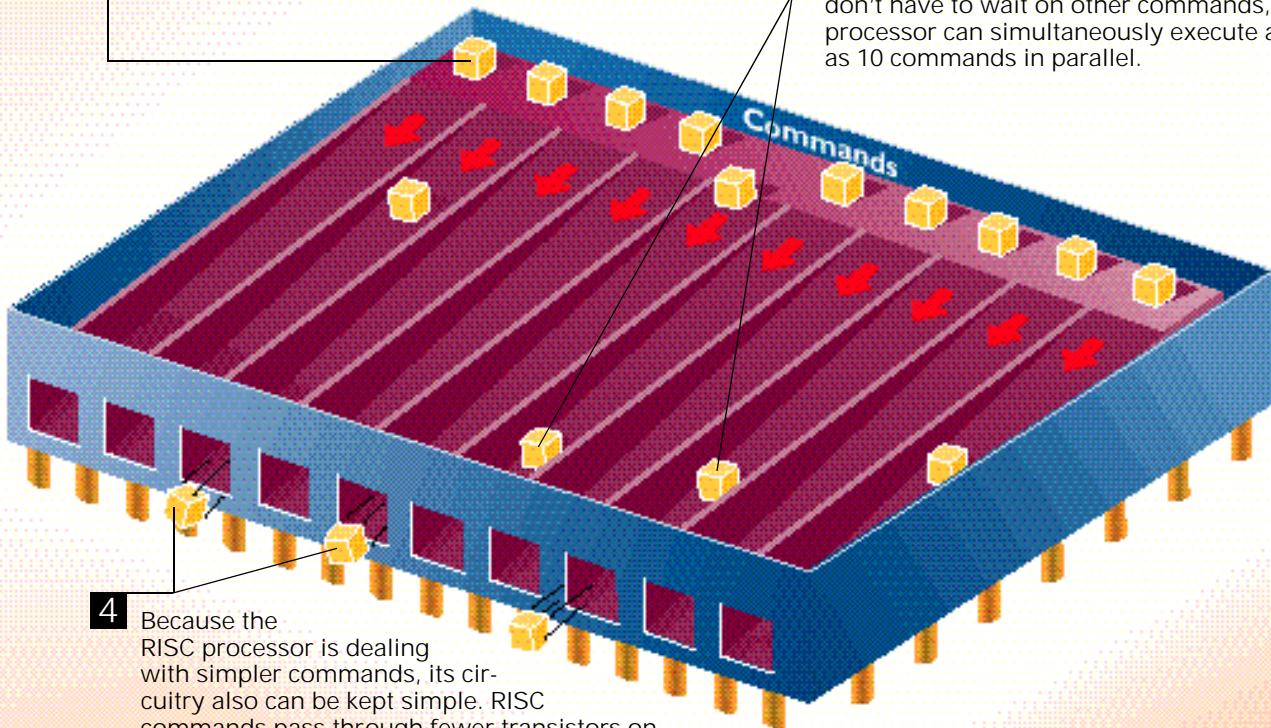


5 The nanoprocessor executes each of the microcode instructions through circuitry that is complex because the instructions may need to pass through several different steps before they are all fully executed. Moving through the complex circuits requires more time. CISC processors typically require between four and ten clock cycles to perform a single instruction. In an extreme case, an 80386 may take as many as 43 clock cycles to perform a single mathematical operation.

4 Because one instruction may depend on the results of another instruction, the instructions are performed one at a time. All other instructions stack up until the current instruction is completed.

## Reduced Instruction Set Computing (RISC)

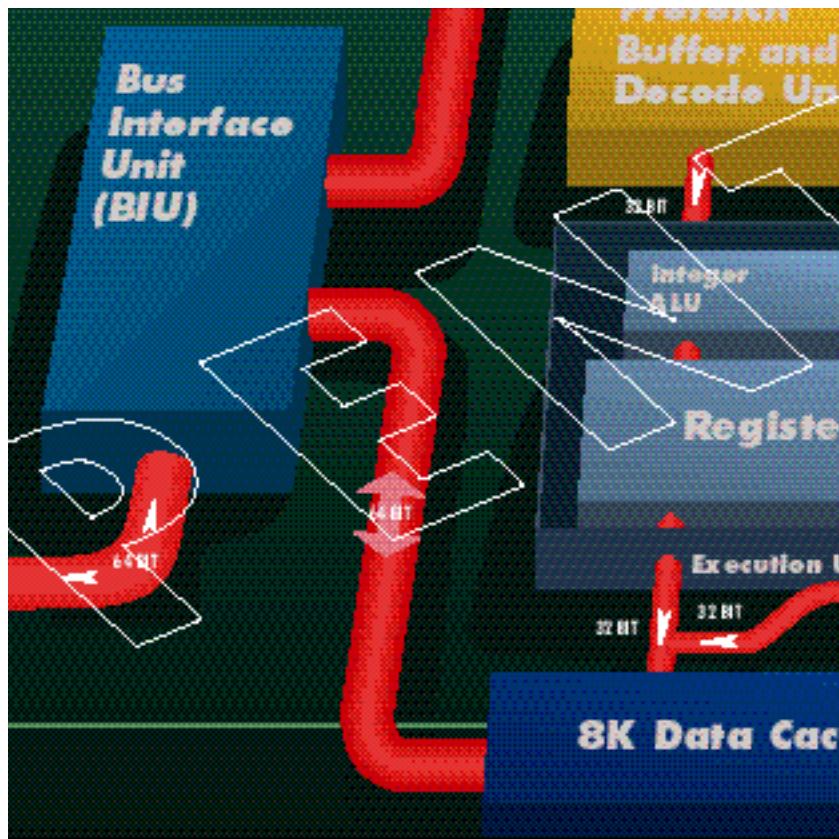
- 1 Command functions built into a RISC processor consist of several small, discrete instructions that perform only a single job. Application software, which must be recompiled especially for a RISC processor, performs the task of telling the processor which combination of its smaller commands to execute in order to complete a larger operation.
- 2 All RISC commands are the same size, and there is only one way in which they can be loaded and stored. In addition, since each command is already a form of microcode, RISC processors don't require the extra step of passing instructions they receive through a decode unit to translate complex commands into simpler microcode. As a result of these differences, RISC commands are loaded for execution far more quickly than CISC commands.
- 3 During the compilation of software specifically for a RISC chip, the compiler determines which commands will not depend on the results of other commands. Because these commands don't have to wait on other commands, the processor can simultaneously execute as many as 10 commands in parallel.
- 4 Because the RISC processor is dealing with simpler commands, its circuitry also can be kept simple. RISC commands pass through fewer transistors on shorter circuits, so the commands execute more quickly. The result is that RISC processors usually require only one CPU clock cycle per instruction. The number of cycles required to complete a full operation is dependent on the number of small commands that make up that operation. But, for a comparable operation, the time required to interpret and execute RISC instructions is far less than the time needed to load and decode a complex CISC command and then execute each of its components.



CHAPTER

## 11

# How a Microprocessor Works



THE microprocessor that makes up your personal computer's central processing unit, or CPU, is the computer's brains, messenger, ringmaster, and boss. All the other components—RAM, disk drives, the monitor—exist only to bridge the gap between you and the processor. They take your data and turn it over to the processor to manipulate; then they display the results. The CPU isn't the only microprocessor in PCs. Coprocessors on graphics, 3-D accelerator, and sound cards juggle display and sound data to relieve the CPU of part of its burden. And special processors, such as the one inside your keyboard that handles the signals generated whenever you press a key, perform specialized tasks designed to get data into or out of the CPU.

The current standard for high-performance processors is Intel's Pentium II chip design. On two combined chips that together are less than a couple of square inches, the Pentium II holds 7.5 million transistors, or tiny electronic switches. All the operations of the Pentium are performed by signals turning on or off different combinations of those switches. In computers, transistors are used to represent zeros and ones, the two numerals that make up the binary number system. These zeros and ones are commonly known as bits. Various groupings of these transistors make up the subcomponents within the Pentium, as well as those in coprocessors, memory chips, and other forms of digital silicon.

Many of the components of the original Pentium II are designed to get data in and out of the chip quickly and to make sure that the parts of the Pentium that do the actual data manipulation never have to go into idle because they're waiting on more data or instructions. These components have to handle the flow of data and instructions through the processor, interpret the instructions so they can be executed by the processor, and send the results back to the PC's memory.

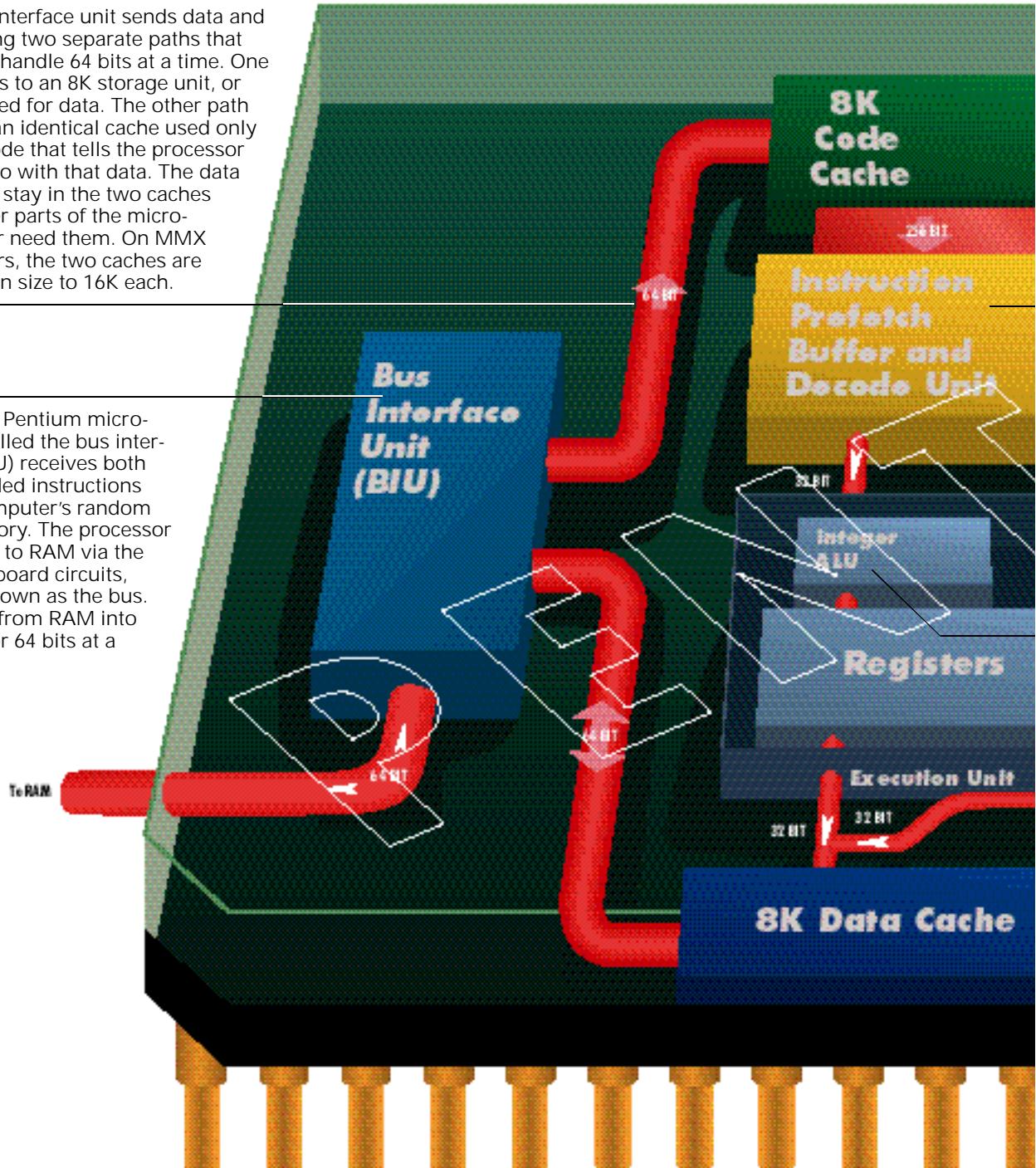
The Pentium family of processors—including the Pentium MMX, Pentium Pro, Pentium II, Celeron and Xeon—has several improvements over its predecessor (Intel's 80486 processor) that help ensure that data and code move through the Pentium as fast as possible. One of the most important changes is in the arithmetic logic unit (ALU). Just think of the ALU as sort of a brain within the brains. The ALU handles all the data juggling that involves integers, or whole numbers, such as 1, 23, 610, or 234. The Pentium is the first Intel processor to have two ALUs so that it can crunch two sets of numbers at the same time. Like the 486, the Pentium has a separate calculation unit that's optimized for handling floating-point numbers, or numbers with decimal fractions, such as 1.2 or 35.8942.

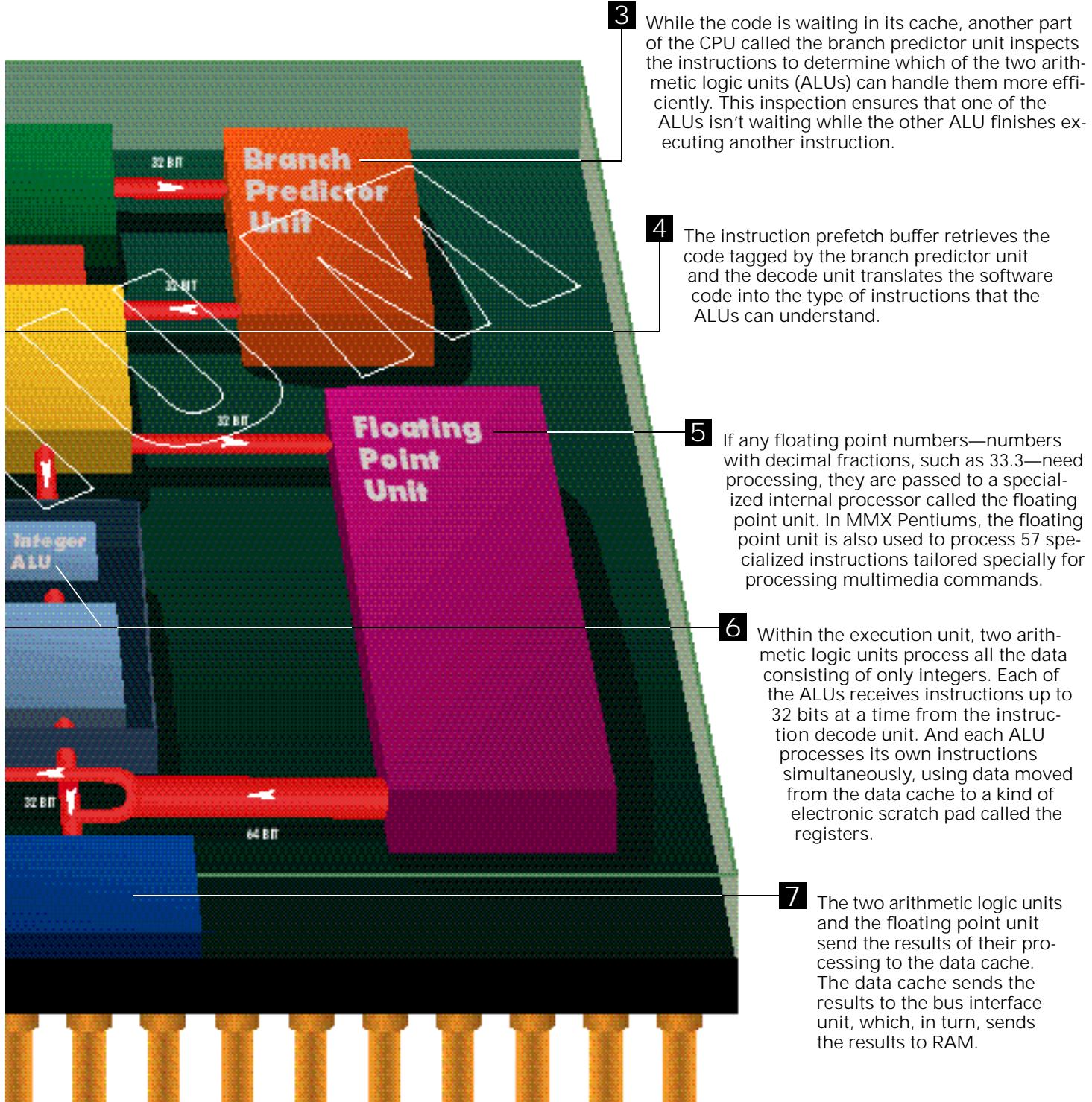
Another significant difference over the 486 is that the Pentium can take in data 64 bits at a time, compared to the 32-bit data path of the 486. Where the 486 has one storage area called a cache that holds 8 kilobytes at a time, the Pentium II has multiple caches totaling more than 512K. They're designed to make sure the ALU is constantly supplied with the data and instructions it needs to do its job.

# Pentium Processor

2 The bus interface unit sends data and code along two separate paths that can each handle 64 bits at a time. One path leads to an 8K storage unit, or cache, used for data. The other path leads to an identical cache used only for the code that tells the processor what to do with that data. The data and code stay in the two caches until other parts of the microprocessor need them. On MMX processors, the two caches are doubled in size to 16K each.

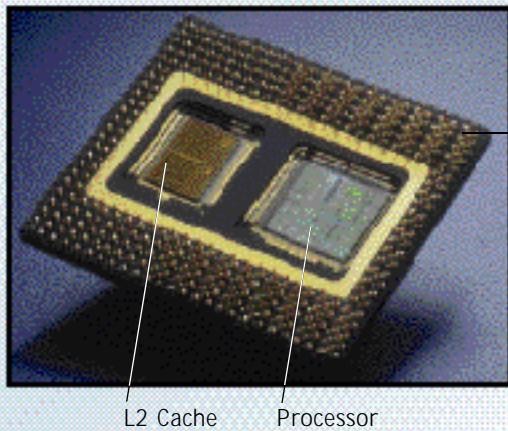
1 A part of the Pentium microprocessor called the bus interface unit (BIU) receives both data and coded instructions from the computer's random access memory. The processor is connected to RAM via the PC's motherboard circuits, which are known as the bus. Data moves from RAM into the processor 64 bits at a time.





# How the Pentium Pro Works

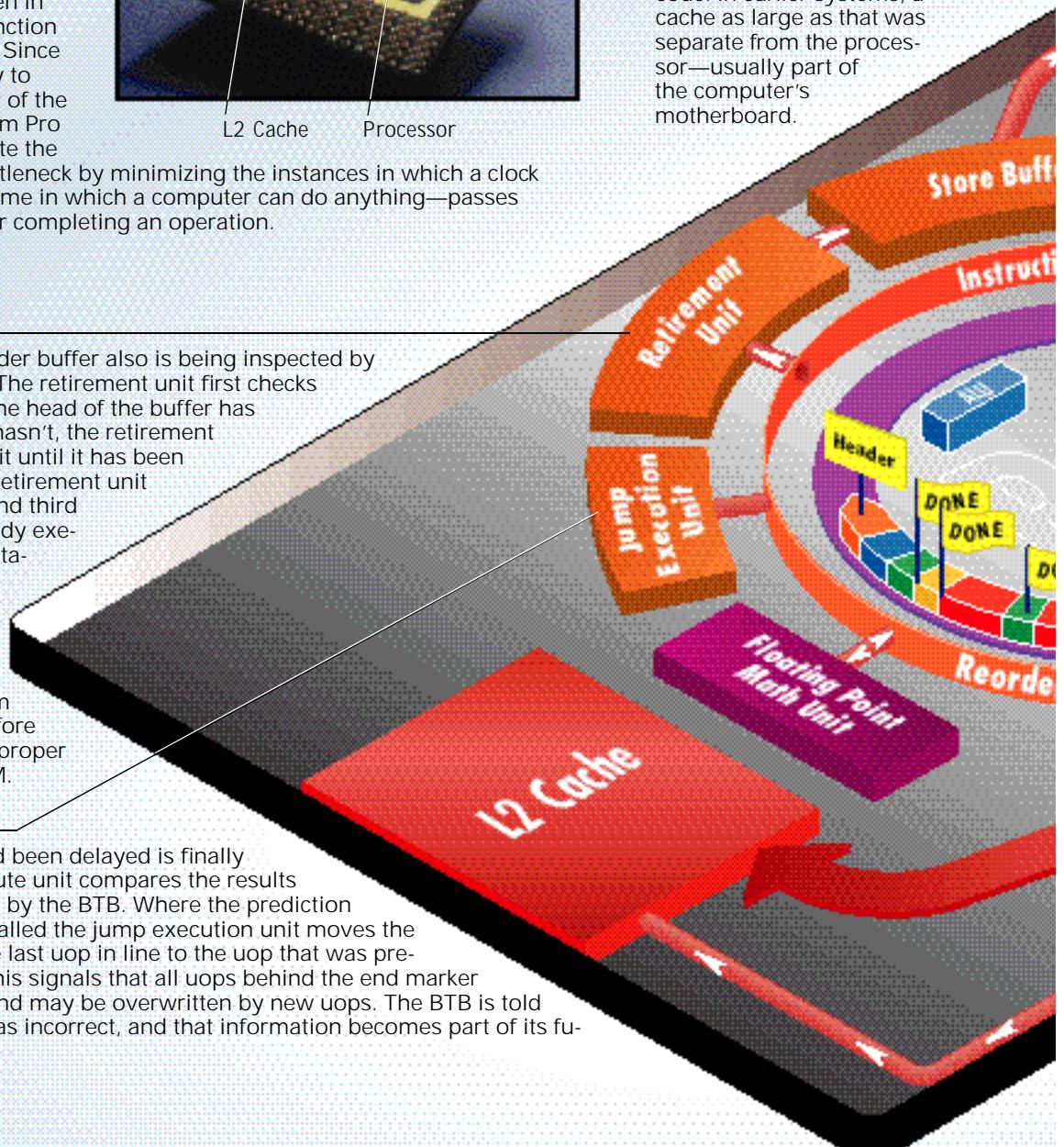
**1** The Pentium Pro processor shares the same 64-bit interface to the computer found in the older, non-Pro version of the Pentium. Information—either program code or data manipulated by that code—moves in and out of the chip at the PC's maximum bus speed, which is no more than 66MHz even in Pentium Pros that function internally at 200Mhz. Since there is no faster way to move data in and out of the processor, the Pentium Pro is designed to alleviate the effects of the bus bottleneck by minimizing the instances in which a clock cycle—the smallest time in which a computer can do anything—passes without the processor completing an operation.

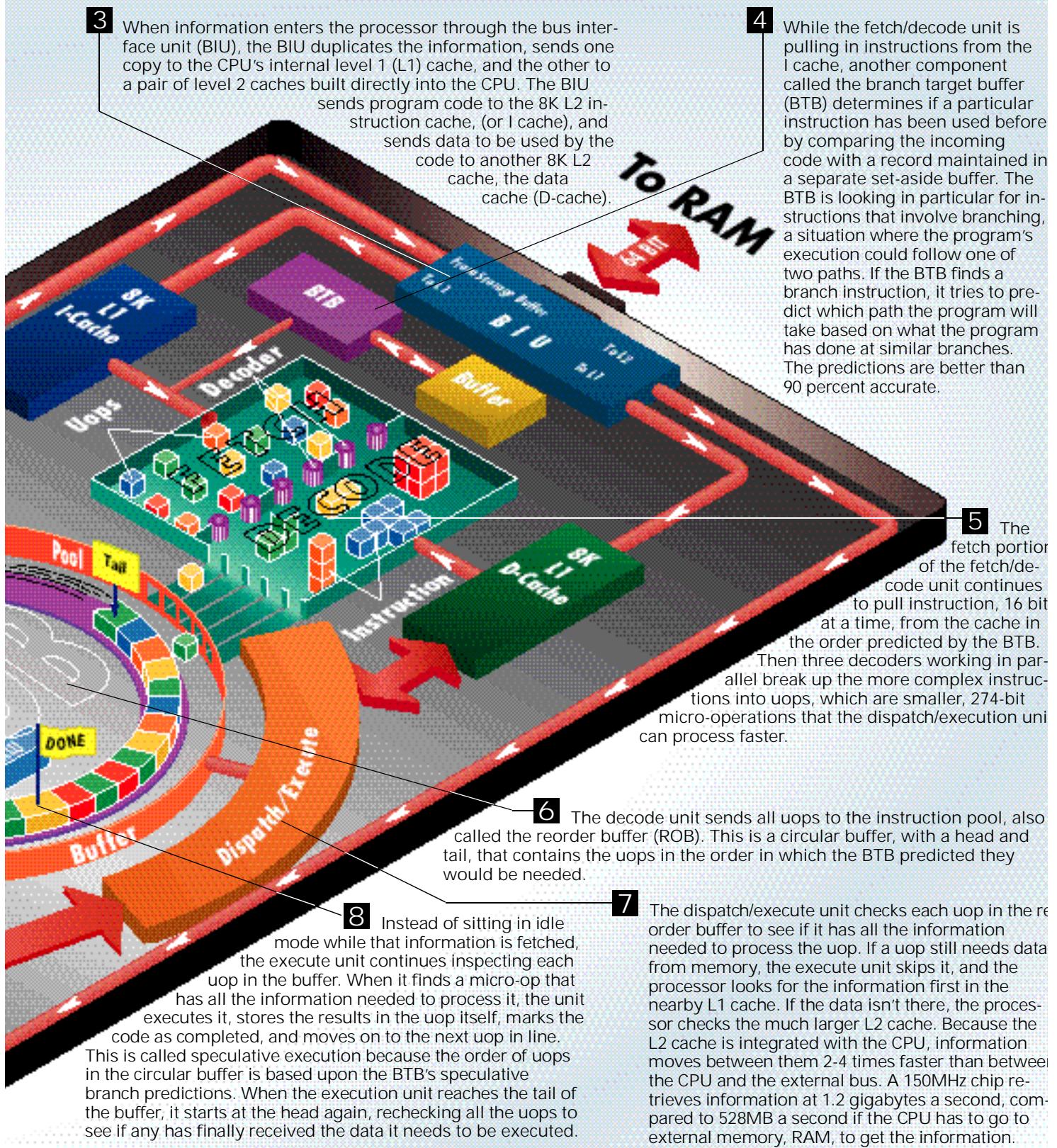


**2** The Pentium Pro is made up of two silicon dies. One is the 5.5-million transistor CPU, where the software's instructions are executed. The other is a level 2 (L2), custom-designed high-speed memory cache. Its 15.5 million transistors store up to 512K of data and code. In earlier systems, a cache as large as that was separate from the processor—usually part of the computer's motherboard.

**10** Meanwhile, the reorder buffer also is being inspected by the retirement unit. The retirement unit first checks to see if the uop at the head of the buffer has been executed. If it hasn't, the retirement unit keeps checking it until it has been executed. Then the retirement unit checks the second and third uops. If they're already executed, the unit simultaneously sends all three results—it's maximum—to the store buffer. There, the prediction unit checks them out one last time before they're sent to their proper place in system RAM.

**9** When a uop that had been delayed is finally processed, the execute unit compares the results with those predicted by the BTB. Where the prediction fails, a component called the jump execution unit moves the end marker from the last uop in line to the uop that was predicted incorrectly. This signals that all uops behind the end marker should be ignored and may be overwritten by new uops. The BTB is told that its correction was incorrect, and that information becomes part of its future predictions.



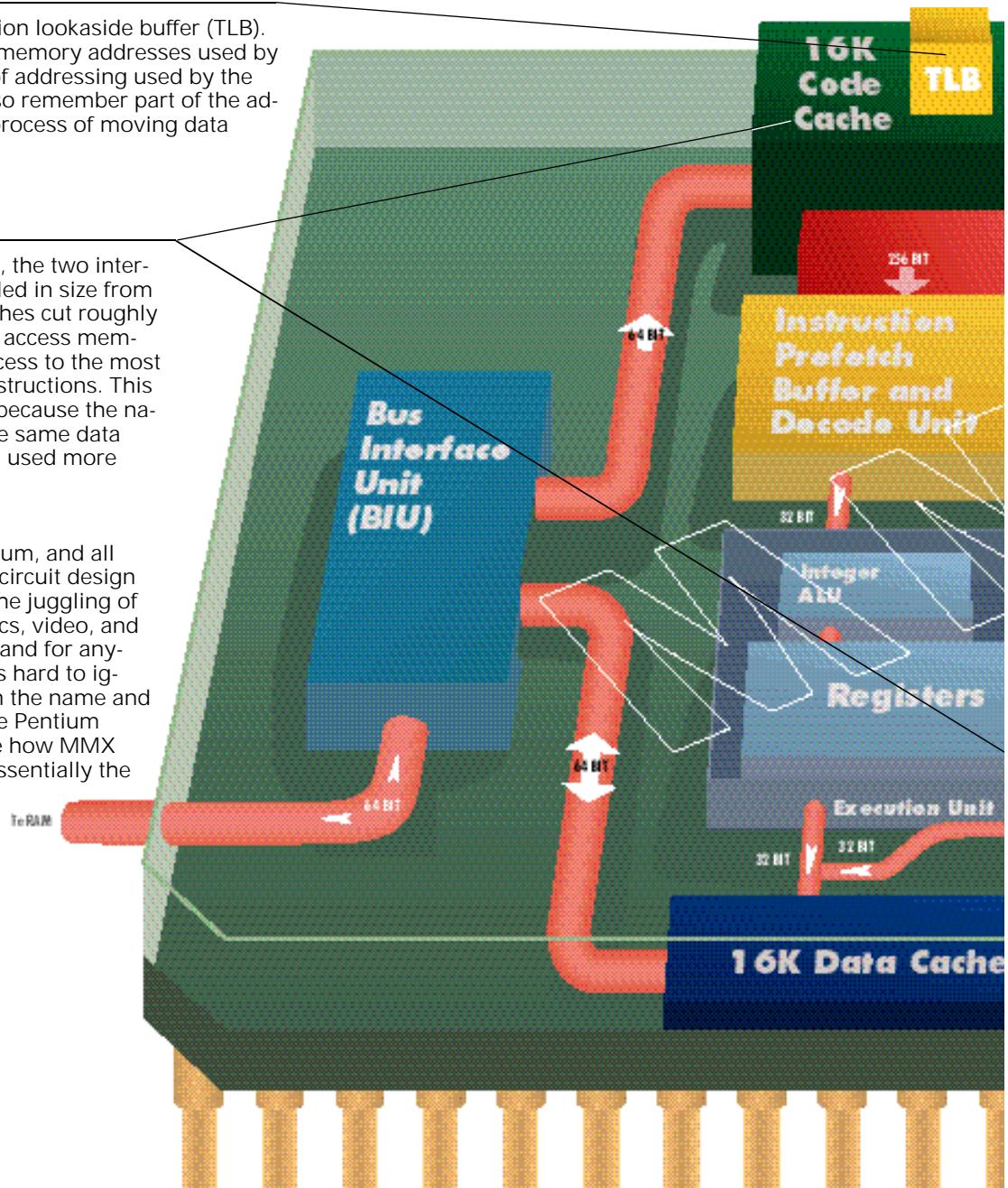


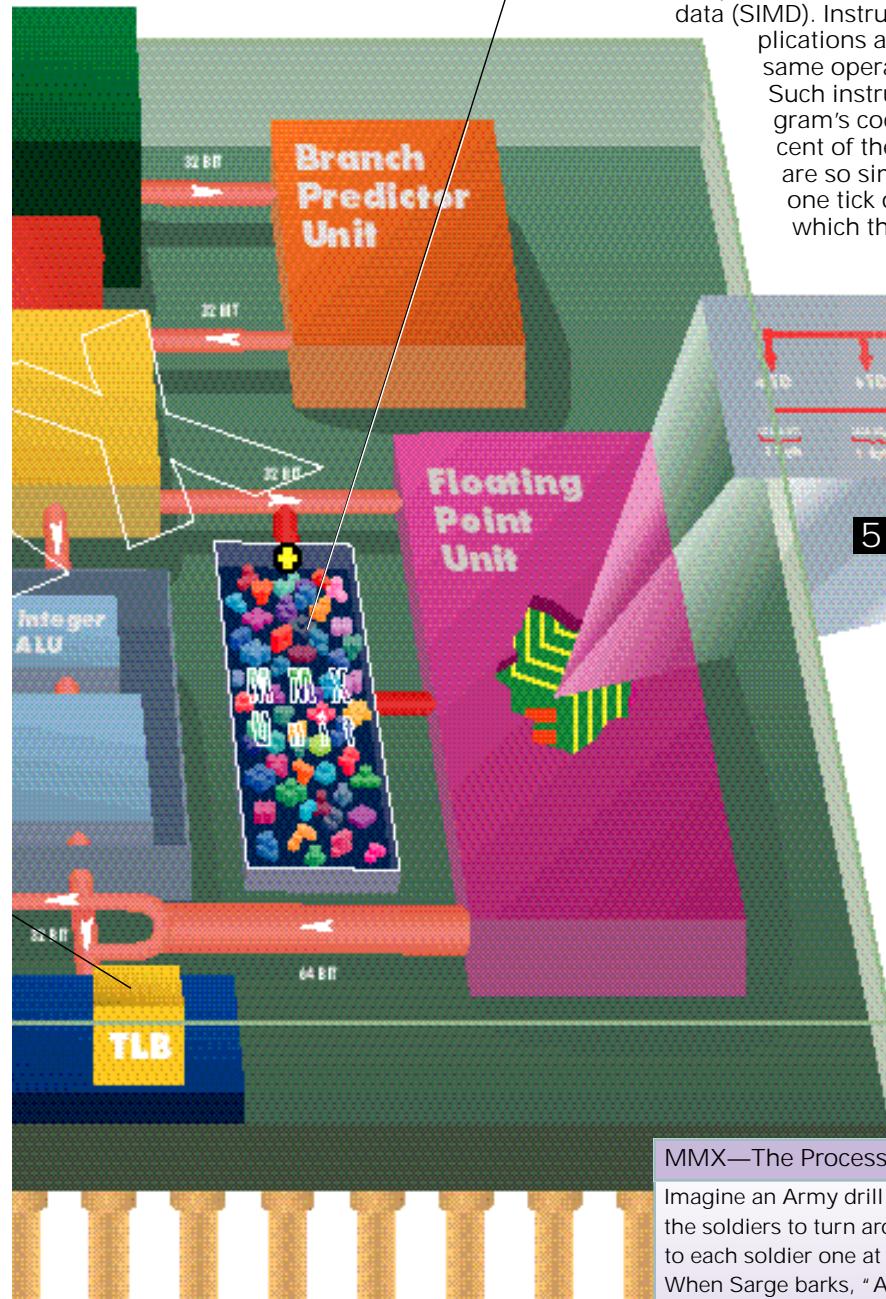
# How MMX Works

3 Each cache has a translation lookaside buffer (TLB). The buffers translate the memory addresses used by the caches into the type of addressing used by the processor. The buffers also remember part of the addresses to speed up the process of moving data from one spot to another.

2 In the Pentium MMX chip, the two internal caches are both doubled in size from 8K to 16K. The bigger caches cut roughly in half the time it takes to access memory and provide faster access to the most recently used data and instructions. This last feature is important, because the nature of software is that the same data and instructions are often used more than once.

1 Embedded into some Pentium, and all Pentium II, processors is a circuit design called MMX. It speeds up the juggling of multimedia data for graphics, video, and sound. "MMX" does not stand for anything officially, although it's hard to ignore the similarity between the name and "multimedia." We'll use the Pentium MMX chip here to illustrate how MMX works, but the process is essentially the same in the Pentium II.  
(See next illustration.)





4

If the software asks the processor to perform certain types of operations that involve graphics, audio, or video, those requests are routed to the MMX unit. The unit contains 57 specialized instructions tailored to perform the small, repetitive operations usually found in multimedia. Similar in concept to RISC processing (see Chapter 10), the MMX process is called single instruction/multiple data (SIMD). Instructions in multimedia and communication applications are often looped, or repeated, performing the same operation during each loop, only to different data. Such instructions may take up only 10 percent of a program's code, but they can account for as much as 90 percent of the software's execution time. SIMD instructions are so simple, the processor usually completes them in one tick of the computer's clock, the shortest time in which the PC can do anything.

5

The MMX unit uses part of the registers in the adjoining Floating Point Unit (FPU). The FPU has 80-bit registers, which means that each register can hold 80 0s or 1s. The MMX unit uses only 64 of each register's places. But this allows MMX to process special instructions faster than the main processing unit—the ALU—for two reasons. One is that the ALU has only 32-bit registers. The second reason is that multiples of multimedia data often fit neatly into the 64-bit MMX registers. For example, video and graphics palettes—the range of colors displayed—are often made of 8-bit (1-byte) values. With SIMD, a graphics program can put eight 1-byte values into a single 64-bit register and then use a single MMX instruction to add, say, the number 10 simultaneously to the value of each byte, brightening equally the portion of the image they represent. Most audio data is in 16-bit units so that four units will fit into a single 64-bit register for parallel execution.

#### MMX—The Processor's Drill Sergeant

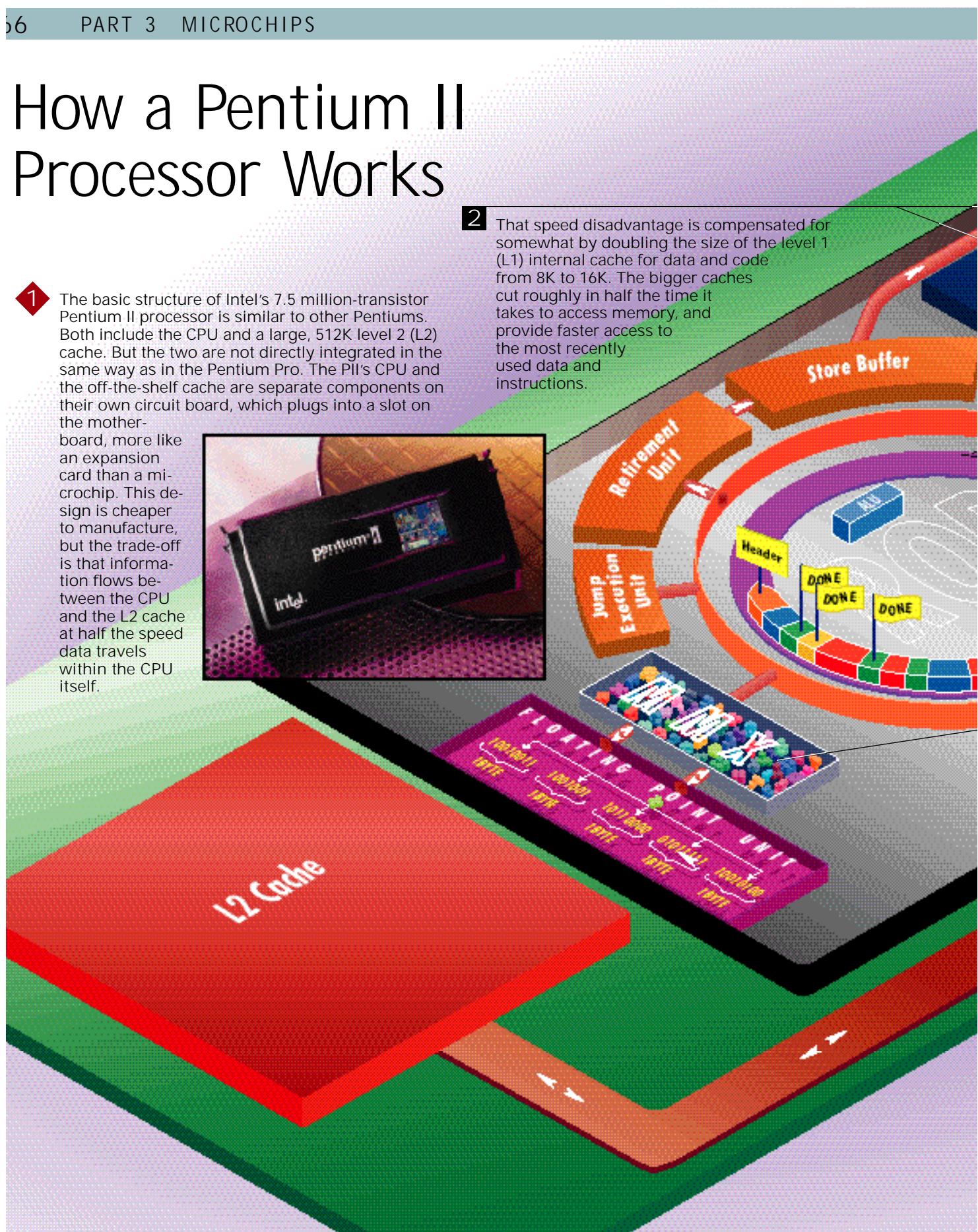
Imagine an Army drill sergeant facing an entire platoon. If the sergeant wants the soldiers to turn around, he could give the same instruction, "About face!" to each soldier one at a time. But drill sergeants are naturally MMX-powered. When Sarge barks, "About face!" he applies the same command to everyone in the platoon, and each soldier executes the command at the same time. The sergeant's command is a real-life SIMD, with a single instruction ("About face!"), and multiple data (the soldiers in the platoon).

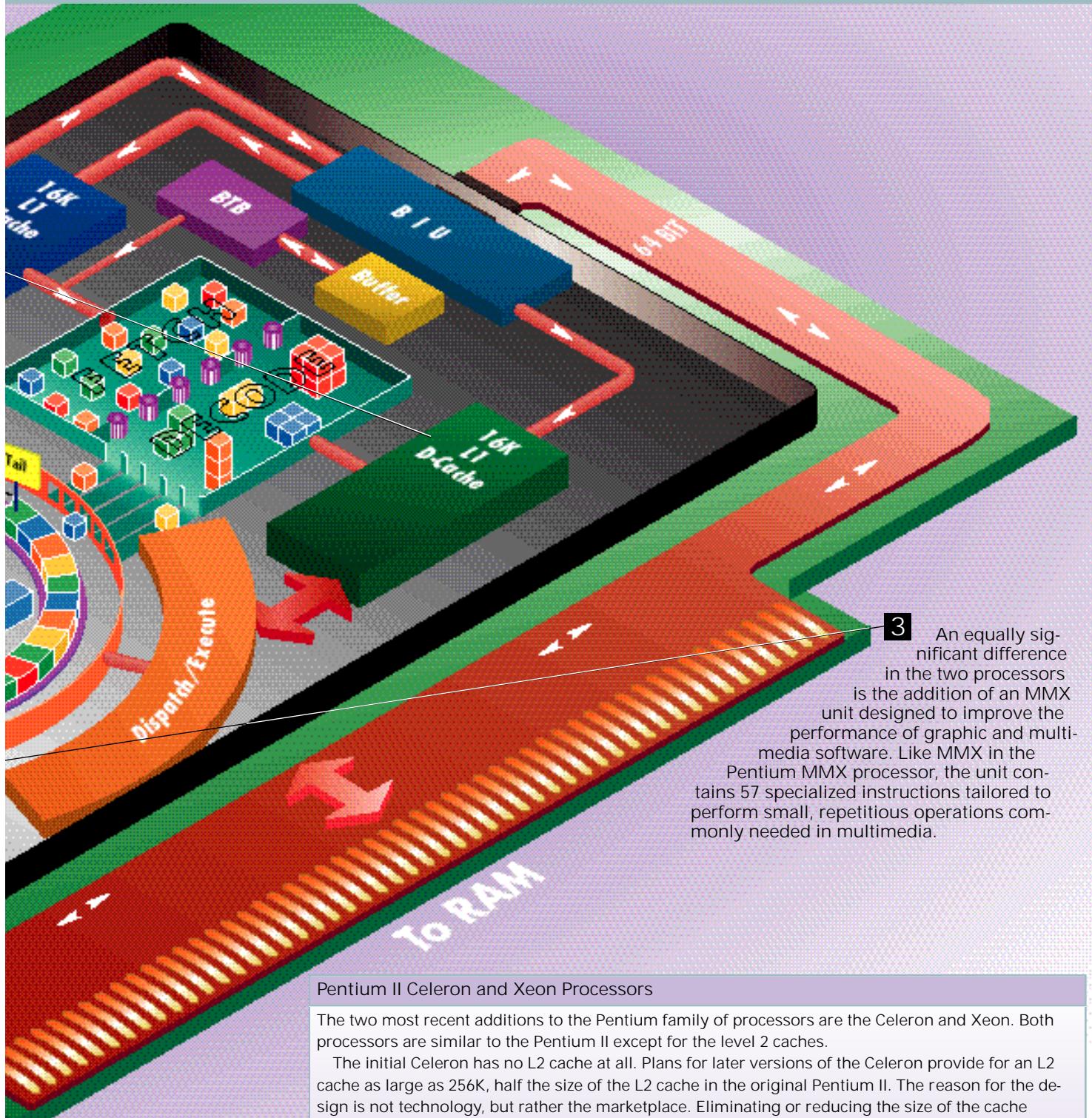
# How a Pentium II Processor Works

**1** The basic structure of Pentium II processor both include the L2 cache. But the two are different in the same way as in the off-the-shelf cache. Their own circuit is built directly on the motherboard, more like an expansion card than a microchip. This design is cheaper to manufacture, but the trade-off is that information flows between the CPU and the L2 cache at half the speed because data travels within the CPU itself.



2 That speed disadvantage is compensated for somewhat by doubling the size of the level 1 (L1) internal cache for data and code from 8K to 16K. The bigger caches cut roughly in half the time it takes to access memory, and provide faster access to the most recently used data and instructions.



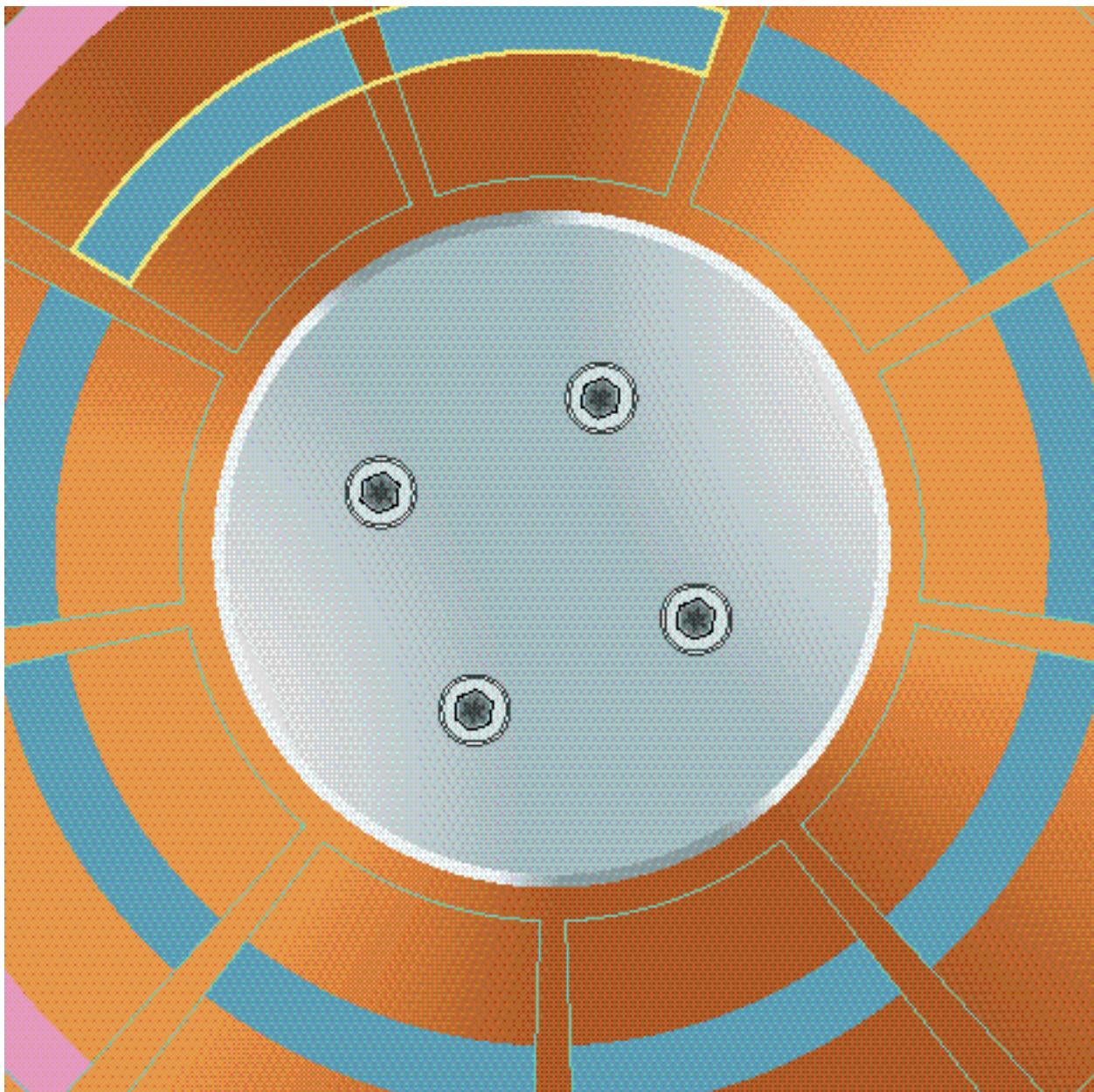


#### Pentium II Celeron and Xeon Processors

The two most recent additions to the Pentium family of processors are the Celeron and Xeon. Both processors are similar to the Pentium II except for the level 2 caches.

The initial Celeron has no L2 cache at all. Plans for later versions of the Celeron provide for an L2 cache as large as 256K, half the size of the L2 cache in the original Pentium II. The reason for the design is not technology, but rather the marketplace. Eliminating or reducing the size of the cache makes the Celeron a less expensive choice for makers of low-end PCs who still want to say their computers have a processor with the same basic design—microarchitecture—as the Pentium II.

The new Pentium II Xeon processor takes the opposite approach. The Xeon includes a larger level 2 cache, either 512K, 1MB, or 2MB. And the data moves between the cache and the processor at whatever speed the processor is running (400MHz+) instead of running at half the speed as in other Pentium II designs. The Xeon is designed for jobs where every bit of speed counts, such as network servers and high-end workstations used, for example, for 3D graphic design or other complex calculations.



P A R T

# 4

## DATA STORAGE

Chapter 12: How Disk Storage Works  
72

Chapter 13: How a Floppy Drive Works  
82

Chapter 14: How a Hard Drive Works  
86

Chapter 15: How Disk Drives Increase Speed and Storage  
94

Chapter 16: How Other Removable Storage Works  
102

Chapter 17: How a Tape Backup Works  
108

AS intelligent and swift as a computer's memory may be, RAM has one fatal flaw. It is a will-o'-the-wisp. With a few exceptions, all memory chips lose the information stored in them once you turn off the computer. All the work you've put into figuring out next year's budget, creating account billings, or writing the great American sitcom will vanish if the electricity constantly stoking the RAM chips' transistors falters for even a fraction of a second.

Fortunately, there are several ways to provide permanent storage for a computer's programs and the work they generate—storage that stays intact even when the power is turned off. The most common form of permanent storage is magnetic disks—both the floppy and hard variety. Magnetic storage is also used in the form of tape drives—a method of permanent storage that's been around almost as long as the first computers. Gaining popularity are optical devices that use lasers to store and retrieve data. And flash memory, nonvolatile chips that, unlike their more common RAM chip cousins, don't lose their contents when you turn off your PC, is becoming increasingly popular for digital cameras and other devices.



Floppy disks are universal, portable, and inexpensive but lack both large capacity and speed, although improvements on the technology, such as the Zip drive and SuperDrive, greatly increase the capacity of removable floppies. Hard disks are, for now, the best all-around storage medium. They store and retrieve data quickly, have the capacity to save several volumes of data, and are inexpensive on a cost-per-megabyte basis. The most recent hard disks are removable and portable, although they don't hold as much data as conventional hard drives. Tape drives provide virtually endless off-line storage at low cost, but they are too slow to use as anything other than a backup medium.

Optical storage serves PC users who need to store enormous quantities of data. CD-ROM drives pack up to 650 megabytes of data on a disc identical to the laser compact discs that play music, and CD-ROM discs are cheap to produce. The newest innovation, DVD (digital versatile disk), stores, theoretically, up to 17 gigabytes of data, the equivalent of a couple of dozen CDs. Both CD and DVD provide both permanent and rewriteable storage, and each has a place in different situations. DVD-RAM has the potential to replace both the floppy and CD-ROM drives. Common CD-ROM and DVD are read-only media, which means that you can only use the data that was stored on them when they were created; you can't erase or change it. Along with magneto-optical and floptical drives, computer users have a choice of cheap, handy solutions to unlimited removable storage with access times fast enough for everyday duty.

Three types of memory chips retain their information when you turn off your computer. EPROMs (for Erasable Programmable Read-Only Memory) are found in every personal computer. They are the chips that store the code and BIOS information needed to boot your PC. (See Part 1.) EPROMs are slow, and their data can be changed only by first exposing them to ultraviolet light. The memory chip that remembers your PC's hardware configuration each time it boots is called the CMOS. That stands for complementary metal oxide semiconductor, a bit of trivia you can safely forget. The CMOS is powered by a separate battery you'll have to change someday if you

keep your PC long enough. Flash RAM chips, which combine the writeability and much of the speed of conventional RAM chips, have their own integrated power supplies and retain data when the main power source is turned off, promise to be in common use in the future and may turn out to be the ideal permanent storage medium. But for now, they are too expensive to completely replace hard disks. You'll find them most often in PC cards (Chapter 25) and digital cameras (Chapter 27).

Despite the different technologies behind these methods of storage, they all have in common a similar notation for recording data and a similar system for filing that information so that it can be found again. Permanent data storage is similar in concept to paper filing systems. Paper files may be handwritten or typed, but they are in the same language. And just as paper files thrown willy-nilly into file cabinets would be impossible to retrieve easily and quickly, electronic files must be stored, too, in an orderly and sensible system and in a common language. In this part of the book, we'll look at how various forms of permanent storage solve the task of saving data so that it can easily be found again, and how different storage devices write and retrieve that data.

## KEY CONCEPTS

**access time** How long it takes to retrieve data from a drive. Access time is made up of seek time (how long it takes for a read/write head to position itself over a particular track), settle time (the time it takes the head to stabilize), and latency (how long it takes for the required sector to rotate until it's under the head). All are measured in milliseconds.



**cluster** One or more successive sectors that contain a contiguous group of data. It is the smallest unit in which data is stored on a drive.

**compression** A process to remove redundant data so that a file is smaller.

**cookie** The Mylar disk to which data is written in a floppy drive.

**data transfer rate** The number of bytes or megabytes transferred from a drive to memory (or from any storage device to another) in a second.

**directory** A set of related files, also called a folder in Windows 95 and 98. The files may be located physically on different parts of a drive, but they are grouped logically in a directory, which can contain other folders or subdirectories.

**drive** Any device for storing computer files.

**drive array** Two or more drives linked to improve file retrieval time and provide error correction.

**format** The process by which a disk is divided into tracks and sectors so that files can be stored and found in an orderly fashion.

**fragmentation** The process by which files become broken up into widely separated clusters.

Defragging or optimization corrects fragmentation by rewriting broken files to contiguous sectors.

**laser** The acronym for Light Amplification by Stimulated Emission of Radiation, it is a device that produces a narrow, coherent beam of light. Coherent means that all the light waves are moving in unison so that the beam does not disperse, or spread and become fainter, as ordinary light does.

**optical storage** Drives that use lasers to store and read data.

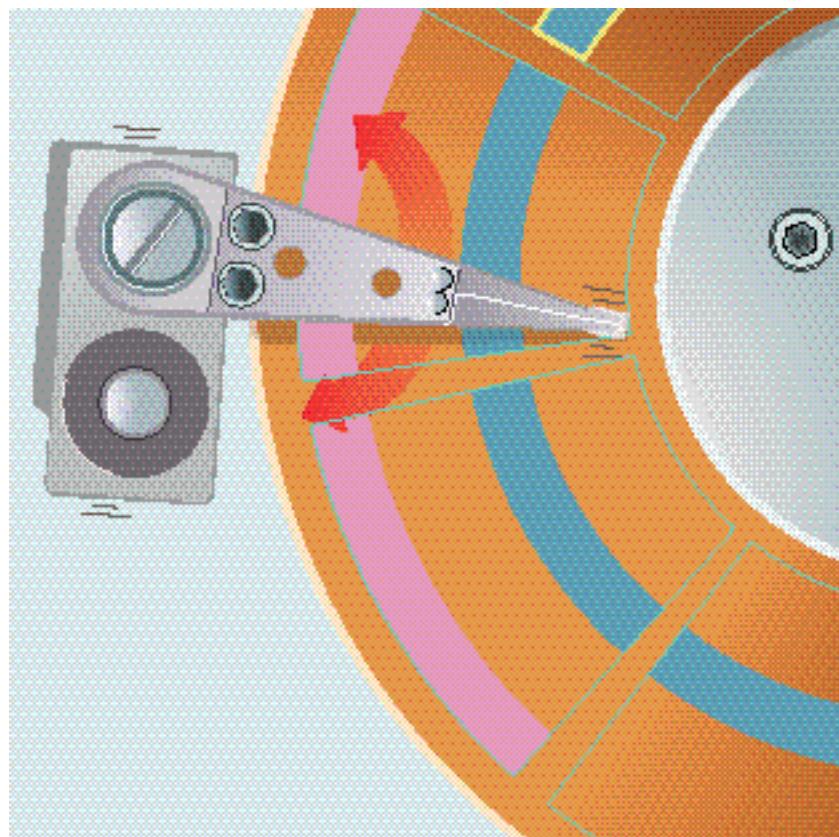
**platter** A rigid, metal disk on which data is stored in a hard drive.

**read/write head** The drive component that writes data to a drive and reads it by using magnetism or a laser.

CHAPTER

# 12

# How Disk Storage Works



**MAGNETIC** disks are the most common form of permanent data storage. Their capacities may range from a few hundred kilobytes to several gigabytes, but they all have some elements in common. For one, the way that a drive's mechanism creates the ones and zeros that make up the binary language of computers may differ, but the goal is the same: to alter microscopically small areas of the disk surface so that some of the areas represent zeros and others represent ones. The disk uses only those two numbers whether it records a great novel or this week's grocery list.

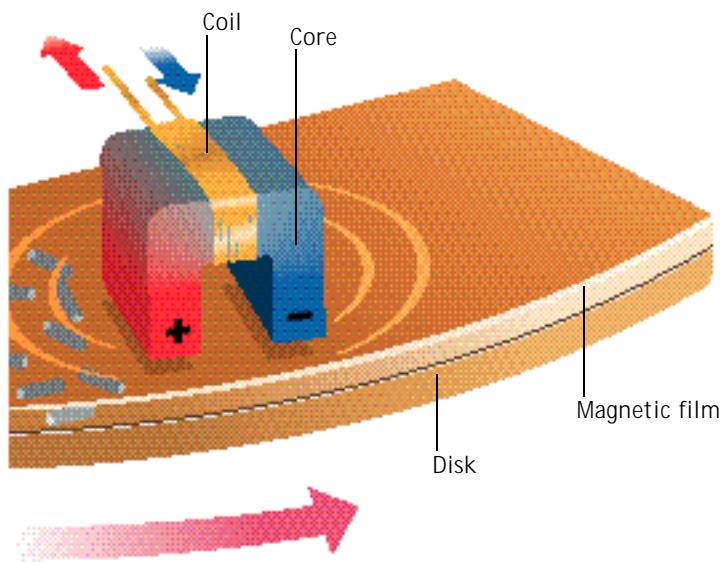
Another common element among magnetic drives is the scheme that determines how the data on the disk is organized. The computer's operating system, which on most personal computers is Windows or MS-DOS, determines the scheme. Even in Windows 98, the older DOS is still there, hidden beneath Windows's graphic interface. The operating system controls so many of a PC's operations that many PC users forget that DOS stands for disk operating system and that, originally, its primary function was to control disk drives.

Before any information can be stored on a magnetic disk, the disk must first be formatted. Formatting creates a road map that allows the drive to store and find data in an orderly manner. The road map consists of magnetic markers embedded in the magnetic film on the surface of the disk. The codes divide the surfaces of the disk into sectors (pie slices) and tracks (concentric circles). These divisions organize the disk so that data can be recorded in a logical manner and accessed quickly by the read/write heads that move back and forth over the disk as it spins. The number of sectors and tracks that fit on a disk determines the disk's capacity to hold information.

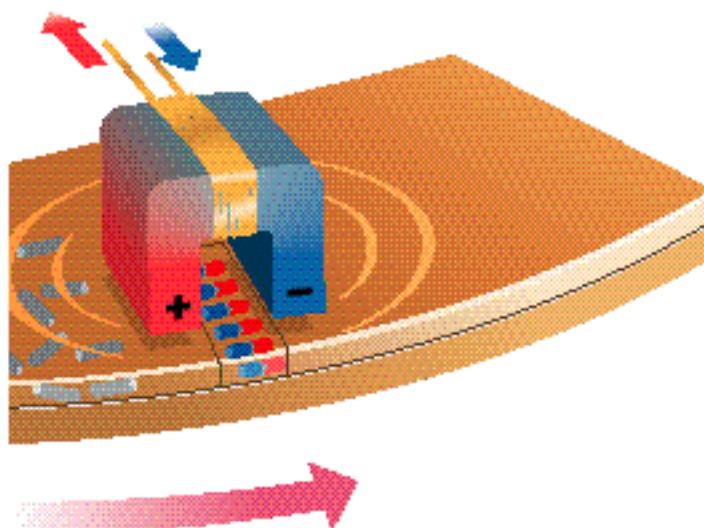
After a disk is formatted, writing or reading even the simplest file is a complicated process. This process involves your software, operating system, the PC's BIOS (basic input/output system), software drivers that tell the operating system how to use add-on hardware such as a SCSI drive or a tape drive, and the mechanism of the disk drive itself.

# Writing and Reading Bits on a Disk

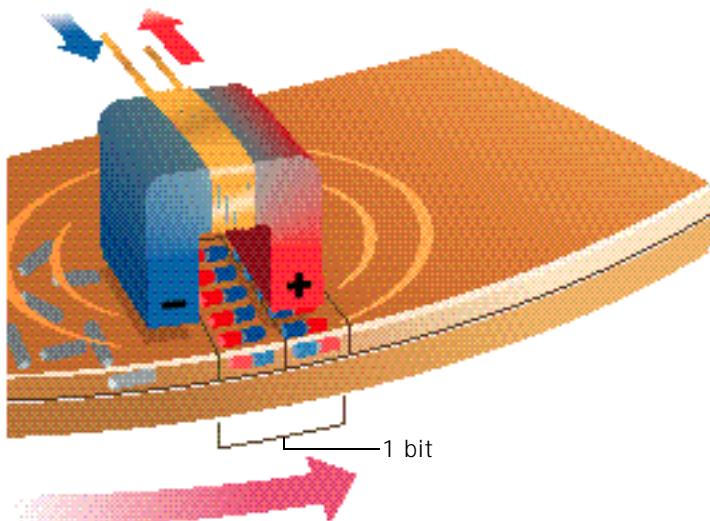
- 1 Before any data is written to a disk, iron particles are scattered in a random pattern within a magnetic film that coats the surface of the disk. The film is similar to the surface of audio and video tapes. To organize the particles into data, electricity pulses through a coil of wire wrapped around an iron core in the drive mechanism's read/write head that is suspended over the disk's surface. The electricity turns the core into an electromagnet that can magnetize the particles in the coating, much as a child uses a magnet to play with iron filings.



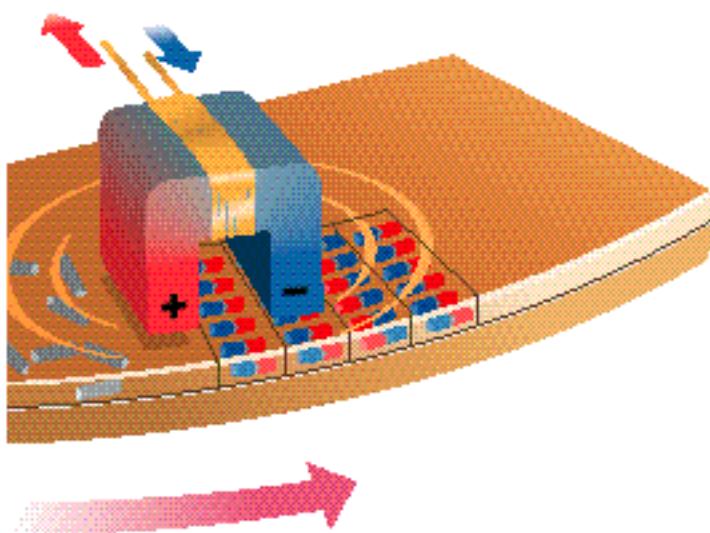
- 2 The coil induces a magnetic field in the core as it passes over the disk. The field, in turn, magnetizes the iron particles in the disk coating so their positive poles (red) point toward the negative pole of the read/write head, and their negative poles (blue) point to the head's positive pole.



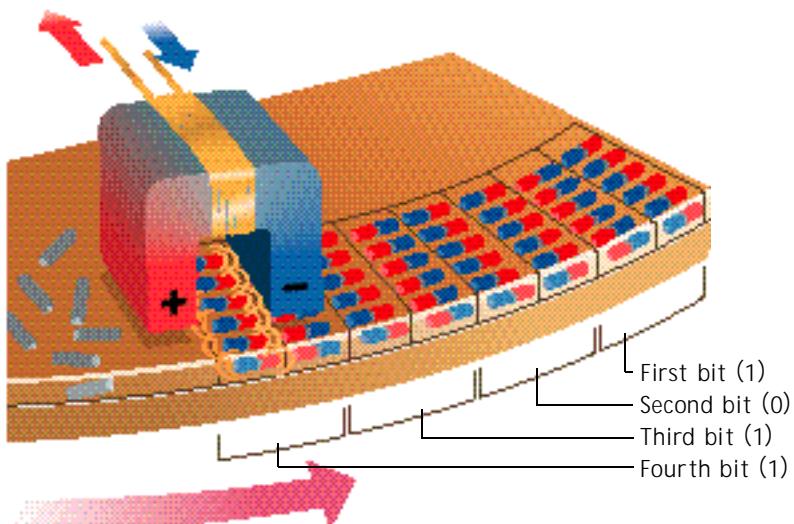
- 3 After the head creates one band of aligned, magnetized particles on the revolving disk, a second band is created next to it. Together, the two bands represent the smallest discrete unit of data that a computer can handle—a bit. If the bit is to represent a binary 1, after creating the first band, the current in the coil reverses so that the magnetic poles of the core are swapped and the particles in the second band are magnetized in the opposite direction. If the bit is a binary 0, the particles in both bands are aligned in the same direction.



- 4 When a second bit is stored, the polarity of its first band is always the opposite of the band preceding it to indicate that it's beginning a new bit. Even the slowest drive takes only a fraction of a second to create each band. The stored bits in the illustration below represent the binary numeral 1011, which is 11 in decimal numbers.



- 5 To read the data, no current is sent to the read/write head as it passes over the disk. Instead, the magnetic opposite of the writing process happens. The banks of polarized particles in the disk's coating are themselves tiny magnets that create a magnetic field through which the read/write head passes. The movement of the head through the magnetic field generates an electrical current that travels in one direction or the other through the wires leading from the head. The direction the current flows depends on the polarities of the bands. By sensing the changes in direction of the current, the computer can tell whether the read/write head is passing over a 1 or a 0.



# Formatting a Disk

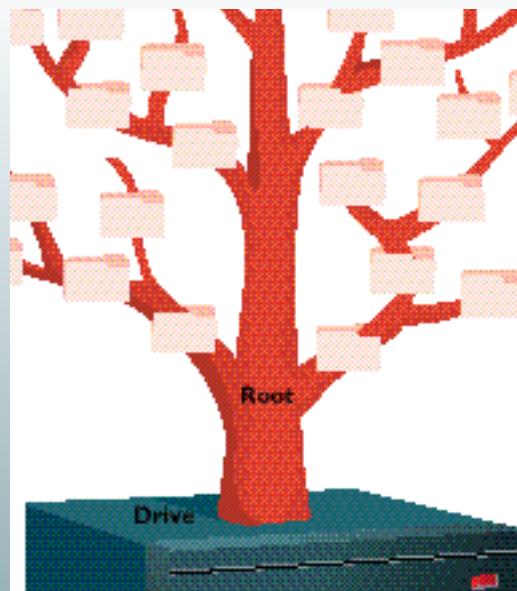
**1** The first task a magnetic drive must accomplish is to format any disk that is used with it so that there will be a way to organize and find files saved to the disk. It does this by writing onto the surface of the disk a pattern of ones and zeros—like magnetic signposts. The pattern divides the disk radially into sectors and into concentric circles called tracks. As the read/write head moves back and forth over the spinning disks, it reads these magnetic signposts to determine where it is in relation to the data on the disk's surface.

**2** Two or more sectors on a single track make up a cluster or block. The number of bytes in a cluster varies according to the version of the operating system used to format the disk and the disk's size. A cluster is the minimum unit the operating system uses to store information. Even if a file has a size of only 1 byte, an entire 256-byte cluster may be used to contain the file. The number of sectors and tracks and, therefore, the number of clusters that a drive can create on a disk's surface determine the capacity of the disk.

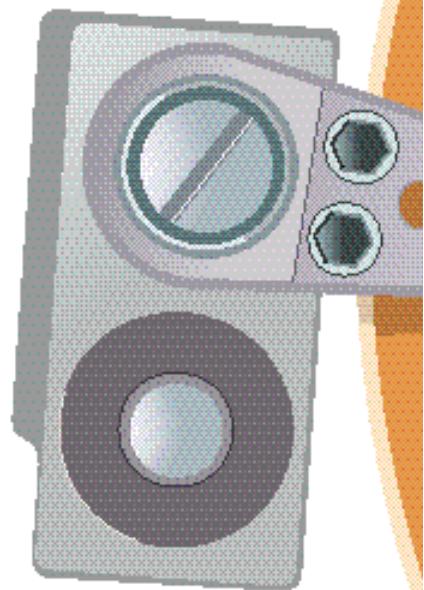
**3** The drive creates a special file located in the disk's sector 0. (In the computer world, numbering often begins with 0 instead of 1.) This file is called the file allocation table, or FAT, in DOS, and the VFAT (virtual FAT) in Windows 95. VFAT is faster because it allows the computer to read files 32 bits at a time, compared to the 16-bit reads of the older FAT. VFAT also allows the use of file names up to 255 characters long, compared to the 11 used by DOS. The FATs are where the operating systems store the information about the disk's directory, or folder structure, and what clusters are used to store which files.

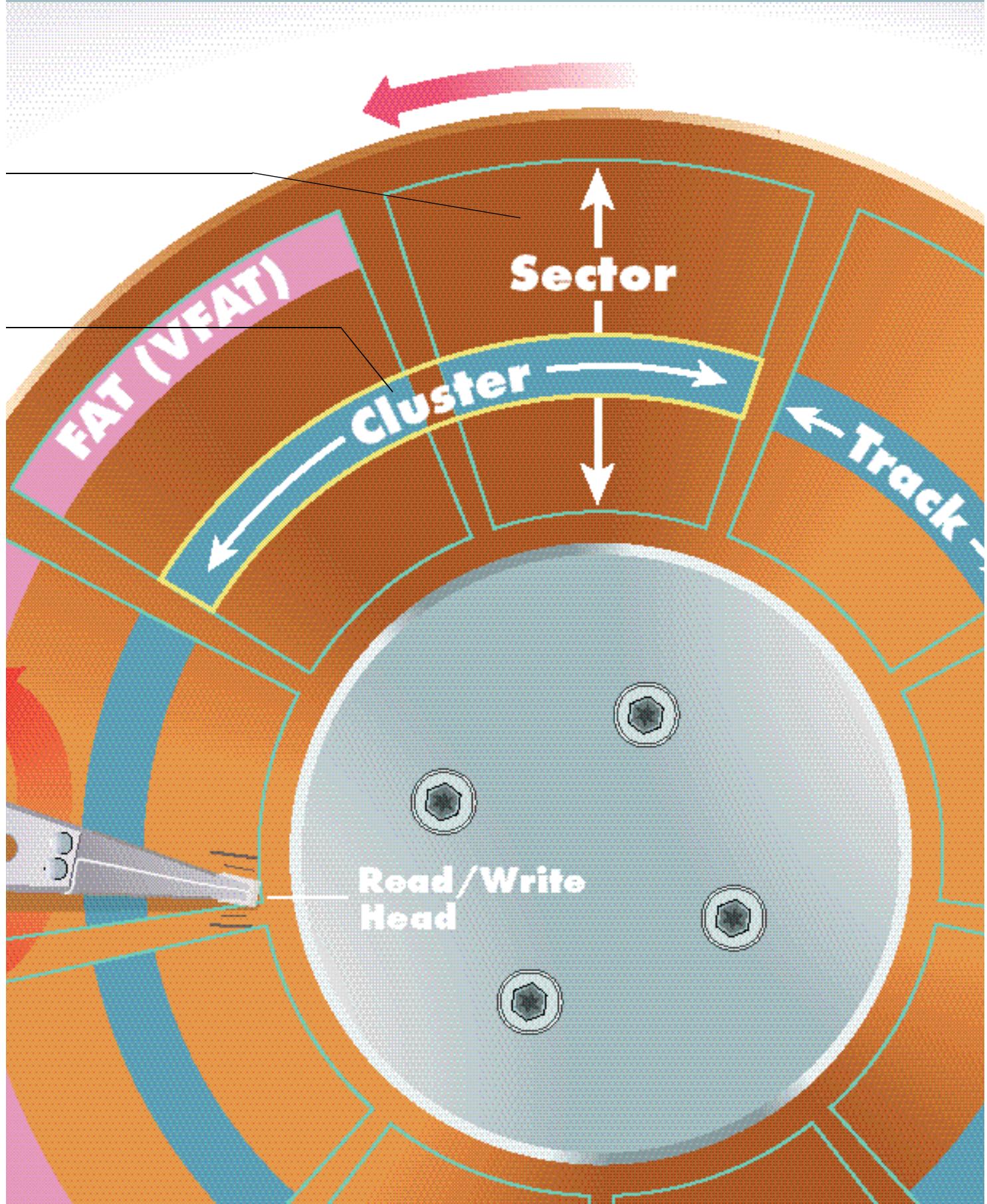
An identical copy of the FAT or VFAT is kept in another location in case the data in the first version becomes corrupted. Ordinarily, you will never see the contents of the FAT or VFAT.

## The Computer Filing Cabinet



Think of a disk as being a filing cabinet in which you keep all your documents. Each drawer in the cabinet is the equivalent of one of your drives—floppy, hard disk, or optical. On each drive, the first level of organization, called the root, contains directories, or folders, the digital equivalent of a file cabinet's cardboard file folders. Each directory contains the individual files—documents, spreadsheets, graphics, programs—the same way that a drawer's file folders contain individual letters, reports, and other hard copy. One important difference is that it's easy for drive folders to contain other folders, which can contain still more folders, and so on indefinitely. This directory/folder structure is called a tree because a diagram of how it's organized looks like the branching structure of a tree.





# Writing a File to Disk

- 1** When you click your mouse to save a file, the program you're using sends a command to Windows 95, asking the operating system to carry out the steps needed to save the file from RAM, where it's being held temporarily, to disk for permanent storage. For this example, we'll assume you're using a word processor to save a file named Letter to Mom.doc.

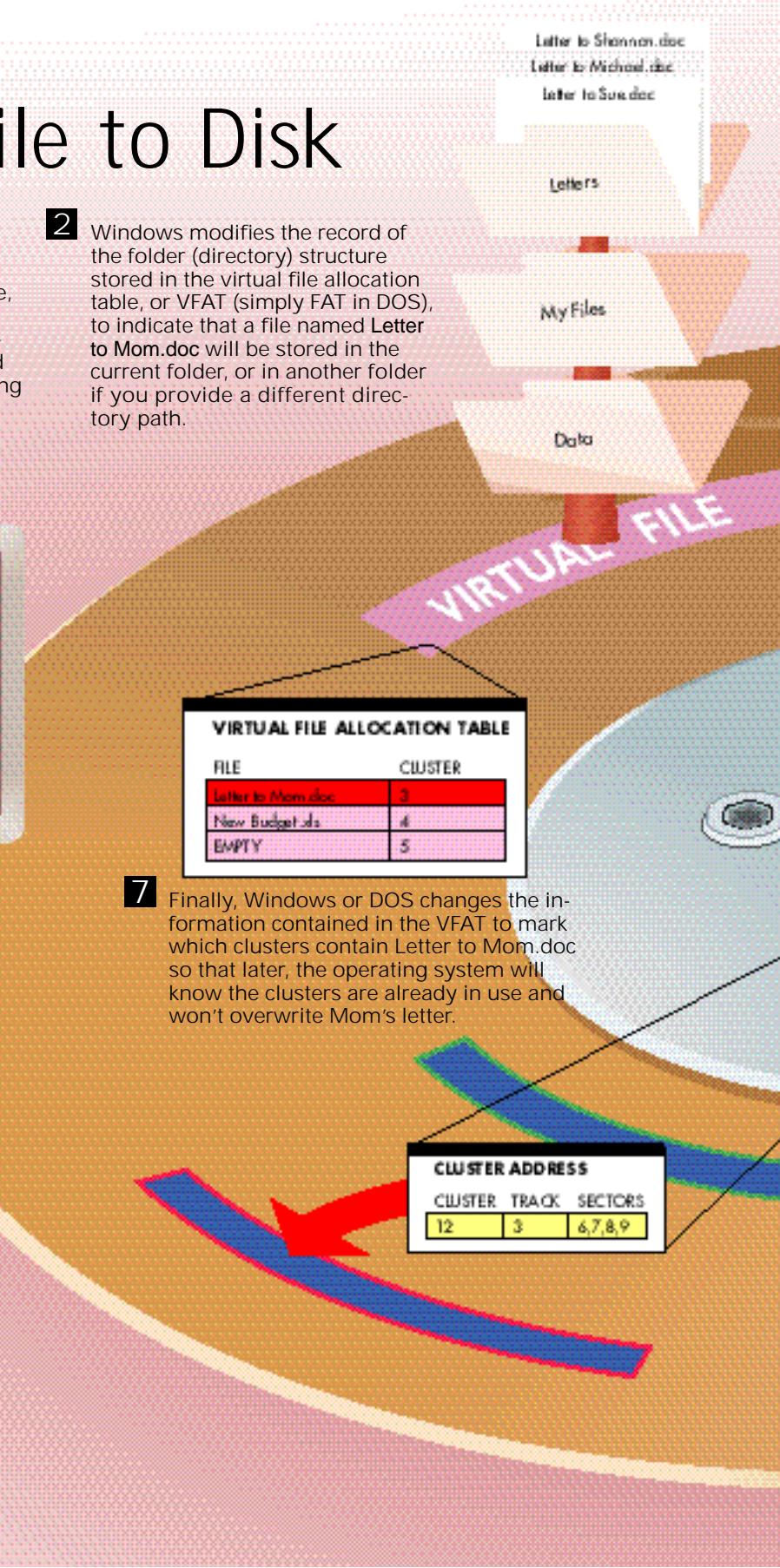


## What Happens When You Delete a File?

When you delete a file, the data that makes up the file is not actually changed on the disk. Instead, the operating system changes the information in the VFAT to indicate that the clusters that had been used by that file are now available for reuse by other files. Because the data remains on disk until the clusters are reused, you can often restore—or undelete—a file that you've accidentally erased.

VIRTUAL FILE ALLOCATION TABLE		
FILE	Available	CLUSTER
Letter to Mom.doc	Y	3
New Budget.xls	Y	4
EMPTY	Y	5
Old Budget.xls	Y	6

- 2** Windows modifies the record of the folder (directory) structure stored in the virtual file allocation table, or VFAT (simply FAT in DOS), to indicate that a file named Letter to Mom.doc will be stored in the current folder, or in another folder if you provide a different directory path.



- 3 The operating system also checks the VFAT for the number of a cluster where the VFAT says Windows can save the file without overwriting any other data that's already been saved. In this example, the VFAT tells Windows that Cluster 3 is available to record data.

VIRTUAL FILE ALLOCATION TABLE	
FILE	1ST CLUSTER
Expenses.xls	1
Annual Report.doc	2
AVAILABLE	3
New Budget.xls	4

CLUSTER ADDRESS		
CLUSTER	TRACK	SECTORS
3	1	2,3,4,5

- 4 From the VFAT, the operating system also determines that the location of Cluster 3 comprises Sectors 2, 3, 4, and 5 on Track 1. Windows sends this information to the PC's BIOS (basic input/output system).

**ALLOCATION TABLE**

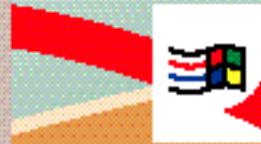
VIRTUAL FILE ALLOCATION TABLE	
FILE	1ST CLUSTER
AVAILABLE	12
AVAILABLE	13
Memo to Boss.doc	14



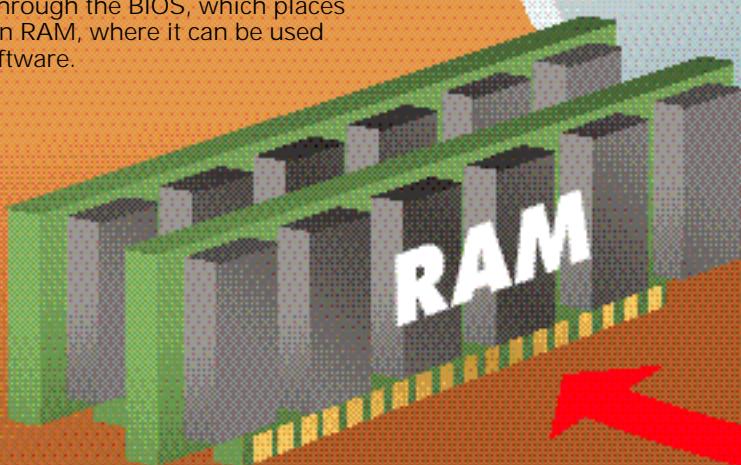
- 5 The BIOS frees the software and operating system from the details of saving the file. It retrieves the data that will make up Letter to Mom.doc from where the word processor is using it in RAM. At the same time, it issues the instructions to the disk drive controller to save the data that the BIOS is sending it, beginning at Sectors 2 through 5 on Track 1.

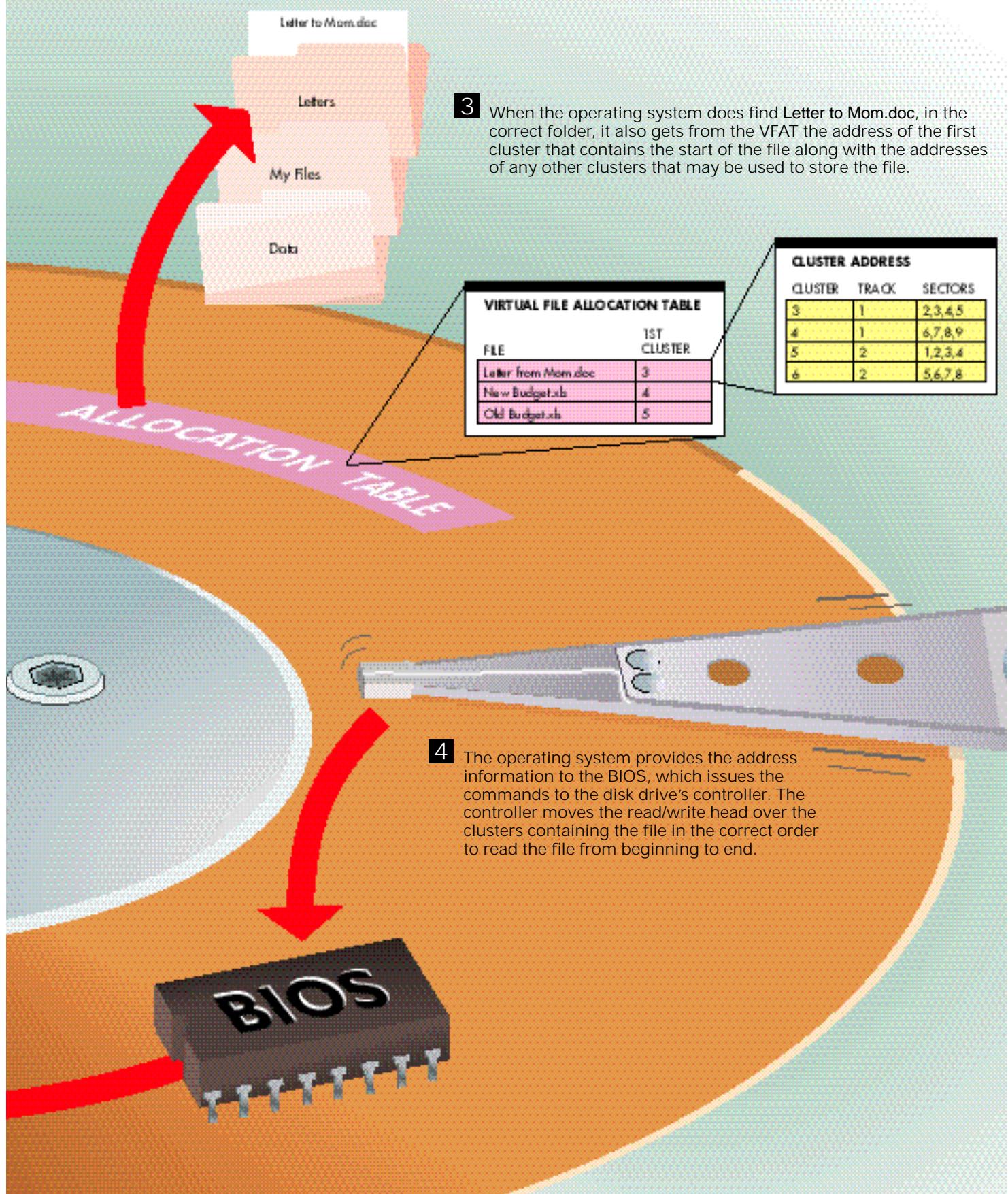
- 6 If the file is larger than the number of bytes contained in a single cluster, the operating system asks the VFAT for the location of another cluster where it may continue saving the file. The clusters need not be adjacent to each other on the disk. The VFAT maintains a record of the chain of clusters over which the file is spread. The process of moving data from RAM to disks repeats itself until the operating system encounters a special code called an end of file marker.

# Reading a File from a Disk



- 1 When you use, for example, your word processor to retrieve the file Letter to Mom.doc, your software passes along the request and the name of the file to the operating system, such as Windows or DOS.
- 2 Windows checks the VFAT (FAT for DOS) for the information on the current folder to see if a file named Letter to Mom.doc is in the current folder. The current folder can be one of many folders or directories on the disk. Each program usually remembers the folder where you last opened or saved a file using that piece of software. It assumes you'll want to use the same folder and opens an Open File dialog box in the most recently used folder for that application. Because in Windows you usually choose a filename from a list to open it, the file is, of course, there. But if you choose to type in the name of a file and make a mistake, or if someone else on a network with you had deleted the file moments before, Windows doesn't find the file listed in the VFAT. Then, the application displays a message telling you that it can't find the file.
- 5 The drive sends the data it reads from the disk through the BIOS, which places the data in RAM, where it can be used by the software.

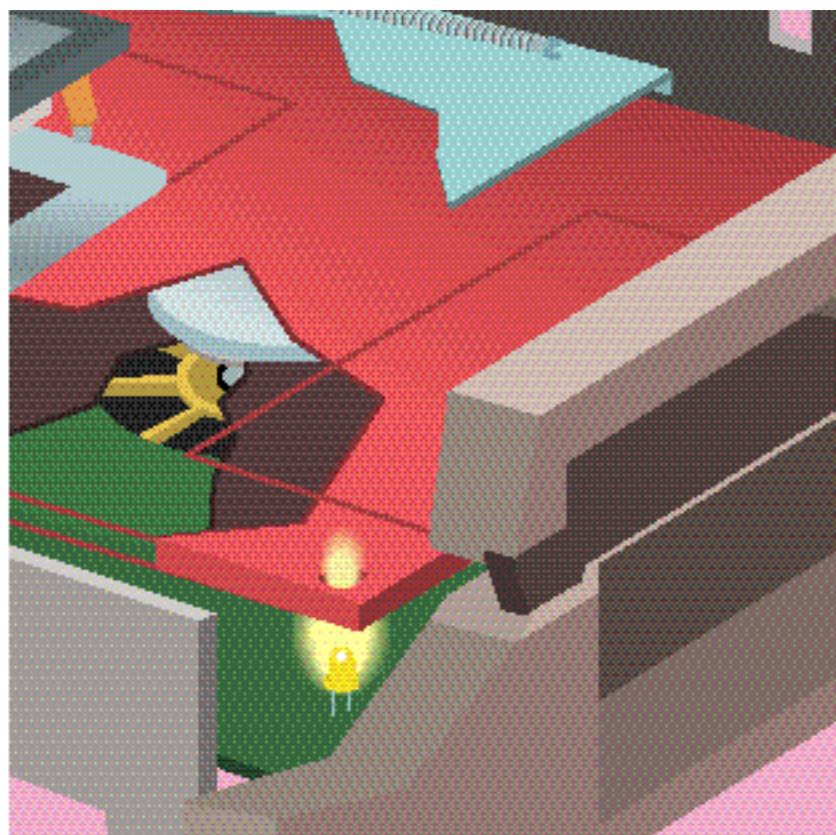




CHAPTER

13

# How a Floppy Drive Works



**AMID** superfast, superbig hard drives, magneto-optical drives, CD-ROM drives, DVD drives, and all the other new high-tech marvels, it's hard to get excited about the common floppy drive. After all, it's slow and a floppy disk doesn't store much information compared to... well, compared to any other type of disk. When the size of software is measured in tens of gigabytes, it's the rare program these days that's still distributed on floppy instead of a CD.

But for all its deficiencies, the floppy drive is an underappreciated wonder. Think of it: An entire book full of information can be contained on a disk that you can slip into your pocket. Floppy drives are ubiquitous, making them a sure and convenient way to get data from one PC to another. No communication lines, networks, gateways, or infrared links are needed; just pull the floppy out of one machine and slip it into another.

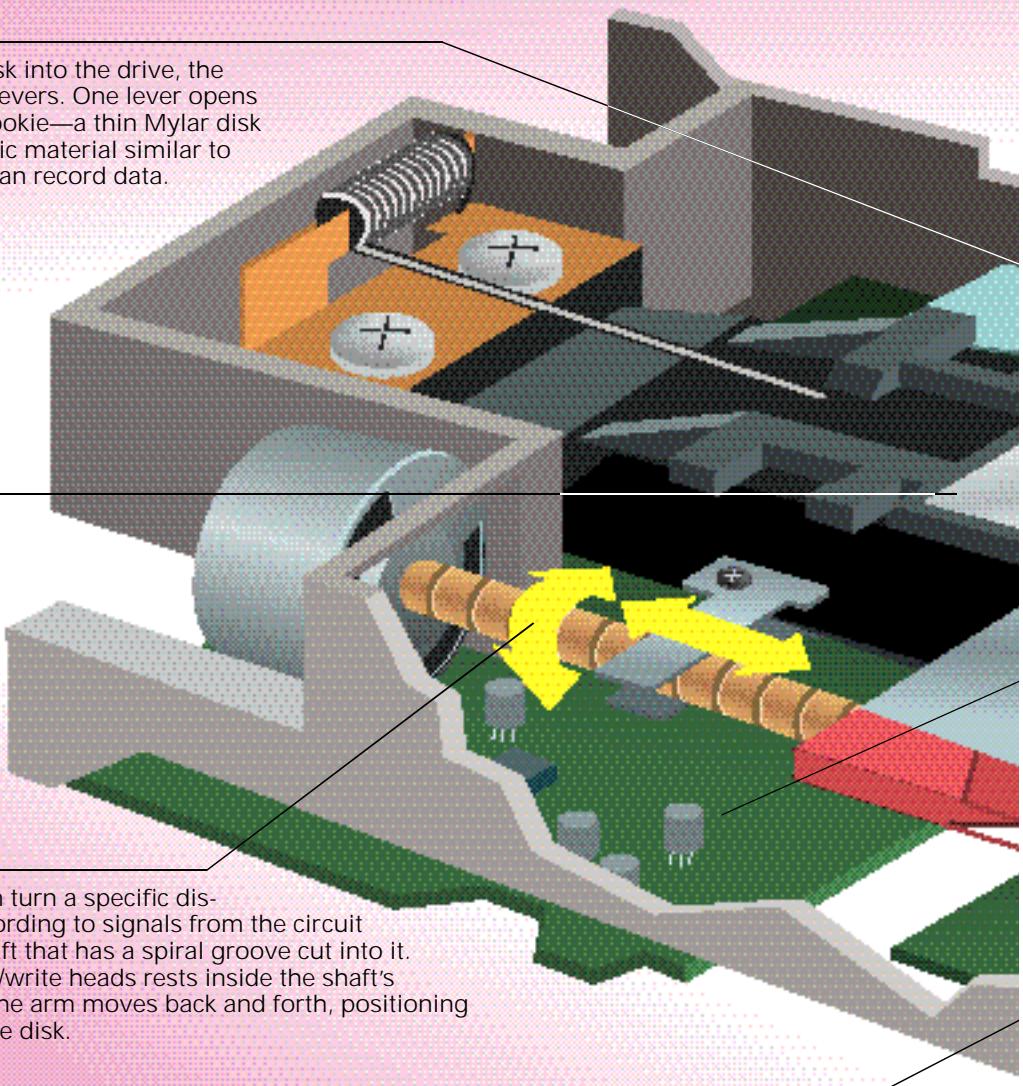
But given the gargantuan size of Windows and its applications, most programs are now distributed on CD-ROM. And the floppy is challenged by overachieving cousins, such as the ZIP drive. Still, for all its commoner heritage, the floppy is cheap and dependable and respectable. It will be with us in some form for a long time to come.

Although smaller, faster, and more capacious floppy drives are now standard components of all new computers, it took years for them to supplant the old 5.25-inch floppy drive. That early drive was the 78-rpm phonograph record of the computer world. Long after smaller records that played more music with greater fidelity were available, phonograph companies continued to produce turntables with 78-rpm settings just because many music lovers had so much invested in 78s. When the first edition of this book was published in 1993, it was common for PCs to be sold with both the 5.25-inch and newer 3.5-inch drives. Today, a 5.25-inch drive is an artifact of forgotten, primitive times.

With capacities today ranging from 700 kilobytes to 2.88 megabytes, 3.5-inch disks hold more data than their older cousins. Their protective cases mean that we can be downright careless about how we handle them, and they are so cheap that their cost is not a factor. And they are so entrenched that it will be years before they're supplanted by the writeable compact discs and DVDs that are just emerging.

# 3.5-Inch Floppy Drive

- 1 When you push a 3.5-inch floppy disk into the drive, the floppy presses against a system of levers. One lever opens the floppy's shutter to expose the cookie—a thin Mylar disk coated on either side with a magnetic material similar to the coating on a cassette tape that can record data.

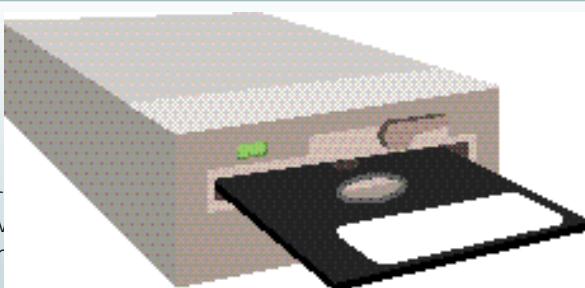


- 7 When the heads are in the correct position, electrical impulses create a magnetic field in one of the heads to record data to either the top or bottom surface of the disk. When the heads are reading data, they react to magnetic fields generated by the metallic particles on the disk by sending electrical signals to the computer.

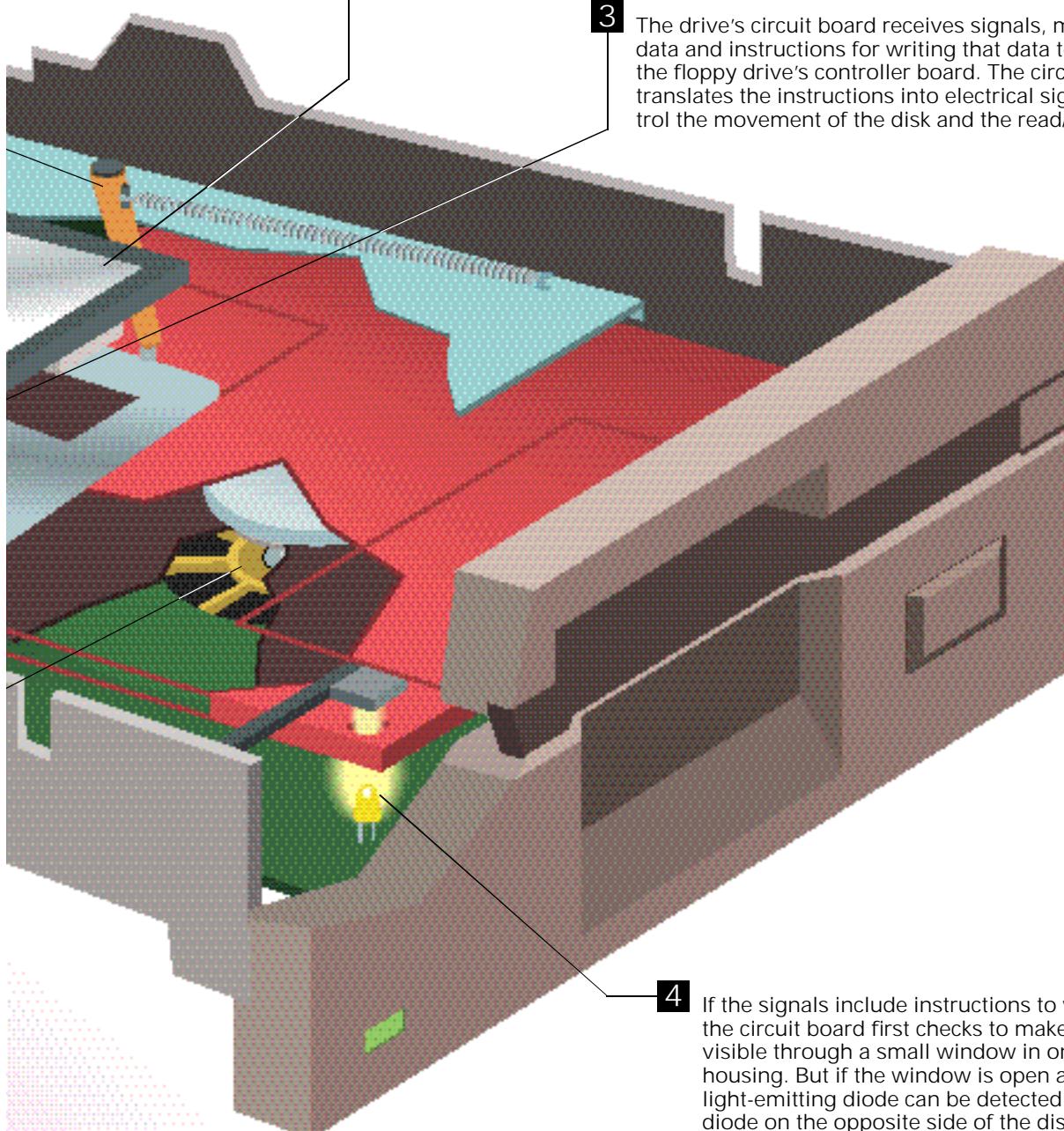
- 6 A stepper motor—which can turn a specific distance in either direction according to signals from the circuit board—moves a second shaft that has a spiral groove cut into it. An arm attached to the read/write heads rests inside the shaft's groove. As the shaft turns, the arm moves back and forth, positioning the read/write heads over the disk.

## The 5.25-Inch Dinosaur

Despite its different size and casing, the 5.25-inch floppy disk is simply a bigger, slower, less complicated version of the 3.5-inch disk. It has no door to open, but a notch on its side is checked for write protection, and the read/write heads of the drive work the same way as those of the smaller drive do.



- 5 A motor located beneath the disk spins a shaft that engages a notch on the hub of the disk, causing the disk to spin.



**2** Other levers and gears move two read/write heads until they barely touch the cookie on either side. The heads, using tiny electromagnets, generate magnetic pulses that change the polarity of metallic particles embedded in the disk's coating. (See Chapter 12.)

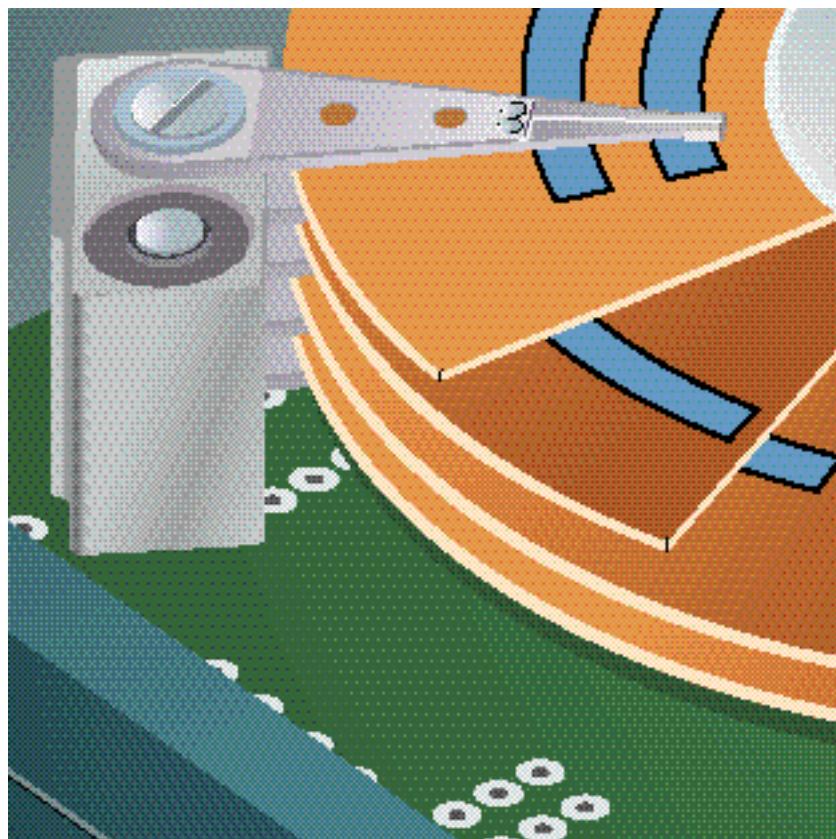
**3** The drive's circuit board receives signals, made up of data and instructions for writing that data to disk, from the floppy drive's controller board. The circuit board translates the instructions into electrical signals that control the movement of the disk and the read/write heads.

**4** If the signals include instructions to write data to the disk, the circuit board first checks to make sure that no light is visible through a small window in one corner of the disk's housing. But if the window is open and a beam from a light-emitting diode can be detected by a photo-sensitive diode on the opposite side of the disk, the drive knows the disk is write-protected and refuses to record new data.

CHAPTER

14

# How a Hard Drive Works



A hard drive is the workaholic of a PC system. The platters on which data is stored spin at speeds of up to 7,200 revolutions a minute—120 spins each second. Each time the hard drive is accessed to read or save a file it causes the read/write heads to burst into a flurry of movement that must be performed with microscopic precision. So exacting are the tolerances in a hard drive—the gaps between the heads and the platters aren't big enough to admit a human hair—that it's a wonder the drive can perform its work at all without constant disasters. Instead, it keeps on plugging away as the repository of perhaps years of work—with surprisingly few failures.

The capacity, size, and performance of hard drives has changed dramatically since the introduction in the early 1980s of the first IBM XT with a hard drive. Back then, a capacity of 10 megabytes was considered generous. The hard drive was 3 to 4 inches thick and filled a 5.25-inch drive bay. An access time of 87 milliseconds was warp speed compared to the access times of floppy drives. A decade later, hard drives that hold 6-9 gigabytes (6,000-9,000 megabytes), smaller than a 3.5-inch floppy drive, and with access speeds of 8 milliseconds are inexpensive and commonplace. Some hard drives pack hundreds of megabytes on removable disks no larger than a matchbox. In the future, the size and prices of drives will continue to decrease while their capacities increase. Who says you can't have a win-win situation?

The price of hard storage is becoming so inexpensive that synchronized groups of hard drives, called drive arrays, are becoming affordable for small businesses and home offices that want the benefits of speed and dependability that the arrays provide.

One thing about hard drives will stay the same. Unlike other PC components that obey the commands of software without complaint, the hard drive chatters and groans as it goes about its job. Those noises are reminders that the hard drive is one of the few components of a personal computer that is mechanical as well as electronic. The drive's mechanical components make it the slowest component in your PC. But they also mean that the hard drive, in more ways than one, is where the action is.

# Hard Disk Drive

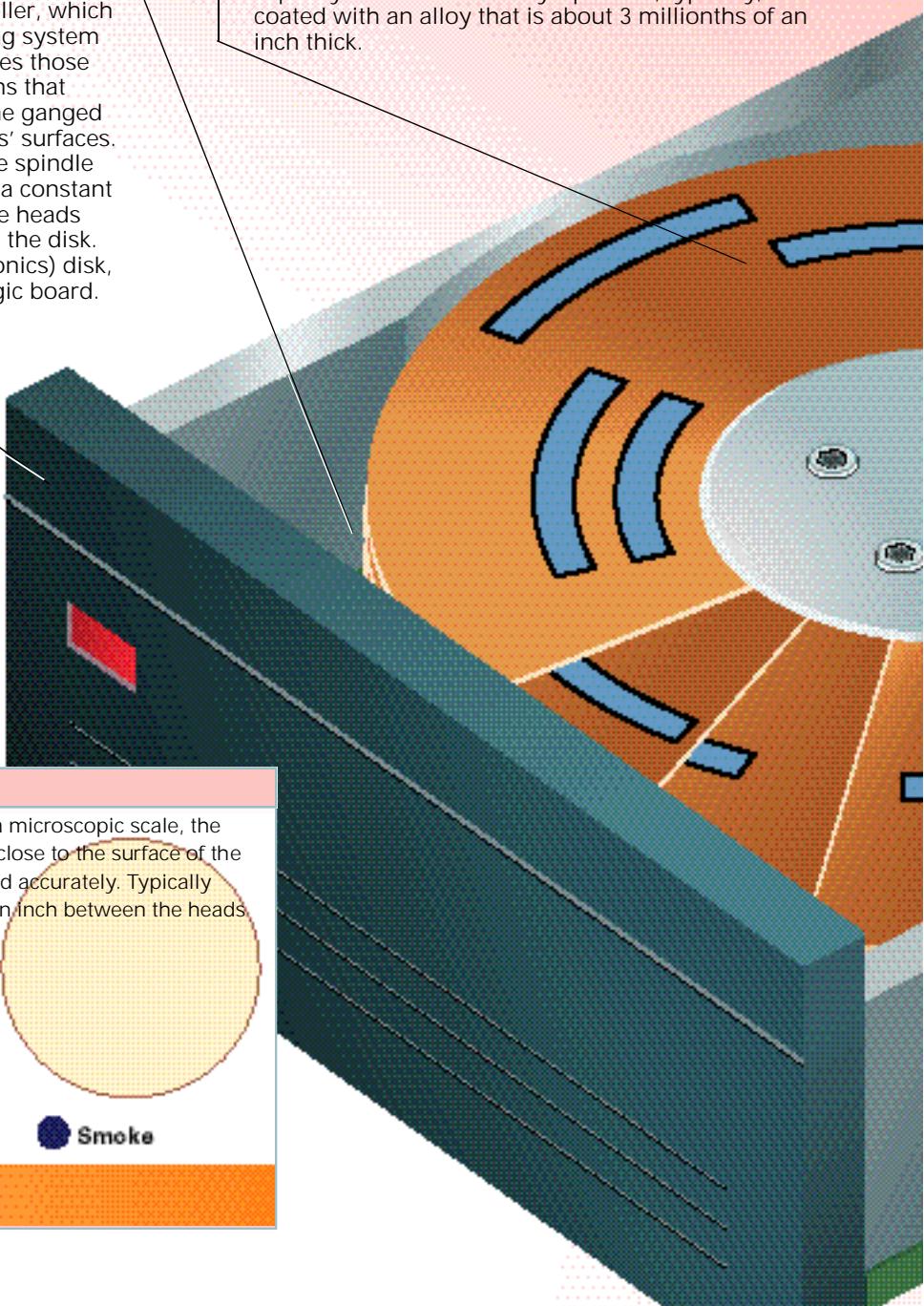
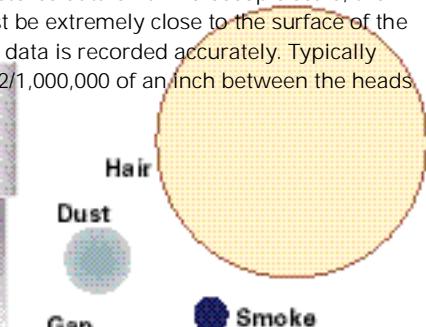
**2** On the bottom of the drive, a printed circuit board, also known as a logic board, receives commands from the drive's controller, which in turn is managed by the operating system and BIOS. The logic board translates those commands into voltage fluctuations that force the head actuator to move the ganged read/write heads across the platters' surfaces. The board also makes sure that the spindle turning the platters is revolving at a constant speed, and the board tells the drive heads when to read and when to write to the disk. On an IDE (Integrated Drive Electronics) disk, the disk controller is part of the logic board.

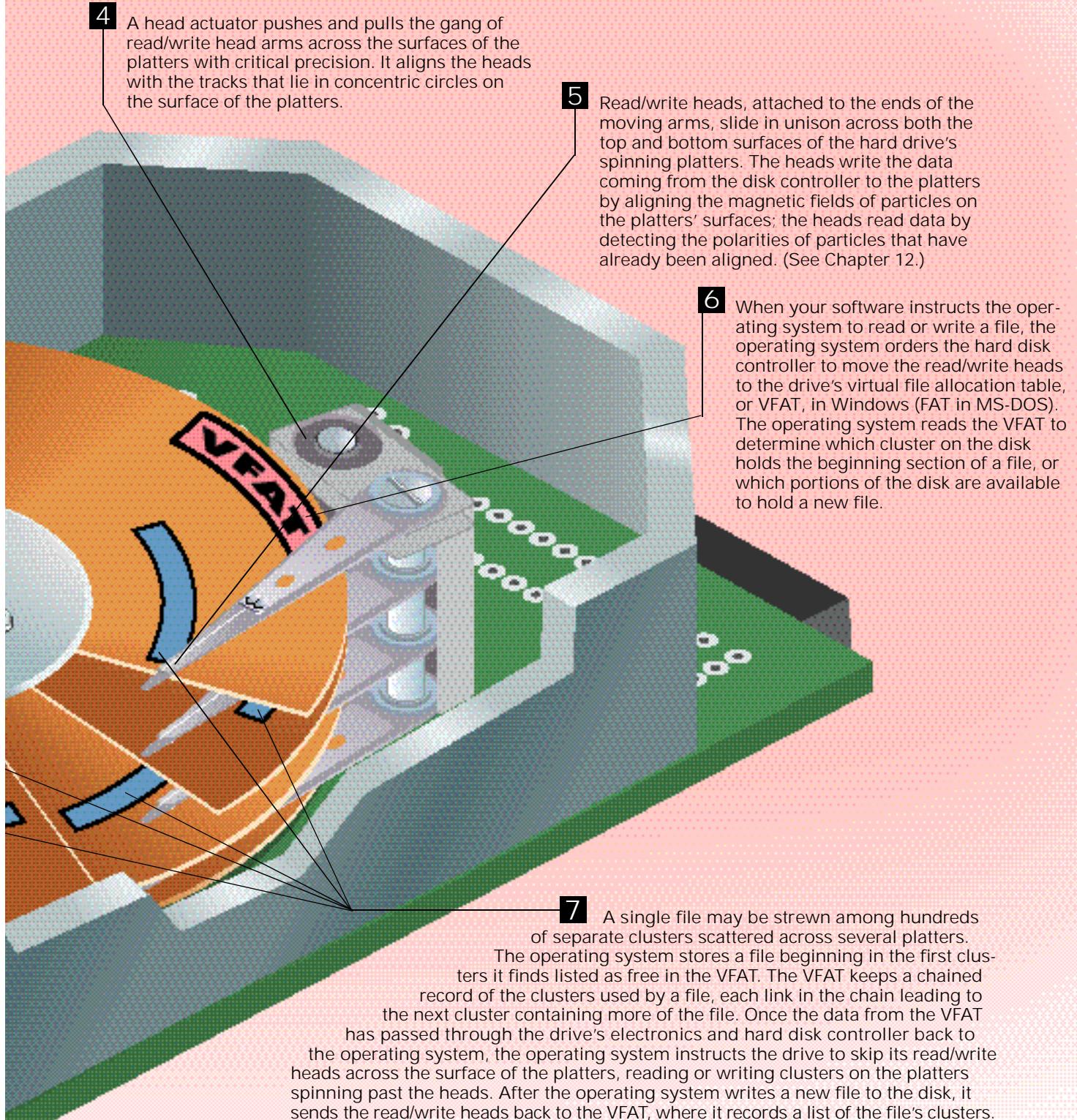
**3** A spindle connected to an electrical motor spins as many as eight magnetically coated platters (made of metal or glass) at several thousand revolutions per minute. The number of platters and the composition of the magnetic material coating them determine the capacity of the drive. Today's platters, typically, are coated with an alloy that is about 3 millionths of an inch thick.

**1** A sealed metal housing protects the hard drive's internal components from dust particles that could block the narrow gap between the read/write heads and the platters. An obstruction there causes the drive to crash, literally, by plowing a furrow in a platter's thin magnetic coating.

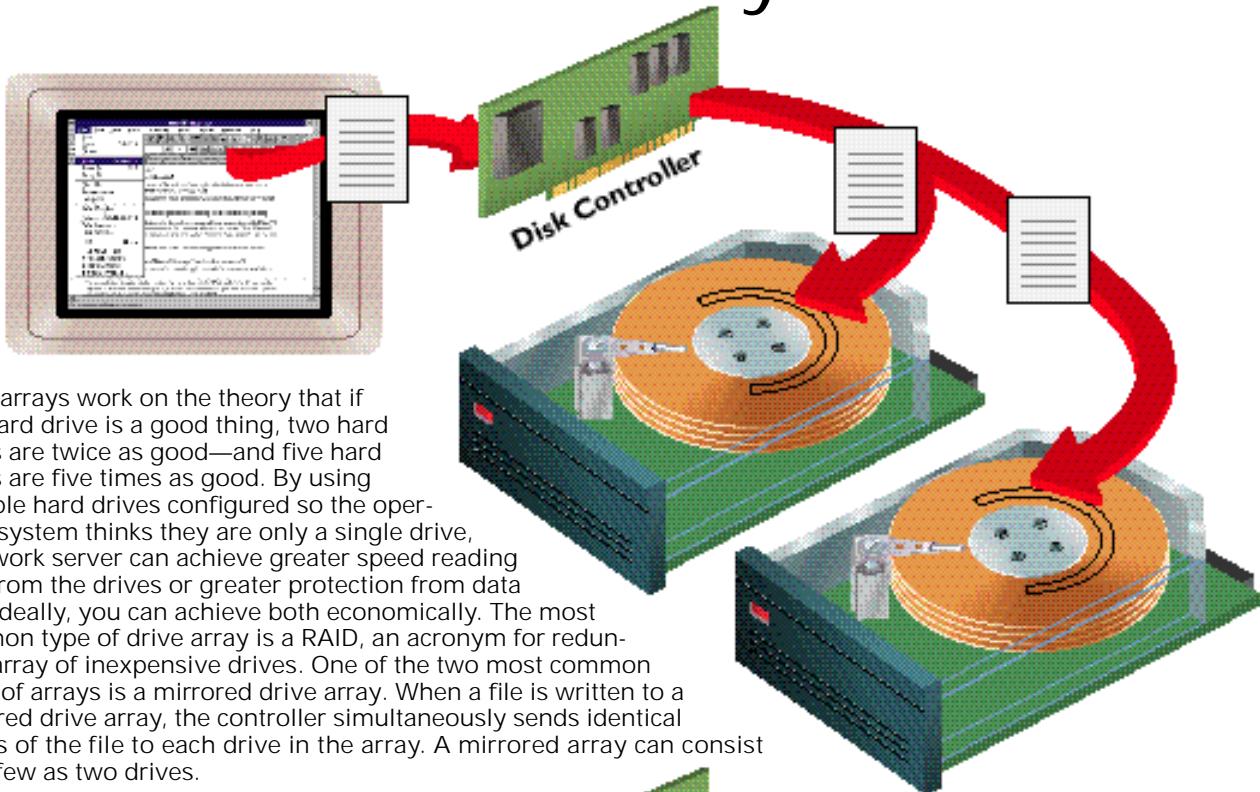
## No Room for Error

Because a hard drive stores data on a microscopic scale, the read/write heads must be extremely close to the surface of the platters to ensure that data is recorded accurately. Typically there is a gap of only  $2/1,000,000$  of an inch between the heads and the disk surface.

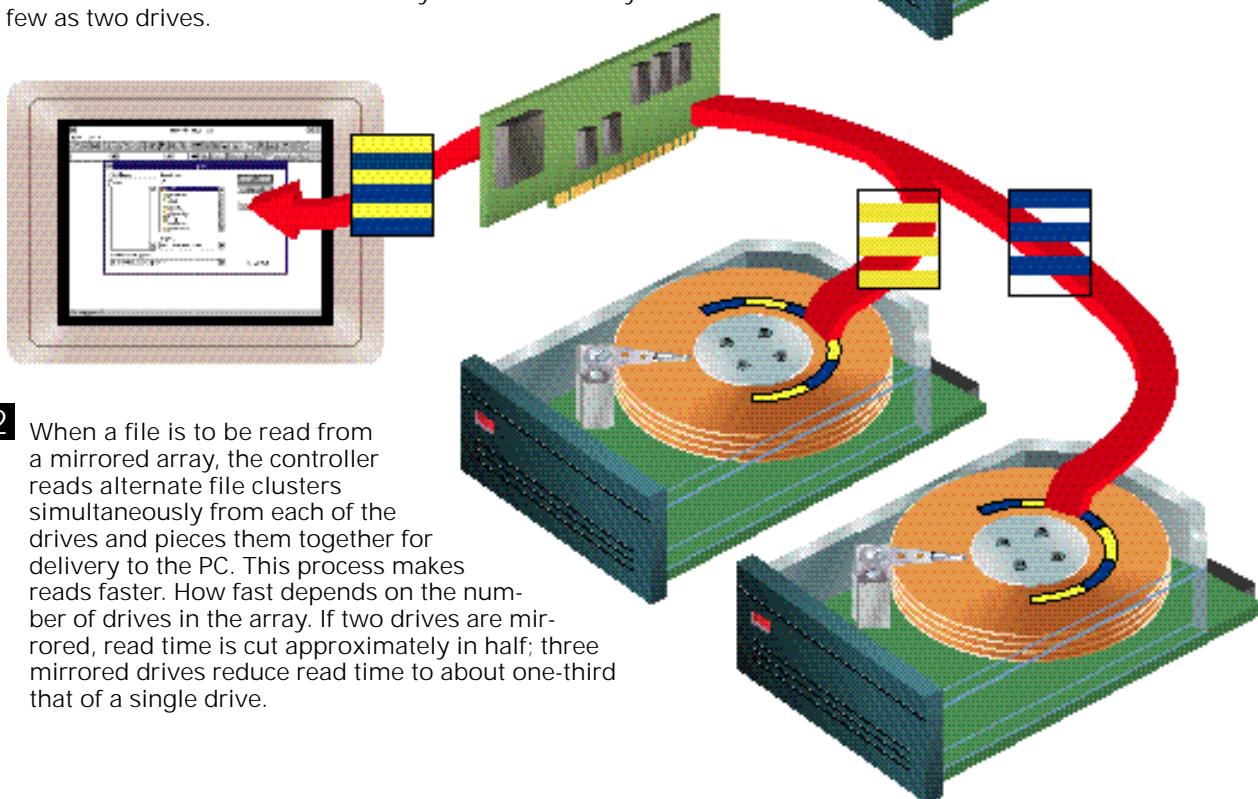




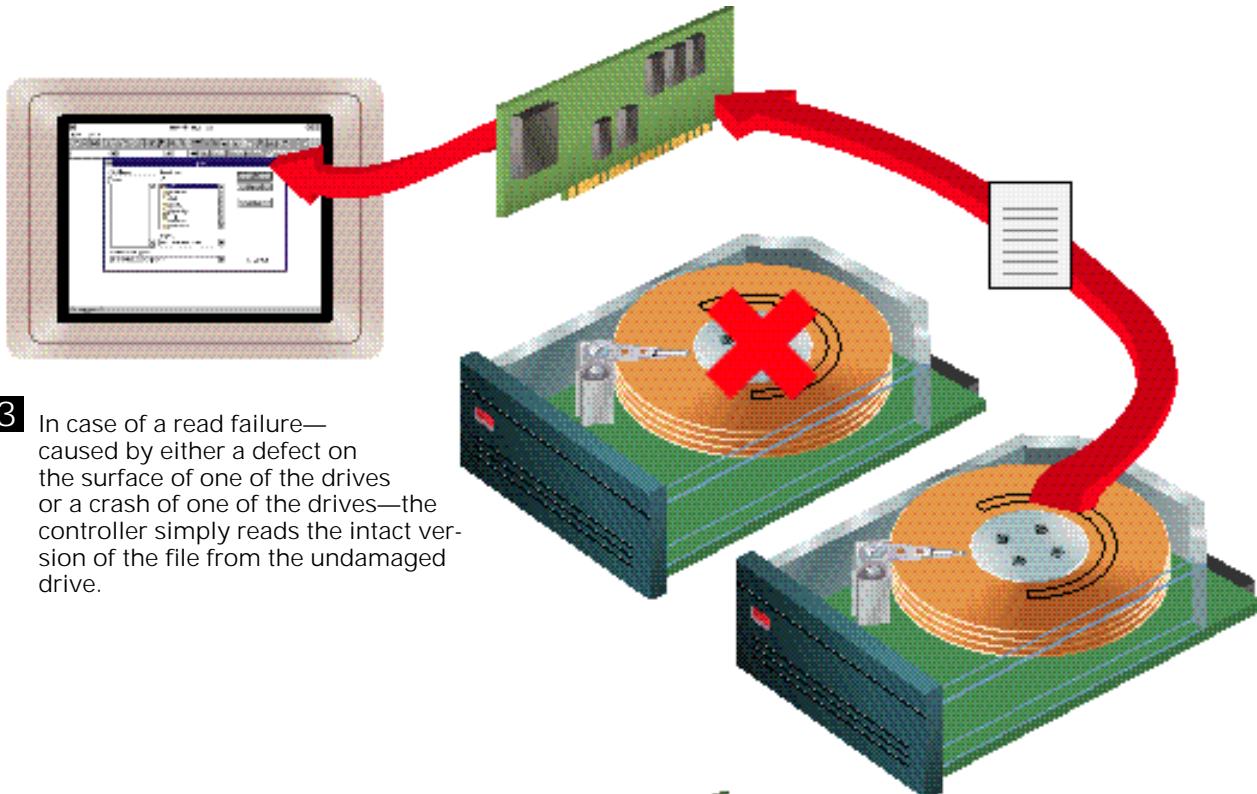
# Mirrored Drive Array



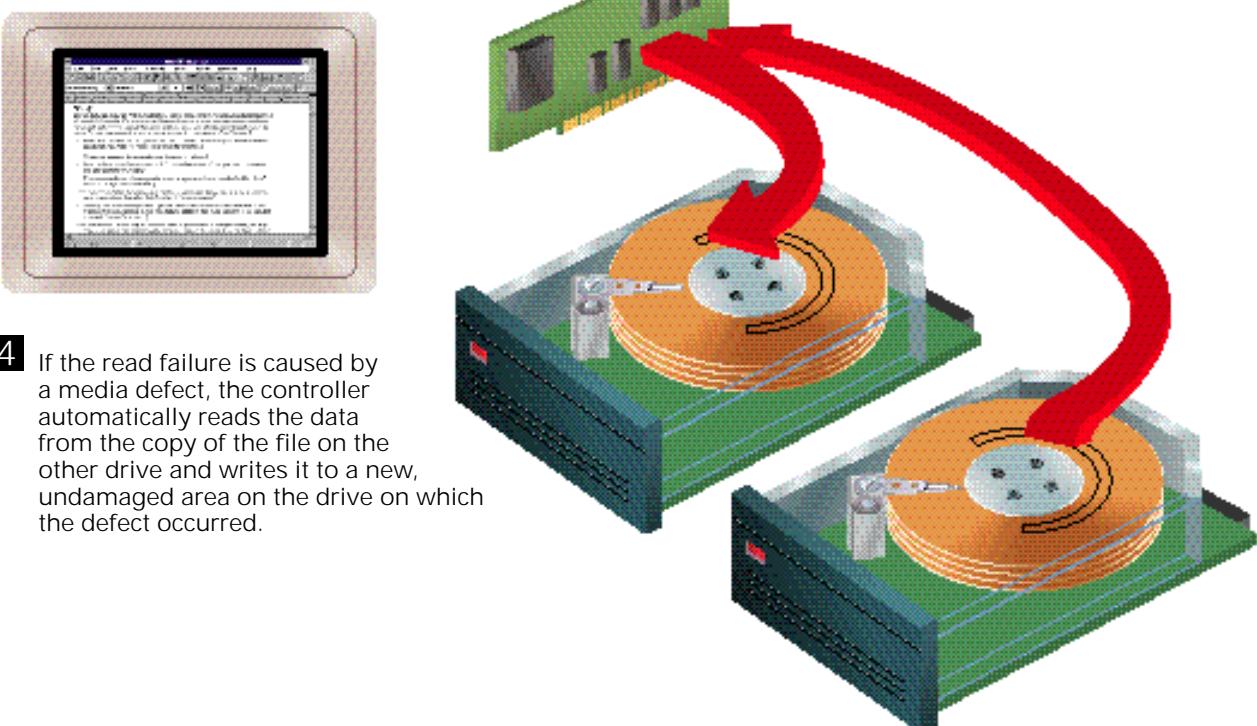
- 1 Drive arrays work on the theory that if one hard drive is a good thing, two hard drives are twice as good—and five hard drives are five times as good. By using multiple hard drives configured so the operating system thinks they are only a single drive, a network server can achieve greater speed reading data from the drives or greater protection from data loss. Ideally, you can achieve both economically. The most common type of drive array is a RAID, an acronym for redundant array of inexpensive drives. One of the two most common types of arrays is a mirrored drive array. When a file is written to a mirrored drive array, the controller simultaneously sends identical copies of the file to each drive in the array. A mirrored array can consist of as few as two drives.



- 2 When a file is to be read from a mirrored array, the controller reads alternate file clusters simultaneously from each of the drives and pieces them together for delivery to the PC. This process makes reads faster. How fast depends on the number of drives in the array. If two drives are mirrored, read time is cut approximately in half; three mirrored drives reduce read time to about one-third that of a single drive.



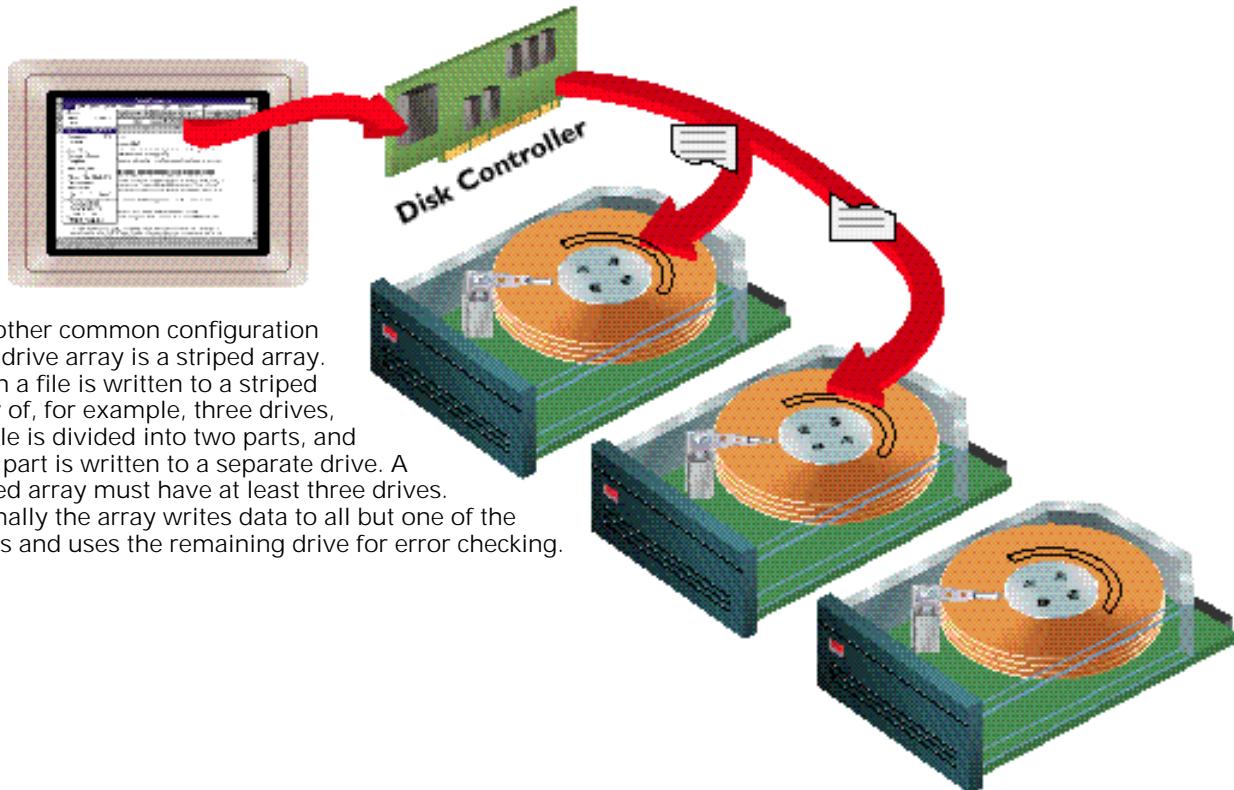
- 3 In case of a read failure—caused by either a defect on the surface of one of the drives or a crash of one of the drives—the controller simply reads the intact version of the file from the undamaged drive.



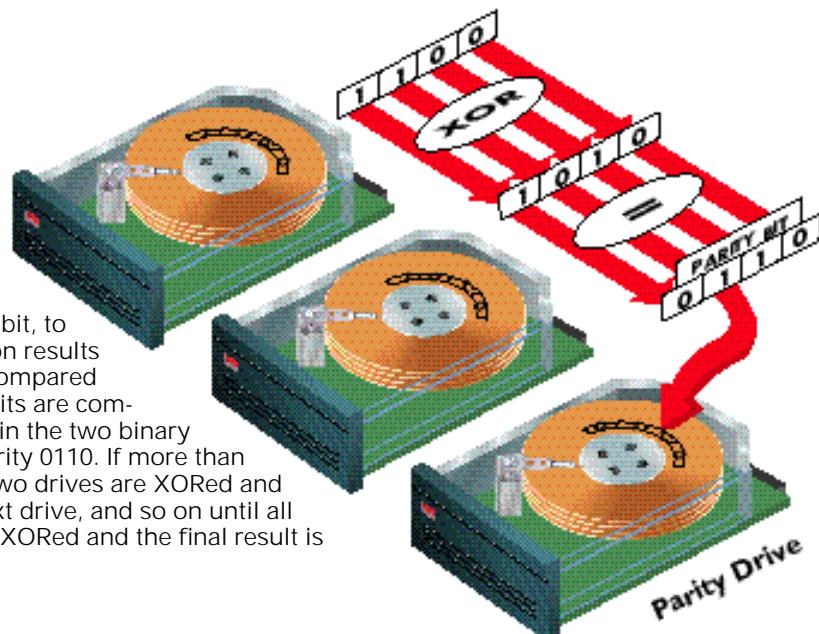
- 4 If the read failure is caused by a media defect, the controller automatically reads the data from the copy of the file on the other drive and writes it to a new, undamaged area on the drive on which the defect occurred.

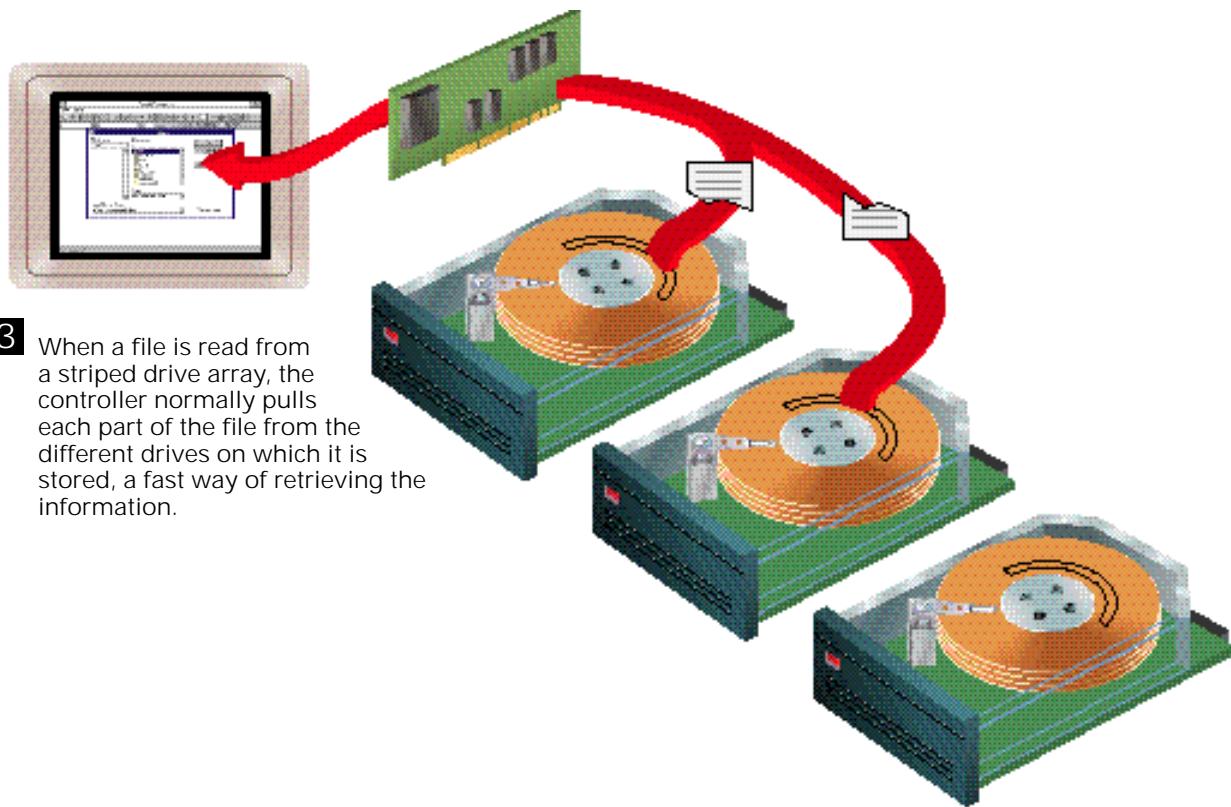
# Striped Drive Array

- 1 The other common configuration for a drive array is a striped array. When a file is written to a striped array of, for example, three drives, the file is divided into two parts, and each part is written to a separate drive. A striped array must have at least three drives. Normally the array writes data to all but one of the drives and uses the remaining drive for error checking.

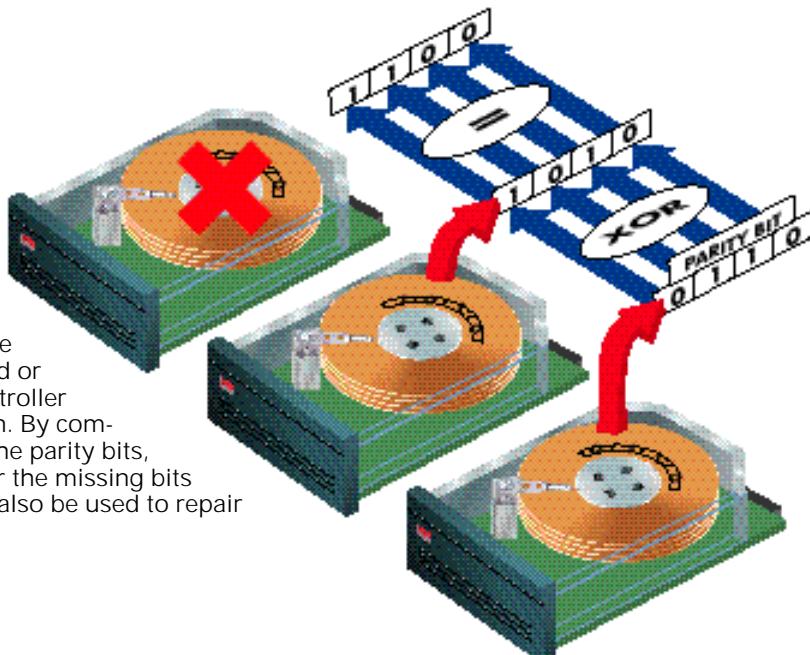


- 2 The controller or array software performs a Boolean XOR operation on the data written to the drives and writes the result, often called a parity bit, to the remaining drive. An XOR operation results in a 0 bit whenever two like bits are compared and a 1 bit whenever two dissimilar bits are compared. For example, XORing each bit in the two binary numbers 1100 and 1010 yields the parity 0110. If more than three drives are in an array, the first two drives are XORed and then that result is XORed with the next drive, and so on until all the drives containing data have been XORed and the final result is written to the parity drive.





- 3 When a file is read from a striped drive array, the controller normally pulls each part of the file from the different drives on which it is stored, a fast way of retrieving the information.

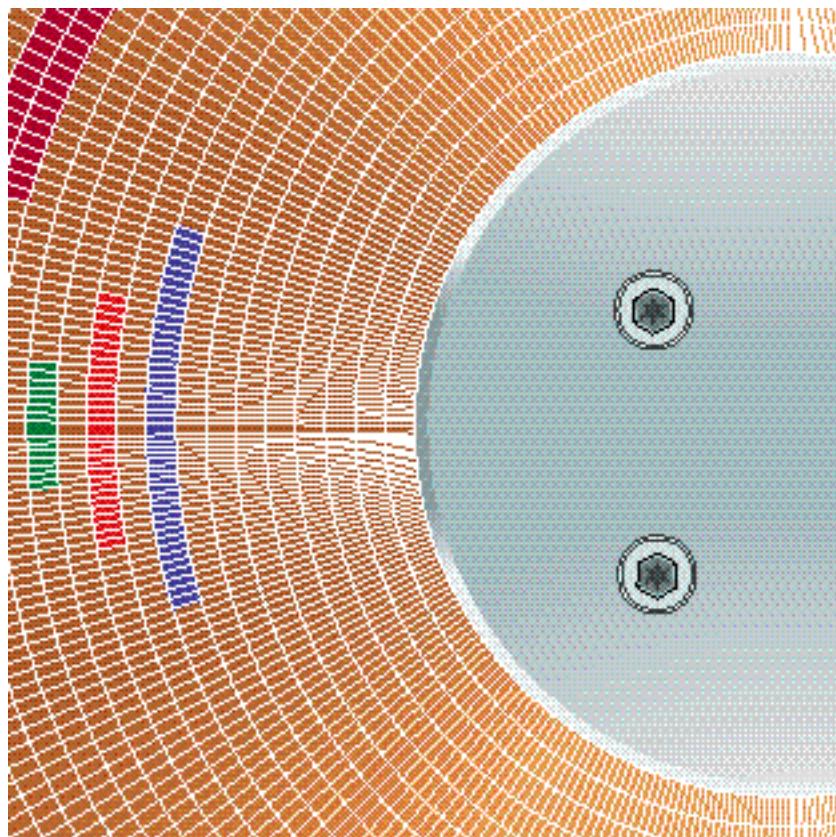


- 4 The security advantage of a striped array shows up in case one of the parts of the files is damaged or one of the drives crashes, the controller performs a reverse XOR operation. By comparing the undamaged bits with the parity bits, the controller can deduce whether the missing bits are 0s or 1s. The information can also be used to repair data caused by media defects.

CHAPTER

**15**

# How Disk Drives Increase Speed and Storage



AT the party attended by today's hotshot computer technology, the hard drive is a wallflower. With other components dancing about it at whirlwind speed, the hard drive just slugs along, dragging the party's performance down with it. Floppy drives and CD-ROM drives are slower, but their services are not as much in demand. If your computer ever makes you wait, chances are the hard drive is the culprit.

And it is a standard rule of computing that the size of any hard drive is always 100 megabytes less than the programs and data you need to store on it. Buy a drive twice as big, and before you know it, all your files have suddenly swollen so you're running out of room again.

The first hard drives for PCs held 10MB of files. Today a 1-gigabyte (1,000MB) drive just gets by, especially if it's holding bloated Windows programs, each one of which can easily take up more than 100MB of drive space. Drives continue to get bigger, and recent changes in how Windows handles files makes bigger drives even more practical. But they will never get big enough.

Speed, on the other hand, has hit a limit of about 9 milliseconds access time—the average time it takes the read-write heads to position themselves over the start of a file. They'll never be fast enough, either.

There are three ways to combat the limitations of hard drives. Disk caching is used automatically by Windows and is included with DOS. With caching, each time the PC reads some clusters from a drive, it stores their contents in memory on the theory—usually right—that they'll be used again soon. Caches also pick up the next few clusters because they are likely to be the next new clusters your system needs. But caching is limited when Windows applications consume so much RAM themselves. You're more likely to have your drive working as artificial RAM (using virtual memory) than your memory pretending to be a drive.

That leaves us with defragmenting—or optimization—and compression as the ways to faster, more capacious hard drives. Defragmenting puts all the parts of a file in one place so the drive doesn't waste as much time scurrying around to read all the scattered fragments. Defragging programs are part of DOS and Windows, and they're something you should use regularly.

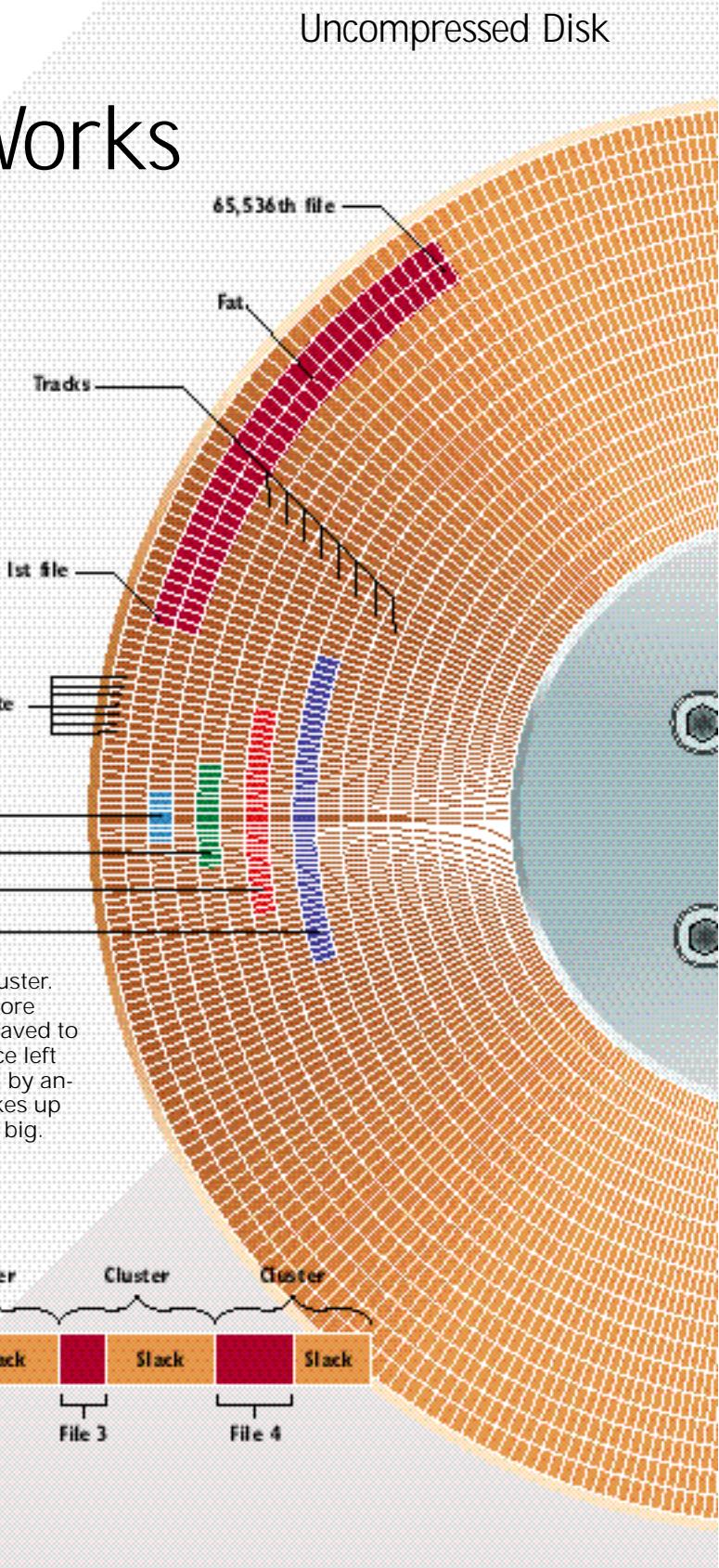
Compression—of both files and entire disks—has double rewards. Compression not only stores files more economically so that your hard drive seems bigger, it also speeds up a computer. With today's speedier processors, the extra executing time involved in decompressing files is more than made up for by the fact that smaller files can be pulled off the drive faster. Compression has had a bad reputation at times, and it was a favorite scapegoat when technicians couldn't figure out any other way a hard drive lost its data. Today, with compression a part of the operating system, that is no longer a serious fear.

# How Disk Compression Works

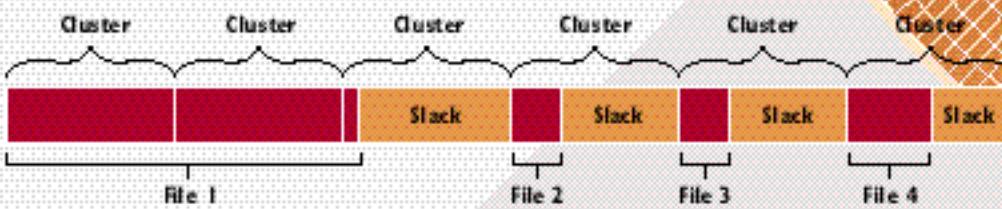
**1** On all but the very newest versions of Windows, the file allocation tables in Windows and DOS are limited to tracking the disk locations of no more than 65,536 files, regardless of the storage capacity of the drive.

**2** The size of a drive's cluster—the smallest unit of storage the drive can write a file to—is determined by dividing 65,536 into the total capacity of the drive. The result is that clusters get bigger as drives get bigger. Up to capacities of 256MB, a drive's clusters are each equal to 4K ( $256\text{MB} \div 65,536$ ).

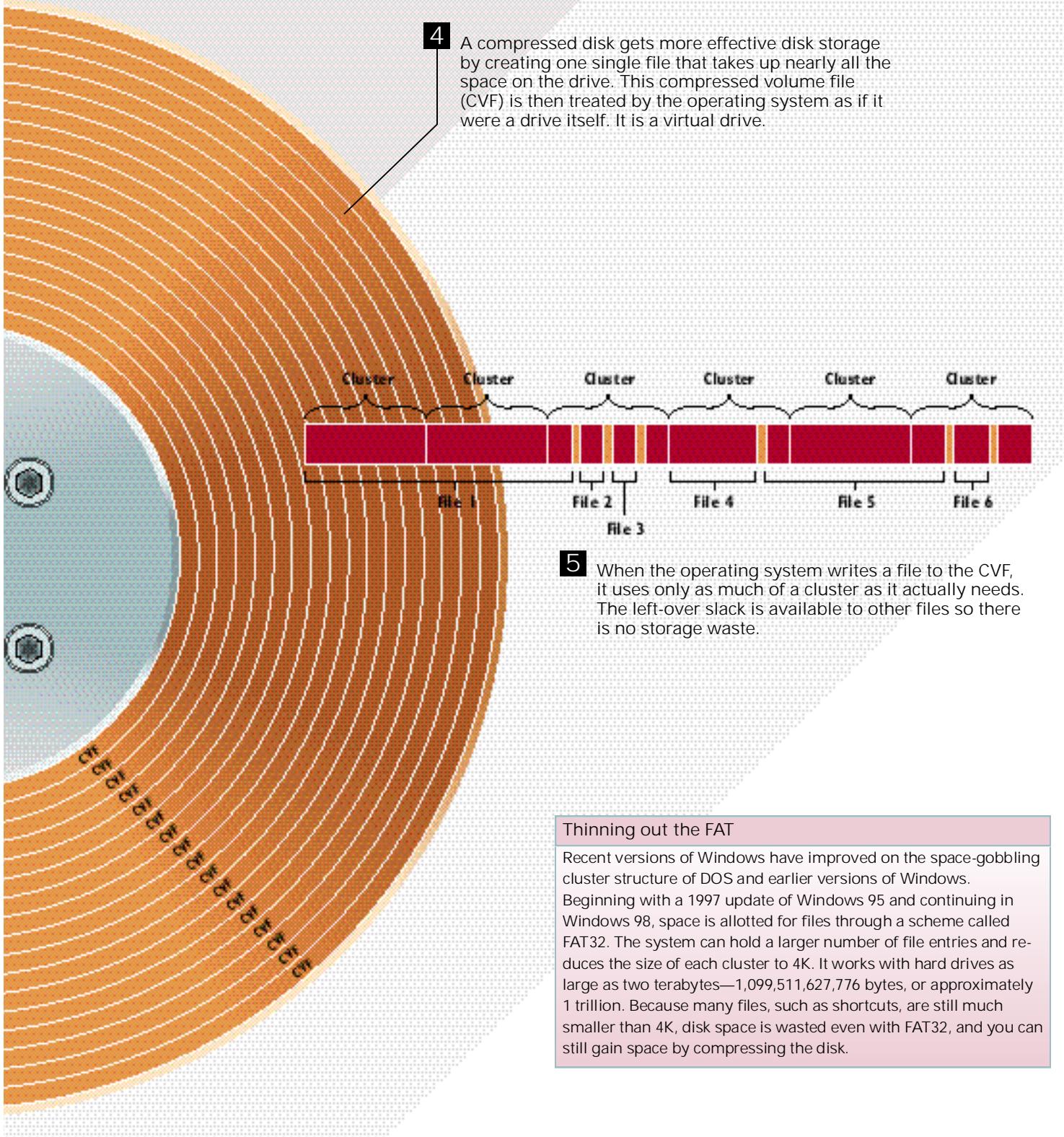
DRIVE SIZE	CLUSTER SIZE
256MB	4K
512MB	8K
1GB	16K
2GB	32K



**3** When a file is saved, it uses all or none of a cluster. An unformatted text file containing nothing more than a single letter is 1K. But when the file is saved to disk, it uses an entire cluster. The storage space left over in the cluster—the slack—cannot be used by another file. On a 2GB drive, a 1K file actually takes up 32K, the size of a single cluster on a drive that big.

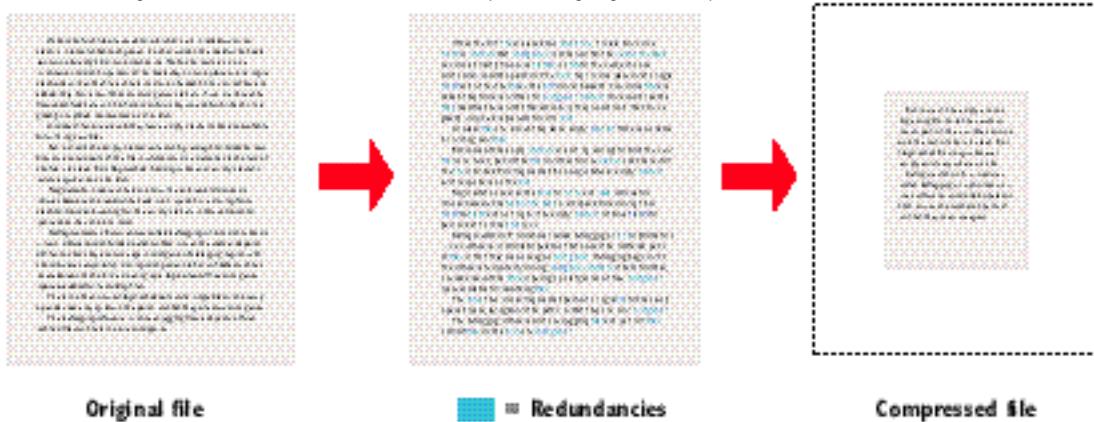


## Compressed Disk

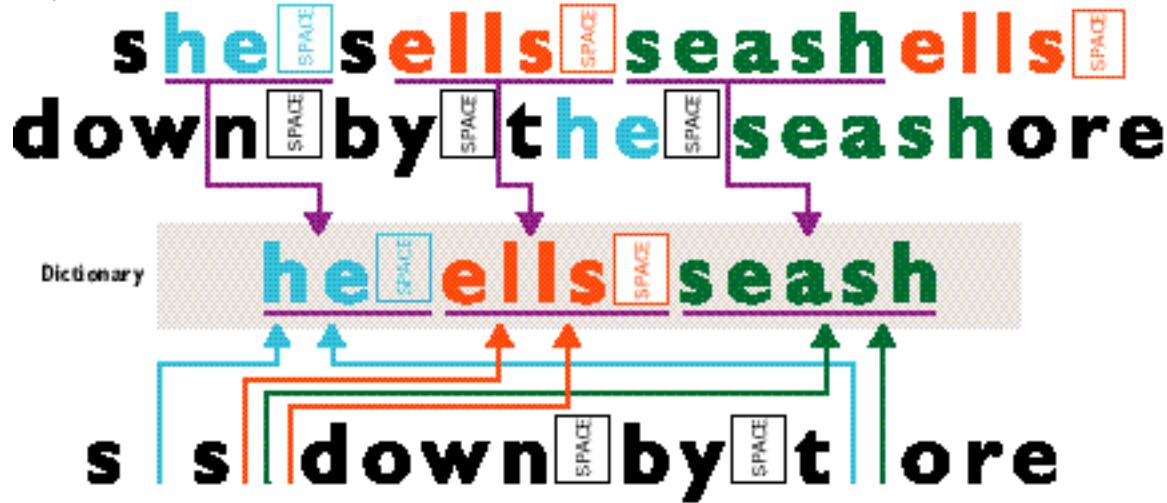


# How File Compression Works

- 1** Files, even those on a compressed disk, can be made still smaller by file compression a process often called zipping a file. This other type of compression reduces file size by eliminating the redundancies in the file. This is called lossless compression, which means every bit in the file can be restored precisely by decompression.



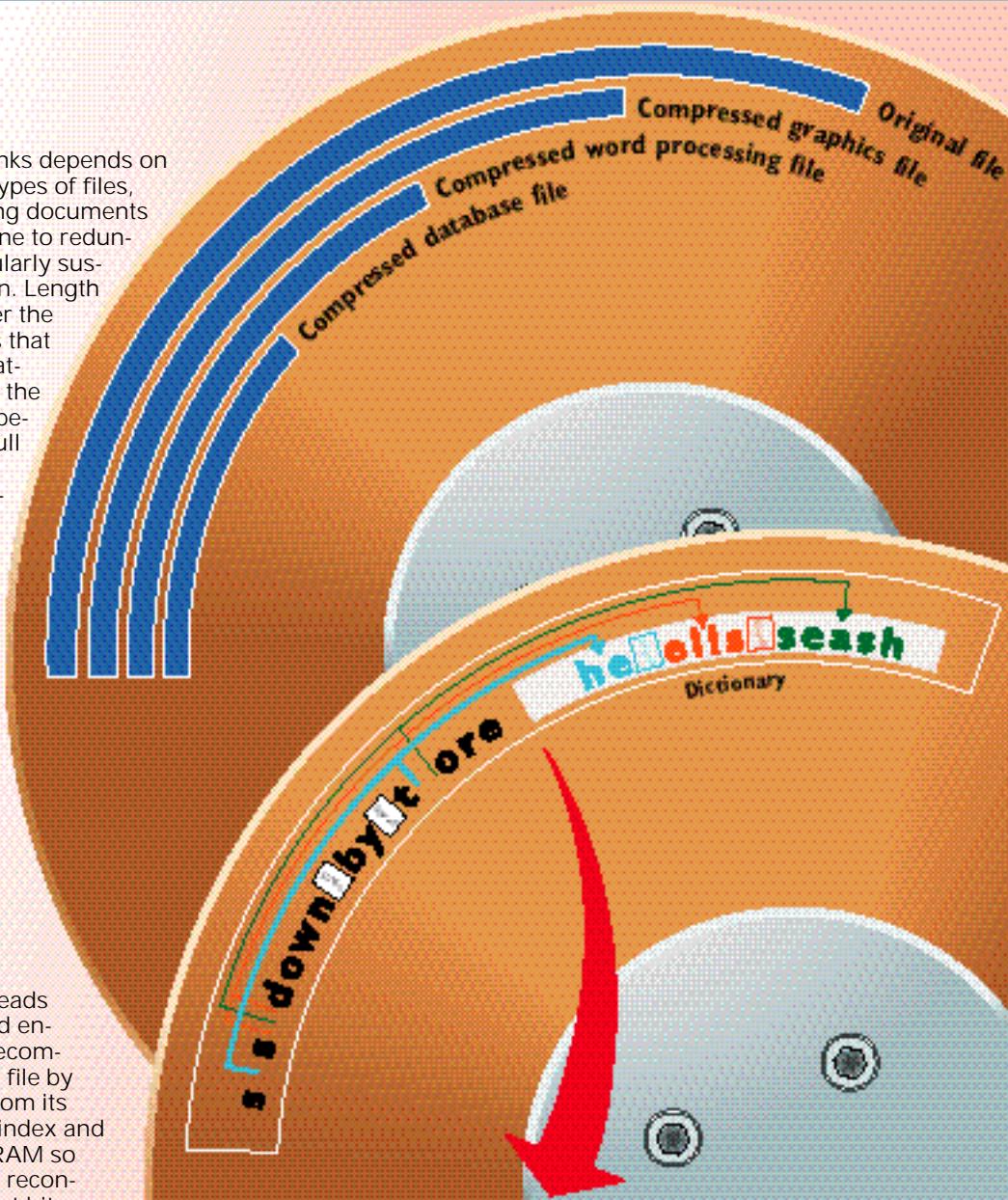
- 2** The compression program uses some variation of a scheme generally called LZ (after its creators, Lempel and Ziv) adaptive dictionary-based algorithm. As the program reads an uncompressed file, it examines the file for recurring patterns of data. When LZ identifies a pattern, instead of writing the pattern as it does other sections of the file, it writes the pattern to a dictionary. The dictionary is stored as part of the compressed file.



- 3** Where the pattern would have been written to disk, LZ instead writes a much shorter pointer that tells where the omitted pattern can be found in the dictionary. Adaptive means that as the algorithm writes more of the file, it constantly looks for more efficient data patterns to use and changes the entries in the dictionary on-the-fly. In the example here, LZ could have chosen "se" in "sells," "seashells," and "seashore" as one of the patterns for the dictionary, but it's more efficient to use "ells" and "seash."

- 4 How much the file shrinks depends on the type of file. Some types of files, such as word processing documents and databases, are prone to redundancies and are particularly susceptible to compression. Length matters, too. The longer the file, the more likely it is that LZ can find repeated patterns. In the sample on the opposite page, picked because it's unnaturally full of redundancies for so short a phrase, the savings is 18 percent (40 bytes for the original compared to 33 for the compressed version). Some files can be compressed to as little as 20 percent of their original size. On the average, however, files are more likely to be reduced to only half the original size.

- 5 When your computer reads the compressed file and encounters a pointer, it decompresses that part of the file by retrieving the pattern from its place in the dictionary index and writing the pattern to RAM so that the original data is reconstructed down to the last bit.



## she sells seashells down by the seashore

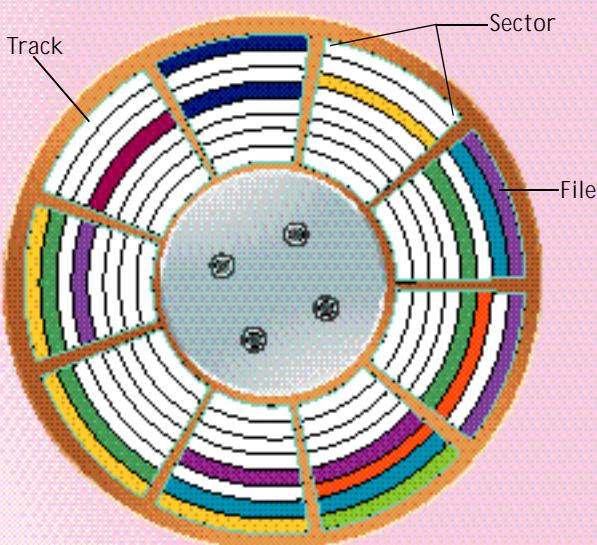
### Other Compression

The formats for some types of files—particularly graphics and database files—include their own compression. Graphics files often use lossy compression, which reduces file size by discarding forever data, such as small variations of color, whose loss won't be noticed. Lossy compressed files cannot be decompressed to their original state.

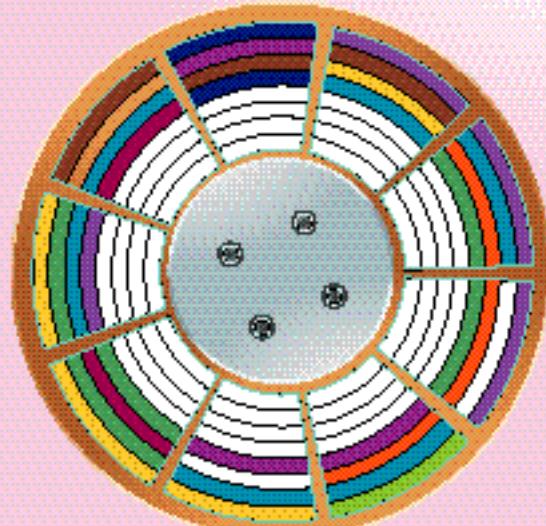
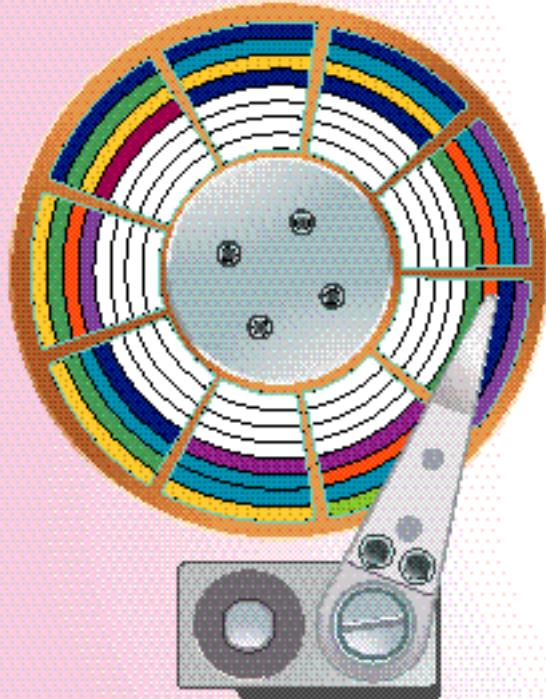
Utility programs such as PKZip can further compress files already squeezed by disk compression or built-in file compressions. The utilities use different algorithms that emphasize storage efficiency at the expense of speed.

# How Disk Defragging Works

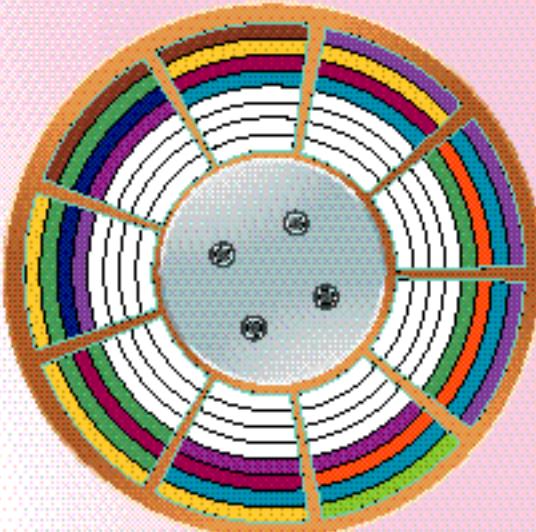
- 1 When the first file is saved to a disk drive, it is laid down on a track in clusters that are contiguous. In other words, the read/write head can move directly from one cluster in a file to the next, all in one continuous, smooth operation. The head stays in one place over a single track and writes the file as the disk moves beneath it. As more files are added, they too are written to contiguous clusters. If we could see the files, identified here with different colors, they would look like this in a greatly simplified representation of a disk.



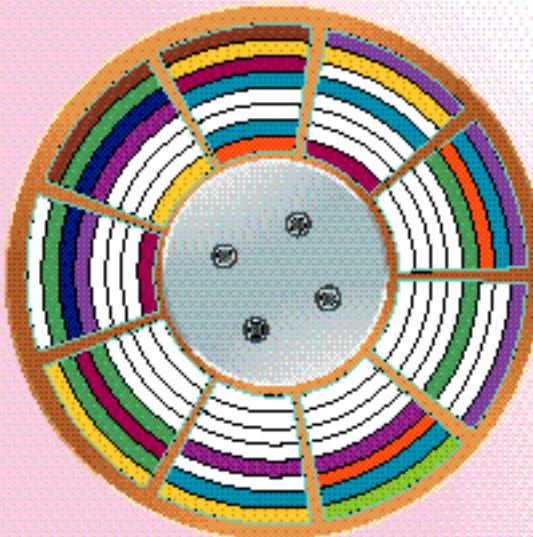
- 2 As older files are erased, they leave empty clusters that are available for writing new files.



- 3 But some of the empty clusters are not big enough to hold the new files. As a result, part of the file is written to one cluster, and the rest of the file is divided—or fragmented—among whatever empty clusters exist anywhere else on the disk.

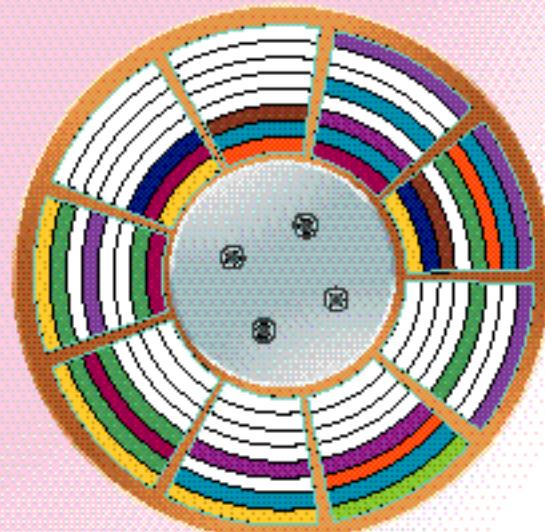
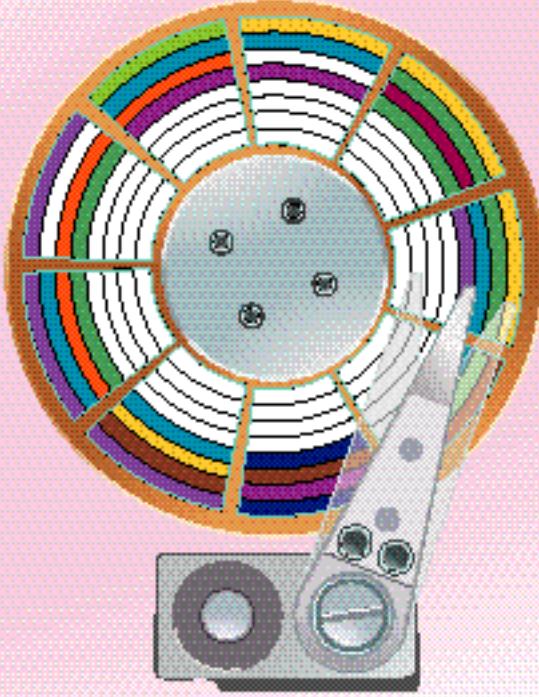


- 7 The defragging software continues juggling files and parts of files until all files on the drive are contiguous.



- 6 The drive then moves fragmented parts of a single file to the newly opened space, laying down the parts so that they are now contiguous.

- 4 Fragmentation causes the drive to write and read information slower because the read/write head must spend time moving from track to track and waiting for the empty clusters in those tracks to pass under it as the disk spins.

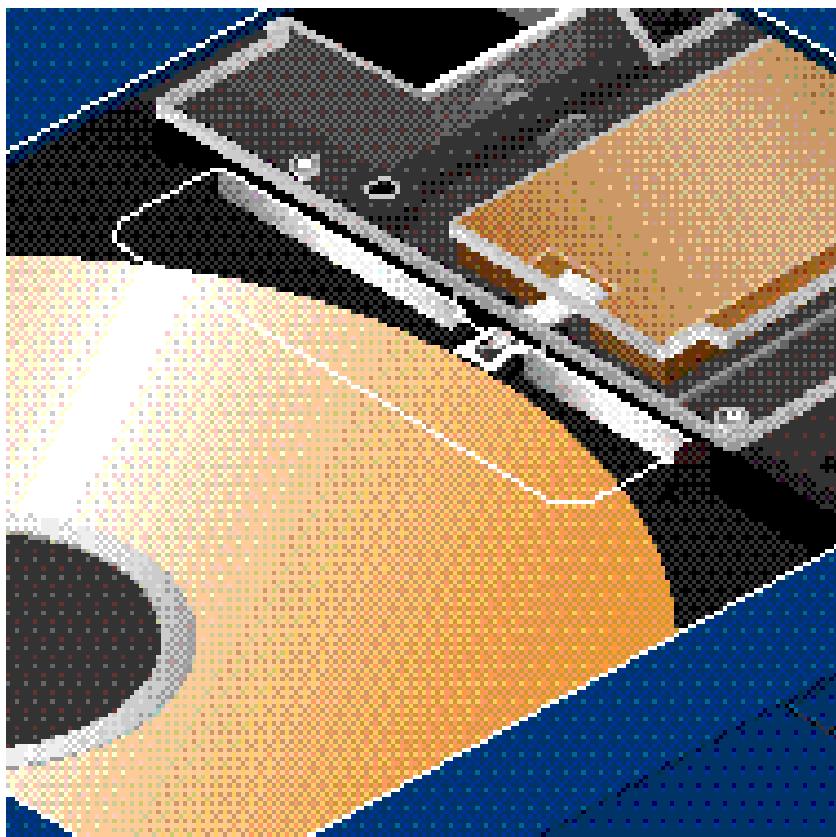


- 5 Defragmentation—sometimes called defragging or disk optimization—is a software-controlled operation that moves the scattered parts of files so that they are once again contiguous. Defragging begins with the software temporarily moving contiguous clusters of data to other, unused areas of the drive, opening up a large area of free contiguous space available for recording files.

CHAPTER

16

# How Other Removable Storage Works



THE magnetic signals used by conventional floppy drives and hard disks save data in strips that are microscopically small. But get down on their level, and the magnetic signals are grossly crude. The area affected by the electromagnets in read/write heads, microscopically speaking, is a wide, wandering river. You could put a lot more data on the same size disk if you could just restrict the data to narrow, tightly packed channels. You can, through increasingly precise mechanics and by combining magnetism with the precision of a laser.

A beam of light produced by a laser can be narrowed to a much smaller area than that affected by a magnetic read/write head. But harnessing lasers alone to write as well as read data isn't easy or cheap—although it is done in writeable compact disc and DVD drives. An easier, cheaper way is to combine magnetic heads with the precision of a laser beam. This creates a drive that puts a lot of data into a very small—and portable—package.

The first attempts to combine lasers and storage resulted in something called a WORM. Standing for Write Once, Read Many, a WORM could, indeed, pack hundreds of megabytes of data onto a single, removable disk. The problem with WORM is that once data was written to a disk, it could not be changed—or even deleted. Writeable CD-ROMs are examples of WORM drives. Today, WORM CD drives are most useful in situations in which you want to keep an unalterable audit trail of transactions for making personalized collections of music tracks, and for distributed samples of software.

Two other types of drives—magneto-optical (MO) and floptical—approach the challenge of using lasers from different angles, but both wind up in the same neighborhood: more writeable data storage. A magneto-optical disk is about 5½ inches in diameter, similar to larger floppy disks. It contains up to 500MB, and magneto-optical drives that can hold twice that much information are being developed. A floptical disk is the same size as a 3½-inch floppy. It holds up to 20MB.

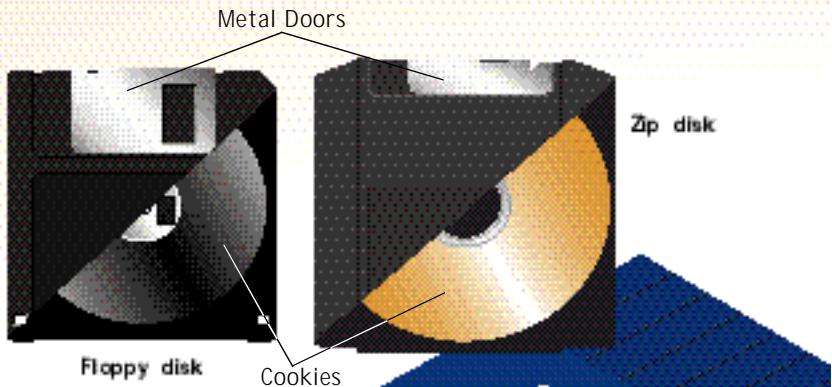
Although MO drives and flopticals use technologies that are different from each other and from that in CD-ROMs, they are similar in that all use lasers to pack data so tightly that you can save dozens to hundreds of megabytes of information on a single disk—a disk you can move from one machine to another.

The most recent and popular solution for removable storage is Iomega's Zip drive. It doesn't use a laser, but it still manages to pack 1MB onto a single disk about the size of a 3½-inch floppy.

Zip drives, MO drives, and flopticals are reasonably priced alternatives to CD-ROM and tape drives for backup, offline storage, and transfer of large files from one PC to another.

# How a Zip Drive Works

- 1** A disk for an Iomega Zip drive looks similar to an ordinary floppy disk. It's about the same size, although slightly thicker. And yet a normal 3.5-inch floppy disk can hold a maximum of only 1.4MB of data. The Zip disk holds 100MB and accesses that data at speeds that rival low-end hard drives. The magnetic-coated Mylar disk inside each, called the cookie, is protected by a hard plastic shell, with a metal plate that slides open to give read/write heads access to the cookie.

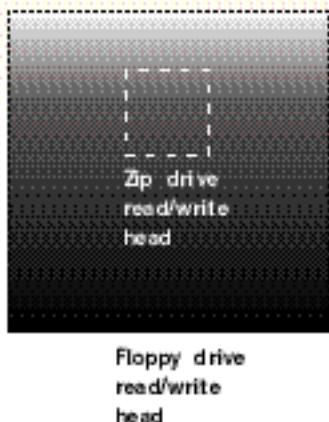


- 2** When you insert a Zip disk into the drive, the metal shield slides to one side to expose a small opening along the leading edge of the plastic case. A metal hub in the cookie is engaged by a motor that spins at 3,000 rpm. This gives the drive a faster access time than a regular floppy turning at 360 rpm.

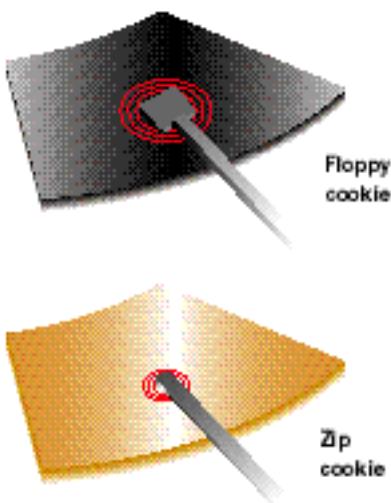
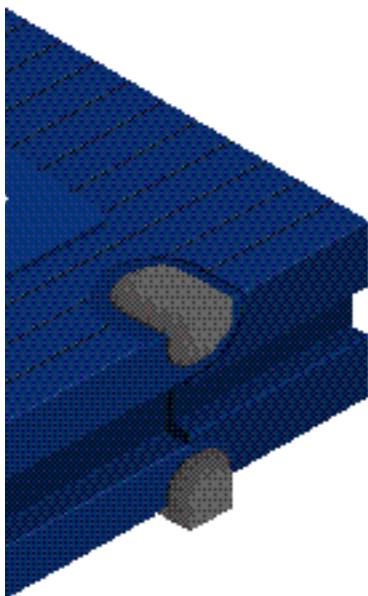
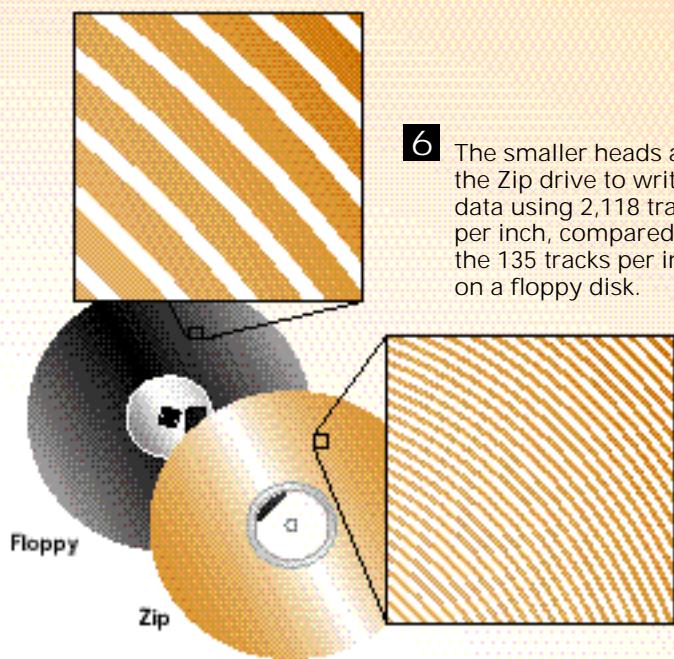
- 3** When the disk is fully inserted, the hole in its casing matches up with a hole in the housing that surrounds the read/write heads. This design minimizes the amount of scratch damage done to the cookie by dust and other contaminants in the air.

- 4** The two read/write heads—one for each side of the cookie—extend into the casing. Unlike hard drive heads, which are never supposed to touch the surface of the disk, the Zip's heads do touch the cookie, but only very lightly compared to a floppy drive. The light touch permits the higher spin rate and produces less wear and tear on the cookie.

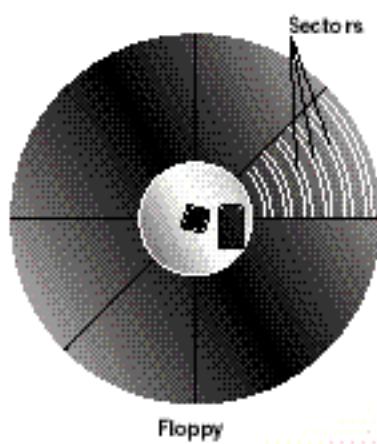
- 5 The read/write heads are about one-tenth the size of a floppy drive's heads—closer to the size of read/write heads in a hard drive.



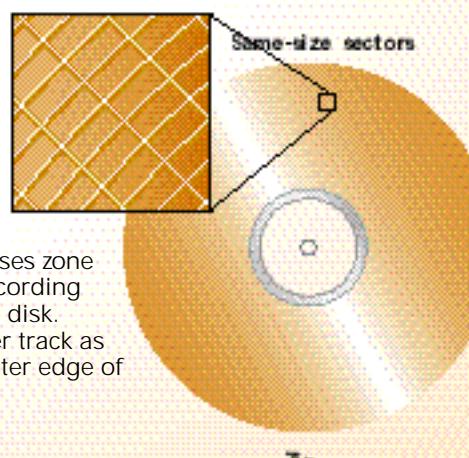
- 6 The smaller heads allow the Zip drive to write data using 2,118 tracks per inch, compared to the 135 tracks per inch on a floppy disk.



- 7 Further storage capacity comes from coating the cookie with the same magnetic particles that are used in S-VHS video tape. The particles have a higher energy level, which means they are not as easily magnetized. As a result, magnetic fields from the write head affect a far smaller area than a floppy drive does. The smaller the surface area needed to write a 0 or 1 bit, the more bits can be packed onto the same track.



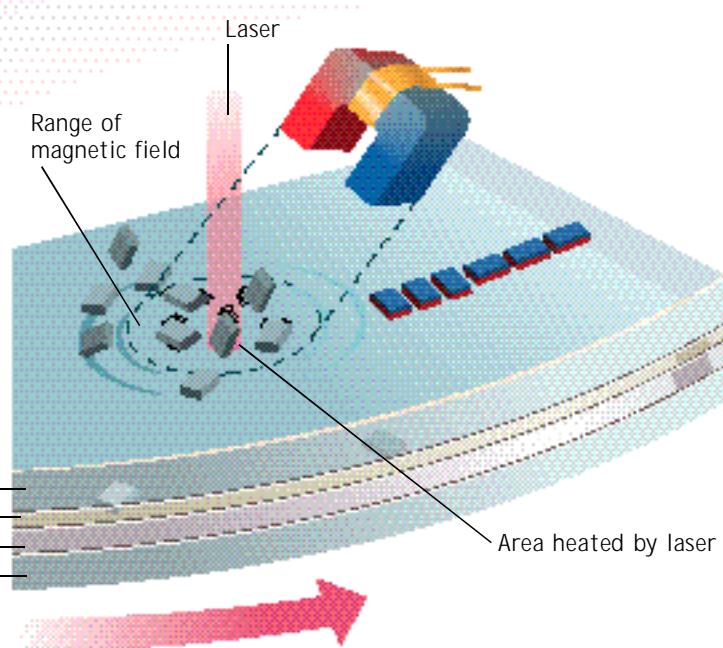
- 8 A conventional floppy is divided into sectors radially. There are as many sectors in the outermost track as there are in the innermost. As a result, the outer tracks use up more surface area than the inner tracks—a waste of the recording surface. A Zip drive, along with hard drives, uses zone recording, so that the same recording density is used throughout the disk. This results in more sectors per track as the heads move toward the outer edge of the cookie.



# How Magneto-Optical and Floptical Drives Work

## Magneto-Optical Drives

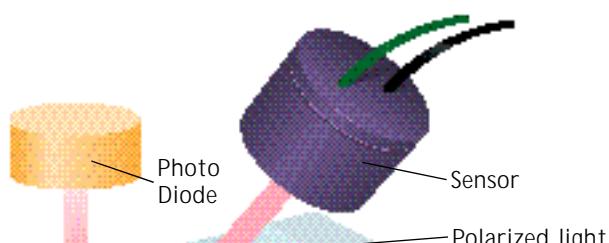
- 1** The electromagnetic read/write head of the magneto-optical drive generates a magnetic field that covers a relatively large area on the drive's disk. But the crystalline metal alloy that covers the surface of the disk is too stable to be affected by the magnetic field alone.



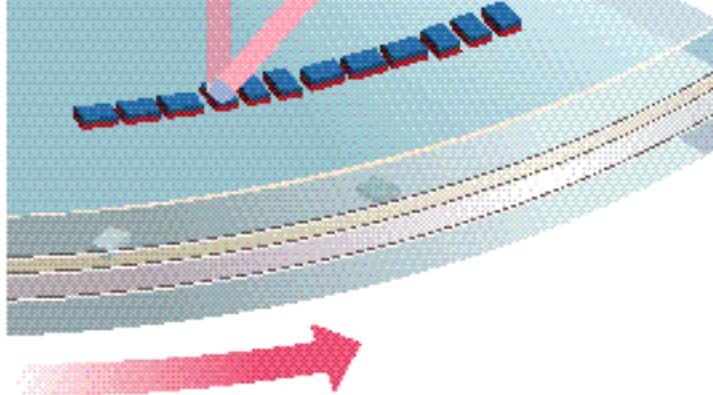
- 2** A thin, precise laser beam is focused on the surface of the disk. The energy in the beam heats up a tiny spot in the alloy to a critical temperature known as its Curie point. At this point, the heat loosens the metallic crystals in the alloy enough that they can be moved by the write head's magnetic field, which aligns the crystals one way to represent a 0 bit and a different way to represent a 1 bit.

- 3** To read data from a magneto-optical disk, a weaker laser beam is focused along the tracks of data that had been created with the help of the more intense laser.

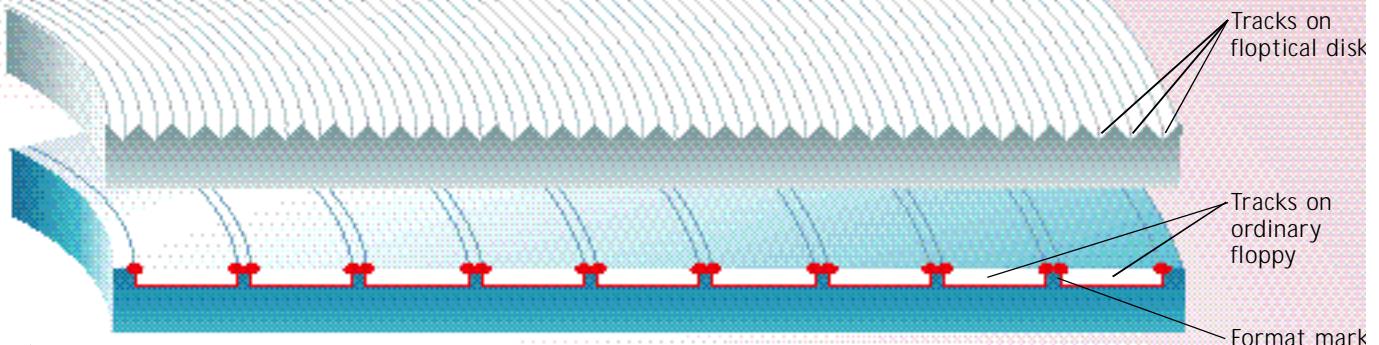
- 4** The crystals in the alloy polarize the light from the laser. Polarization allows only light vibrating in a certain direction to pass through the crystals. The alignment of crystals in 0 bits polarizes the light in one direction, and the crystals in 1 bits polarize light in a different direction.



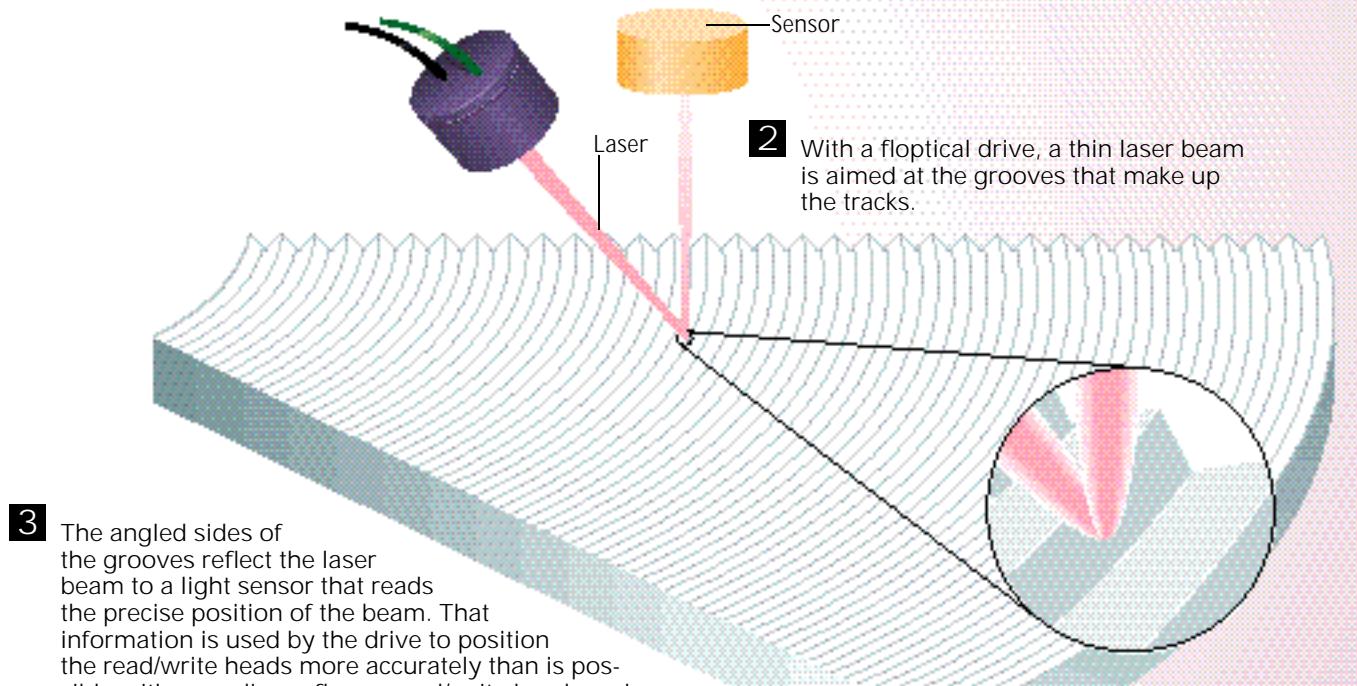
- 5** The polarized light is reflected from the aluminum layer of the disk to a photo diode, which senses the direction in which the light is polarized and translates that information into a stream of 0s and 1s.



## Floptical Drives



- 1** A floptical disk is created with a series of small, precise concentric tracks stamped in its barium-ferrite surface coating. The tracks, where data is written, are thinner and more numerous than those created by ordinary floppy disk formatting. On conventional floppies, the operating system formats the disk by recording magnetic markers on the surface of the disk to create a road map of tracks and sectors. (See Chapter 12.) The markers are used to find areas where files exist or where new files can be written. But this method of formatting has wide tolerances for error to allow for the imprecision with which the drive head reads those markers. Those wide margins limit how much can be stored on a regular floppy.

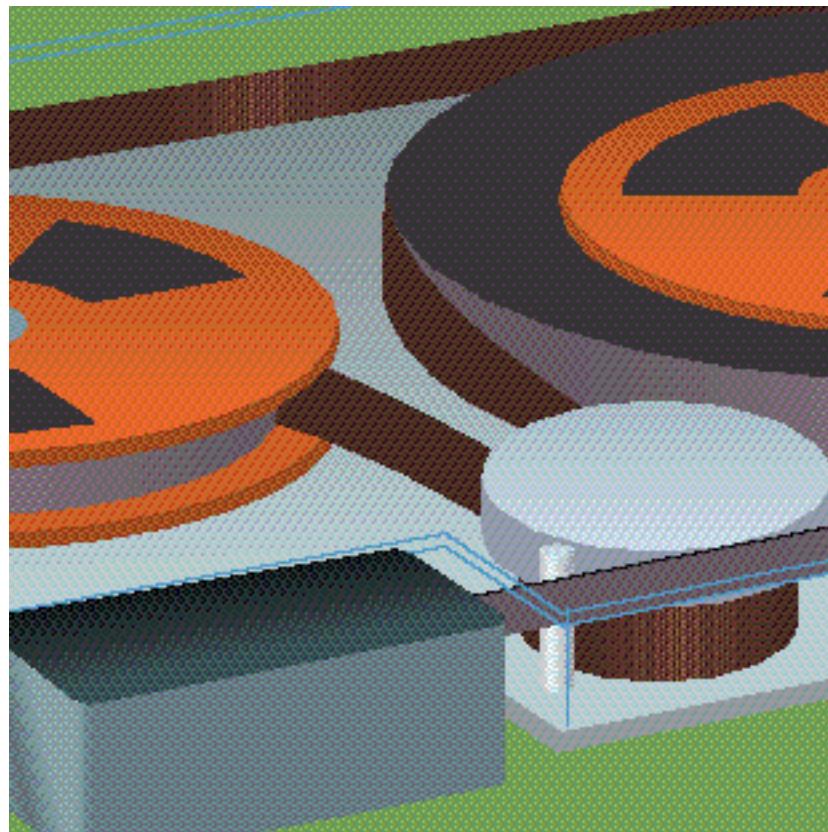


- 2** With a floptical drive, a thin laser beam is aimed at the grooves that make up the tracks.
- 3** The angled sides of the grooves reflect the laser beam to a light sensor that reads the precise position of the beam. That information is used by the drive to position the read/write heads more accurately than is possible with an ordinary floppy read/write head mechanism. The combination of more tracks and more precise writes of bits allows the floptical to hold nearly 20 times as much data as a conventional floppy.

CHAPTER

**17**

# How a Tape Backup Works



**BACKING** up your hard disk to a tape drive used to be like one of your mom's warnings: Take an umbrella with you on cloudy days and always wear a raincoat. Sure, Mom was right once in a while—rain would come down and you would get wet, but it wasn't all that terrible. So what if your hard disk's file allocation table got scrambled and you lost half your files? A few years ago, as long as you'd copied a few essential data files to floppies, re-creating a couple of megabytes of programs from their original distribution disks wasn't that much trouble.

Today, however, the implications of a "little" hard-disk disaster have mushroomed. You're more often talking about hard drives that contain not just a few megabytes of files, but hundreds. A single Windows program may include 100 megabytes of files. And with a complex environment such as Windows, no program exists alone. Every Windows program you install modifies the Windows Registry. Plus, how many tweaks have you made to your system? From arcane parameters to a device drive to the Windows icon collection, not to mention wallpaper and system sound schemes you spent hours perfecting—you could never hope to remember all those tweaks.

But if backing up data has never been so critical, it's also true that there have never been more ways to back it up. Forget about backing up to floppy disks; you don't want that many floppies lying around. But writeable media such as Zip drives, floptical, and rewriteable CD-ROMs make back-up practical.

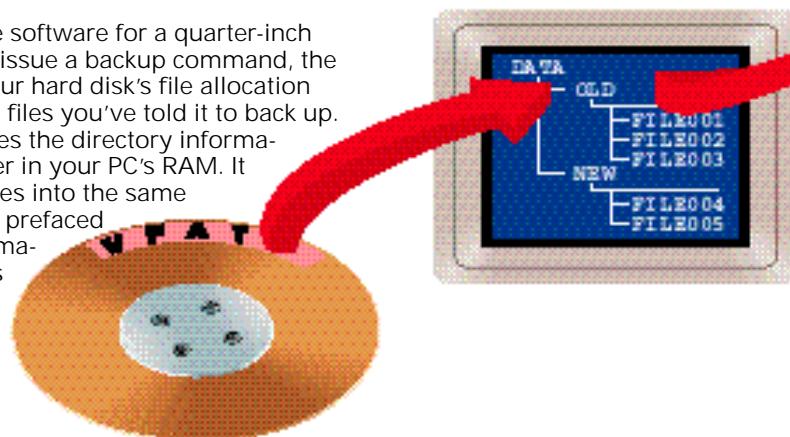
Here, we look at the most traditional method of backing up—tape drives. Tape drives are more capacious than Zip and floptical drives. But best of all, tape drives excel at unattended operation. With the right software, backups take place every night without you thinking about it.

Prices for tapes drives are now under \$300, making them affordable even for home systems. And the ability to copy several gigabytes to a single tape makes them simple to use for even the biggest hard drives.

Here are the workings inside two of the most popular types of tape backups: quarter-inch cartridge (QIC) and digital audio tape (DAT).

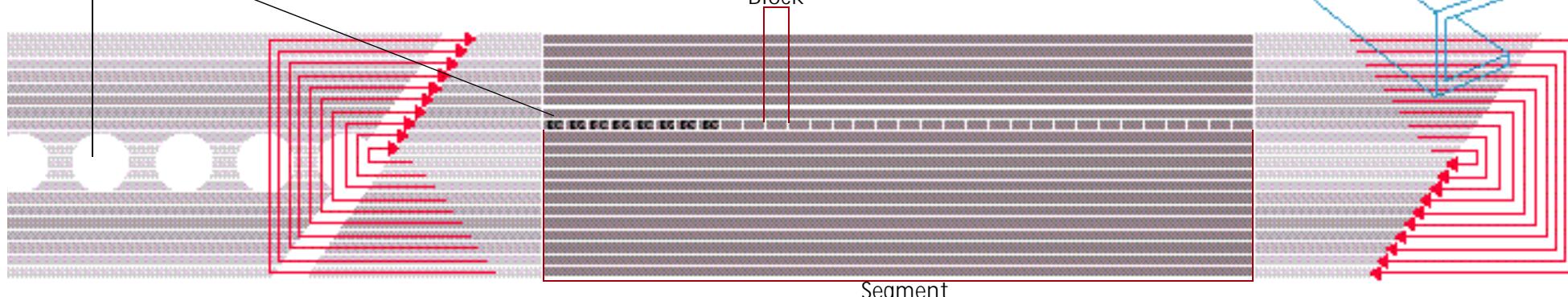
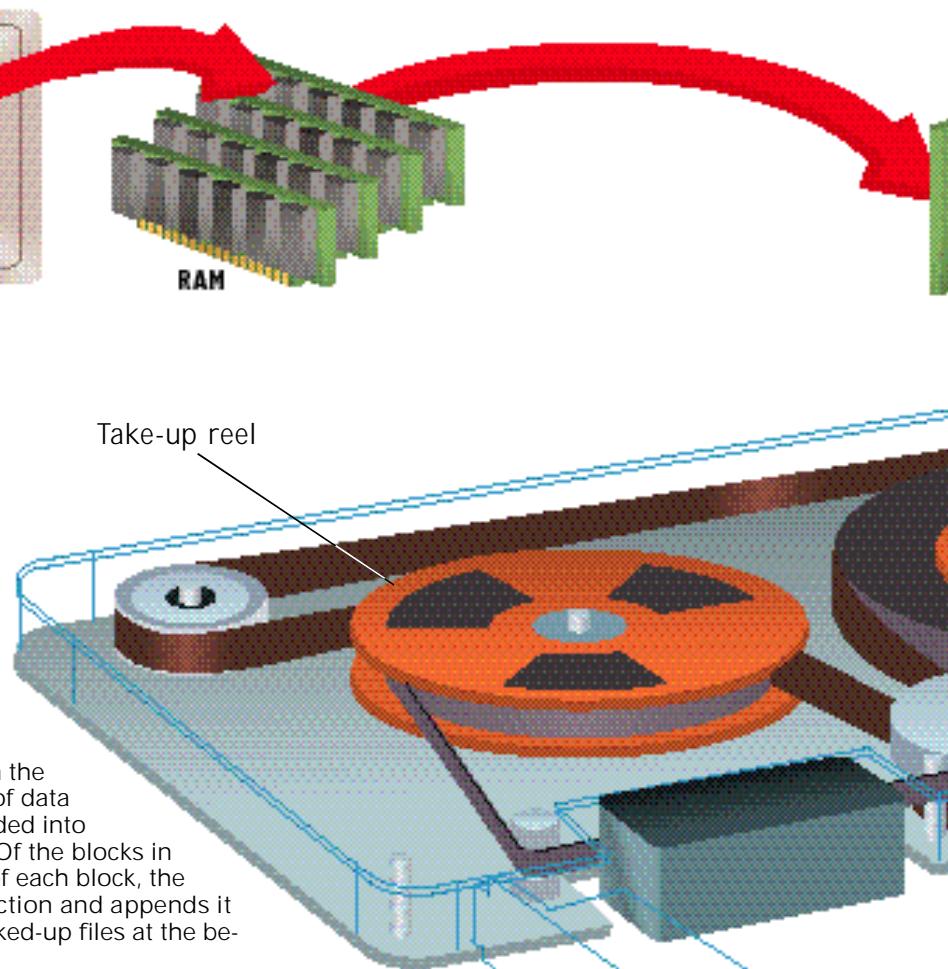
# Quarter-Inch Cartridge (QIC) Tape Backup Drive

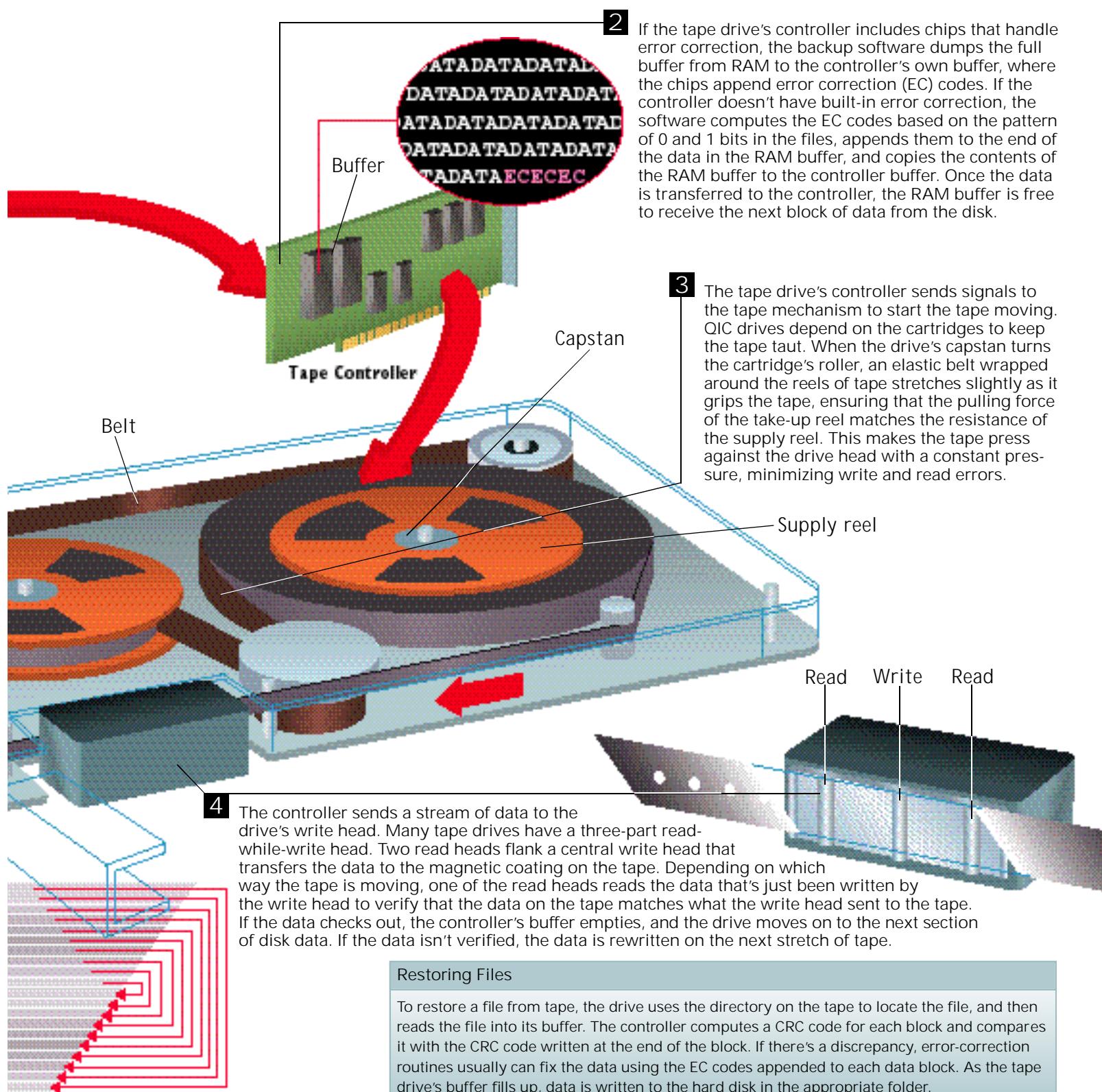
- 1** When you use the software for a quarter-inch cartridge drive to issue a backup command, the program reads your hard disk's file allocation table to locate the files you've told it to back up. The software writes the directory information to a 32K buffer in your PC's RAM. It then copies the files into the same buffer. Each file is prefaced with header information that identifies the file and its location on the hard drive's folder tree.



- 6** As either end of the tape approaches the drive head, holes punched in the tape signal the drive to reverse the direction of the tape and to shift the active area of the recording head up or down to the next track and then continue recording. When all the data has been written to the tape, the backup software updates the tape's directory with the track and segment locations of the files that it has backed up.

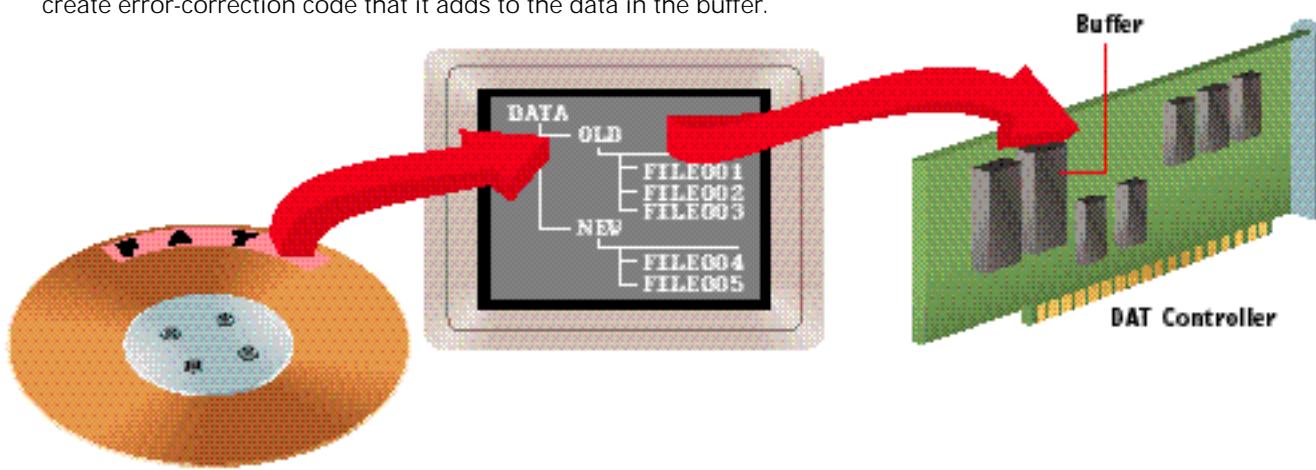
- 5** The format of a QIC tape typically contains 20 to 32 parallel tracks. When the tape reaches either end of a spool, its movement reverses and the flow of data loops back in a spiral fashion to the next outside track. Each track is divided into blocks of 512 or 1,024 bytes, and segments typically contain 32 blocks. Of the blocks in a segment, eight contain error-correction codes. In addition, at the end of each block, the drive computes a cyclic redundancy check (CRC) for further error correction and appends it to the block. Most backup software reserves space for a directory of backed-up files at the beginning of track 0 or in a separate directory track.



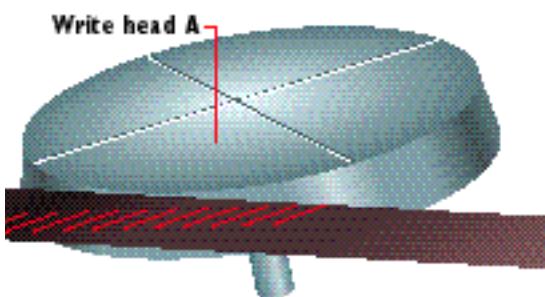


# Digital Audio Tape (DAT) Backup Drive

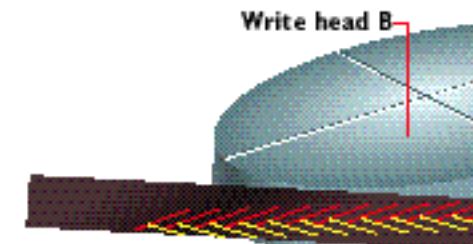
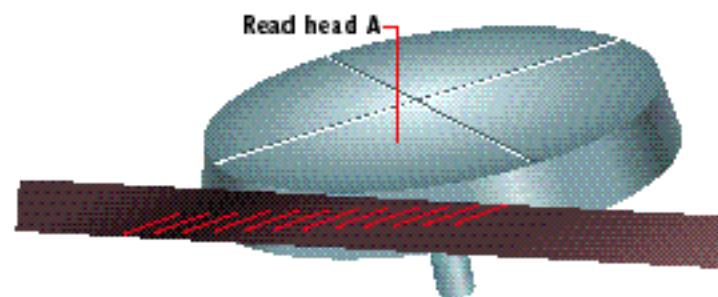
- 1 When you issue a backup command from your software, the program checks your hard disk's file allocation table to find the files to back up. Then it copies the data, file by file, into the DAT drive's buffer, which usually has room for 512K to 1MB of data. Like a QIC tape drive, the DAT drive performs an algorithm on the data to create error-correction code that it adds to the data in the buffer.

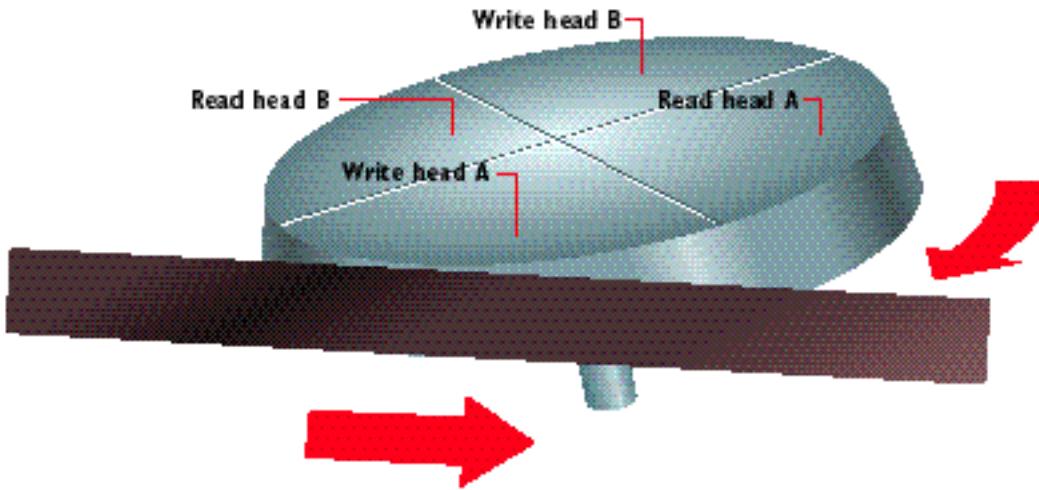


- 3 During the time that write head A is in contact with the tape, it writes about 128K of data and error-correction codes from the drive's buffer to a track on the tape. Because the cylinder is tilted, the head encounters one edge of the tape at the beginning of the write head and moves diagonally across the tape until it reaches the other side. This results in a narrow diagonal track about eight times longer than the width of the tape.

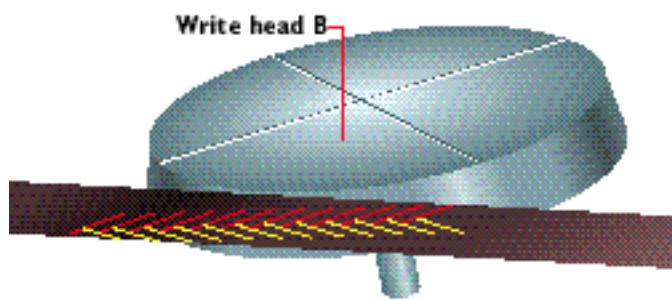


- 4 Read head A reads back and verifies the data in track A, bit by bit, against the data still in the buffer. If the data on the tape checks out, it's flushed from the buffer, and more data is read from the hard disk. If the data in track A contains errors, the data will be rewritten on the next pass.

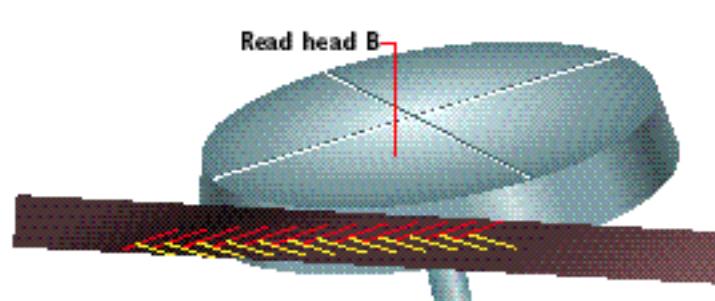




- 2** The distinctive design of the DAT drive's read/write head is what allows it to back up huge amounts of data onto a tape cartridge about the size of a matchbox. The mechanism is a rotating cylinder with four heads 90 degrees apart. Two of these heads, write heads A and B, write backup data, and two corresponding read heads verify the data. The cylinder tilts slightly so it rotates at an angle to the tape. The cylinder spins 2,000 times a minute while the tape, at a rate of half an inch a second, passes in front of the cylinder in the opposite direction of the cylinder's spin.



- 5** As write head B passes over the tape, it writes data in a track at a 40-degree angle to track A, making a crisscross pattern that overlaps track A. The overlapping data packs more information per inch of tape; it isn't misread later because the magnetic bits written by the two write heads have different polarities, and the different read heads read data only from properly aligned tracks.



- 6** Read head B and write head B go through the same steps, alternating with the A heads until all the data is backed up. Then the drive rewinds the tape and writes a directory of stored files either in a special partition at the front of the tape or in a file on the hard disk.

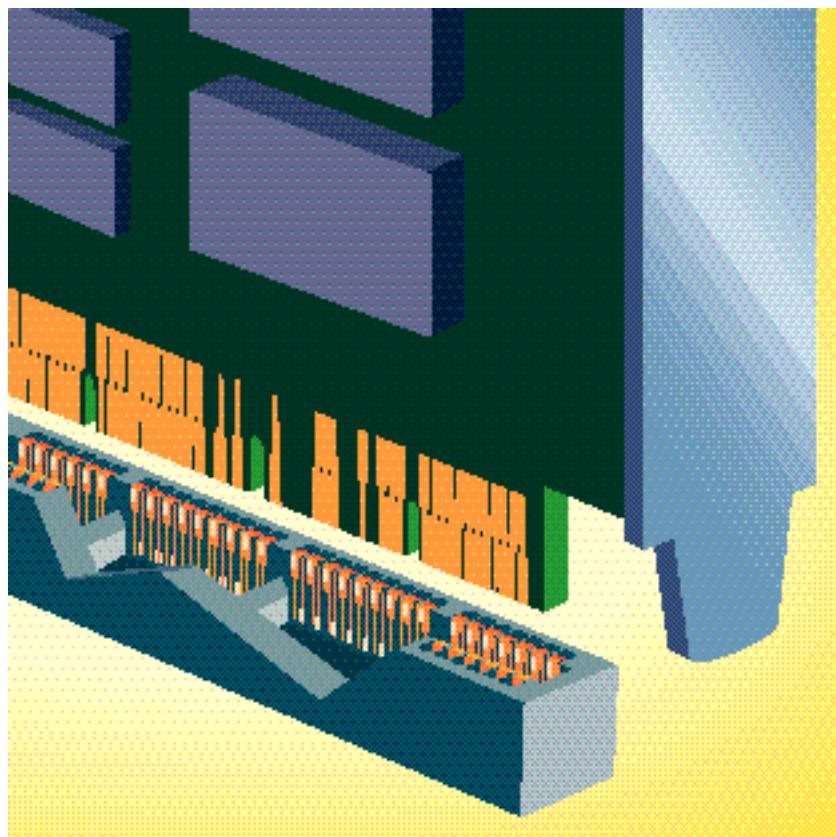
#### Restoring Files

When you restore files from the DAT drive, the software reads the directory, winds the tape to the spot where the requested files begin, and copies the files to the hard disk.

CHAPTER

18

# How a Bus Works



WE usually think of input and output (I/O) as ways for us to communicate with our computers. That may be true from our viewpoint, but as far as your PC is concerned, it's got a lot more I/O to worry about than just yours. Millions of bits of information are constantly flashing among the components of your PC even when it appears to be simply sitting there. Various traffic cops called input/output controllers work with the processor to make sure that all this data swapping doesn't cause a traffic jam, or worse still, a crash.

The bus is the highway system for this data. The bus transports data among the processor, memory, and other components. The bus includes a complex conglomeration of skinny electrical circuits called traces printed on the top and bottom of the motherboard, which is the main circuit board in your PC. But there's no single part of the PC's motherboard you can point to and say it's the bus. The bus also includes assorted microchips and the slots into which we plug expansion circuit boards—often called adapters or expansion cards. Sometimes the bus is called the expansion bus, and the slots lined with dozens of metallic contacts are called expansion or adapter slots. Just to complicate matters, there is more than one bus. In addition to The Bus (the motherboard-based bus), there are buses for the processor, memory, SCSI connections, and the latest innovation, a universal serial bus.

The idea of having slots into which you can plug other circuit boards that work with the main motherboard is one of the best features of personal computers. Without the slots, you'd be stuck with whatever video, disk controller, and other circuitry that was permanently wired into the motherboard. For example, expansion slots allow you to remove one card that controls the video display and replace it with a new video card that handles 3-D graphics faster. You can even add circuit boards, such as sound cards, that weren't even imagined when your PC was built. Today there's a trend toward making some components, such as parallel ports, serial ports, and video controllers part of the motherboard. But in the case of, for example, an integrated video controller, it can be disabled if you want to install an expansion card that handles video better.

The basic idea of the bus introduced on the IBM PC in 1981 was so good and so versatile that for years, there were few changes. But today there are a half-dozen types of PC buses. All of them represent improvements in moving data still faster among components.

The first change in the original PC bus was to increase its ability to move only 8 bits of data at a time. When IBM introduced the IBM AT computer in 1984, the new system included expansion slots with more connectors to send 16 bits of data at a time—twice as much information as the original bus. This bus, called ISA, for Industry Standard Architecture, still appears in most new PCs today, although usually in combination with other types of expansion slots.

The ISA expansion slots have the advantage that you can still plug older 8-bit adapters into them. The older cards simply use fewer of the slots' connectors. But in 1987 IBM introduced the PS/2 computer, with a radically different type of bus, which IBM called the Microchannel Architecture, or MCA. It handles 32 bits of data at a time, and it has a primitive intelligence to allow it to adjust to the rest of your system automatically. This helps eliminate conflicts caused when two components want to use the same system resources, such as a specific location, or address, in memory.

MCA was a good idea. But it never really caught on for two reasons: First, it wouldn't accept the older 8-bit and 16-bit ISA expansion cards, and PC owners didn't like the idea of abandoning perfectly good adapters. Second, IBM originally didn't let other companies clone the bus as IBM had permitted with the earlier bus designs. Without other companies behind MCA, not only did the design languish, but it also inspired a countermove by seven IBM competitors. Led by Compaq, in 1988 the rivals introduced the EISA (Extended Industry Standard Architecture) bus. It provides the faster, 32-bit data flow and autoconfiguration of MCA, but an ingenious slot design lets it also use ISA cards. But EISA is complex and expensive, and it has never caught on except for use on high-end systems used for servers where every drop of speed counts.

In 1992, computer manufacturers came up with a new twist in bus design. Previously, they had concentrated on making buses push more bits of data at one time—from 8 to 16 to 32 bits. But even EISA and MCA buses were still operating, respectively, at 8.22 and 10 megahertz (MHz), despite the release of newer processors capable of churning out data at 33MHz or faster. To bring the bus up to speed, the local bus was created. “Local” refers to bus lines used by the processor. (If you think of the bus lines as being located in the neighborhood of the processor, then the term “local” makes a bit more sense.) Some of those local bus lines lead to expansion slots, giving the slot local, or direct, access to the processor. The advantage of the local bus is that it theoretically communicates with the processor at the processor's own speed. In reality, the speed for now is less, but still a terrific improvement over ISA. Typically, local bus expansion slots exist side by side with ISA slots, and the local bus slots are used for components, such as video and drive controller cards, that most affect the overall performance of the computer because they move enormous amounts of data. Local bus offers no enhancement for sending data to modems and printers, which are their own bottlenecks.

There are two versions of the local bus. The Video Electronics Standards Association (VESA) is an alliance of PC vendors who developed the VESA local bus, or VL-Bus, to accelerate video displays with working speeds up to 50MHz. Intel Corporation and other big PC companies developed the PCI (Peripheral Component Interconnect) local bus. Although PCI permits speeds up to

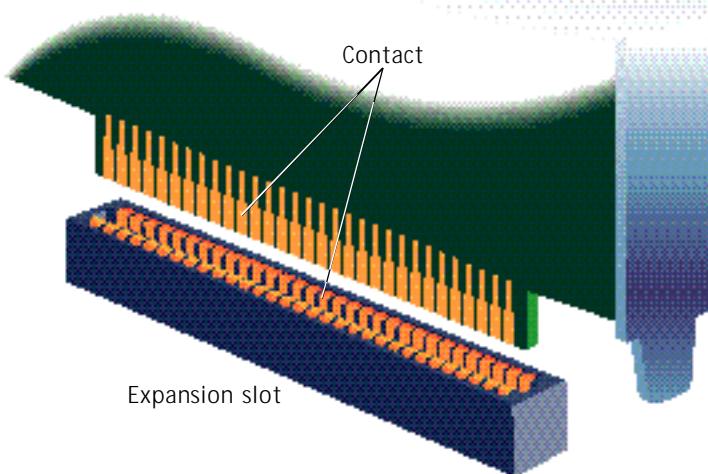
only 33MHz, PCI local bus is a more comprehensive—and expensive—design that is the first to incorporate Plug and Play setup. Despite its slower bus speed, the PCI local bus is capable of moving a maximum of 132 megabytes a second, compared to the VESA transmission rate of 107MB/sec and to the ISA transmission rate of 8MB per second.

Since the previous paragraph was written for the second edition of this book, the VL-Bus has all but passed out of sight. PCI has become the standard as mass production has reduced its cost and as video and hard drives place even more importance on moving mass quantities of data. MCA and EISA, despite their technological advantages, have become the victims of pricing and marketing and are seen only on the highest-end systems. But they're all included here for the benefit of the curious.

Even PCI local bus is showing its age. Its 132MB per second transfer rate was impressive a few years ago. Now it's becoming a bottleneck as processors such as the Pentium operating at 200+ megahertz, faster synchronous DRAM, and 3-D graphics accelerators are leaving PCI in the dust. The advanced graphics port introduced in 1997 by Intel is an improvement over the PCI bus.

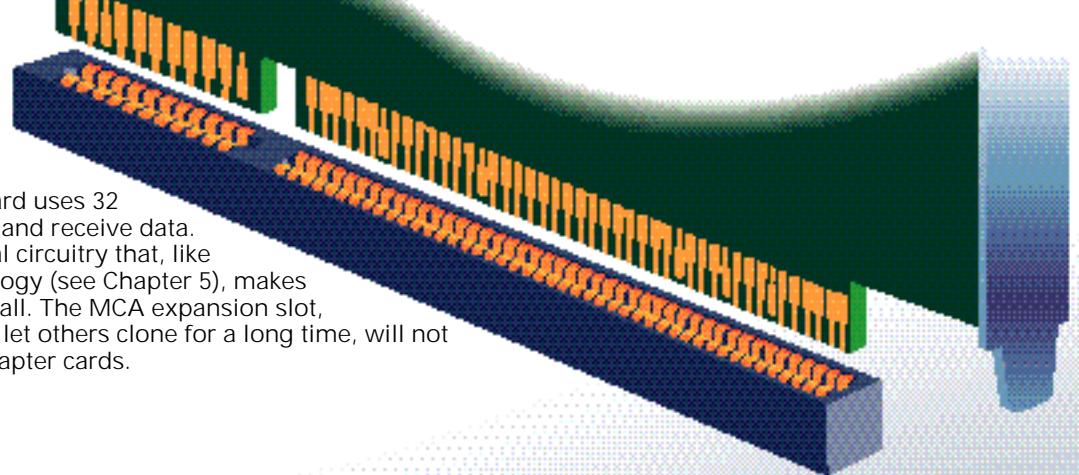
# Differences in Expansion Cards

**8-Bit Expansion Card** Data is transmitted among expansion slots and other components on the bus only along 8 parallel data lines. The data lines use only a fraction of the 31 pairs of connectors that fit into the expansion slot. As remains the same with newer boards, the other connectors supply the board with power, instructions, and addresses for data locations either on the expansion boards or in memory.

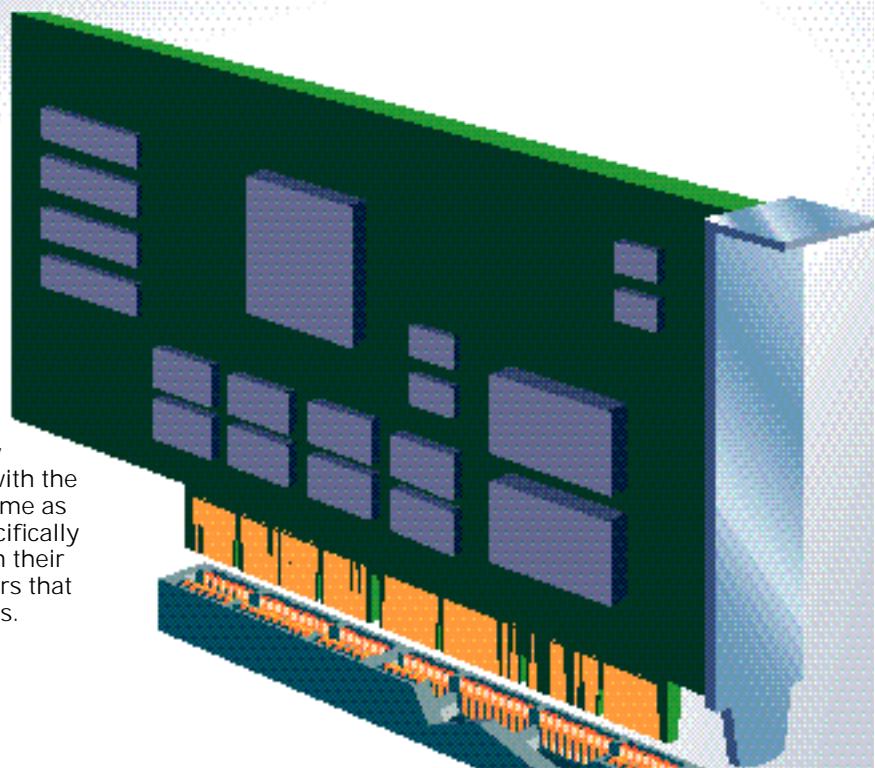


**16-Bit or ISA Card** With 18 more pairs of connectors, the ISA (Industry Standard Architecture) card transmits data over 16 data lines, doubling the amount of data it moves compared to an 8-bit card. This is the most prevalent type of expansion card, and even PCs with faster and newer local bus slots still have ISA expansion slots. A 16-bit card is powerful enough for components, such as keyboards, serial and parallel ports, and internal modems that don't handle the extreme amounts of data transmitted by video, network, and disk controller cards. This is also called an AT card because it first appeared on the IBM-AT.

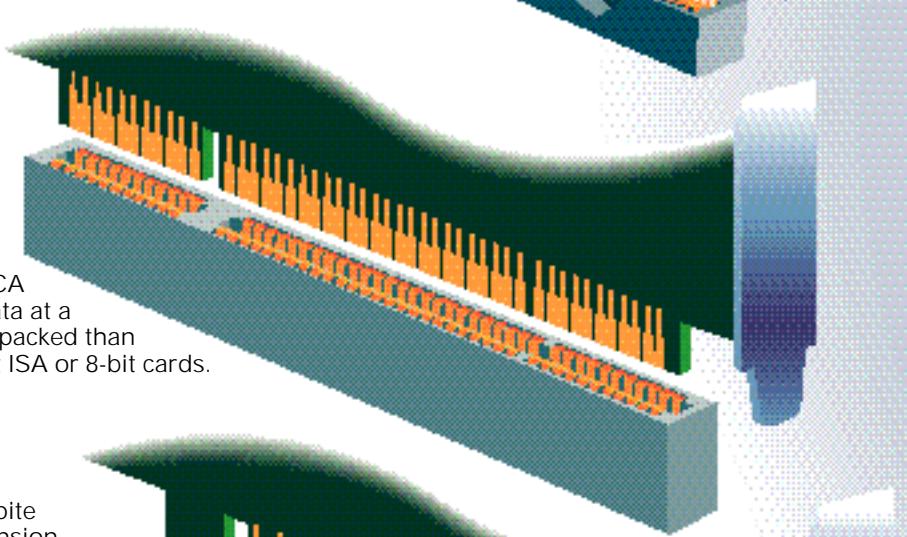
**32-Bit MCA Card** The IBM Microchannel Architecture (MCA) card uses 32 of its 93 lines to send and receive data. It also includes special circuitry that, like Plug and Play technology (see Chapter 5), makes the card easier to install. The MCA expansion slot, which IBM refused to let others clone for a long time, will not accept 8-bit or ISA adapter cards.



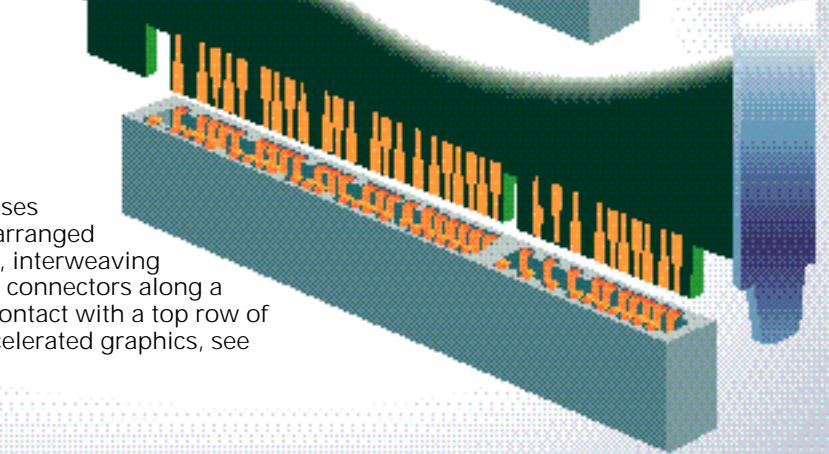
**32-Bit EISA Card** The design of Extended Industry Standard Architecture (EISA) can use expansion cards designed specifically to work with the slot's 97 connectors that are divided between two levels. These EISA-specific cards transmit 32 bits of data at a time, and, like MCA and Plug and Play, are easier to set up. But EISA slots also accept 8-bit and 16-bit cards. Plastic tabs allow the older cards to fit only far enough into the slots to make contact with the first level of connectors, which work the same as ISA connections. But boards designed specifically for the EISA slot can enter farther and align their connectors with the lower row of connectors that handle signals based on EISA specifications.



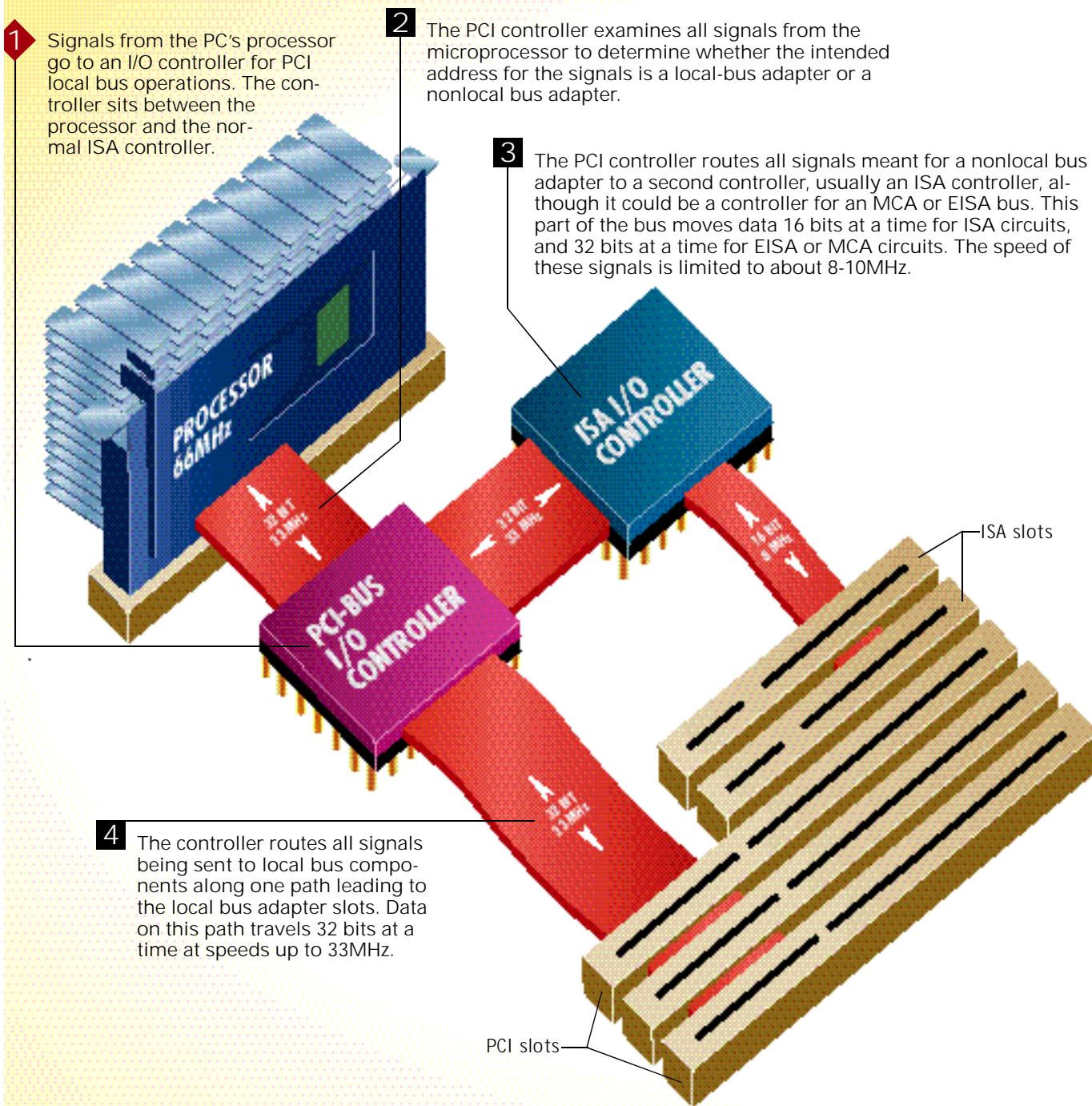
**32-Bit PCI Local-Bus Card** The Peripheral Component Interconnect (PCI) local-bus adapters have connectors similar to those on MCA and EISA cards. All handle 32 bits of data at a time, and are smaller and more tightly packed than ISA connections. PCI slots won't accept ISA or 8-bit cards.



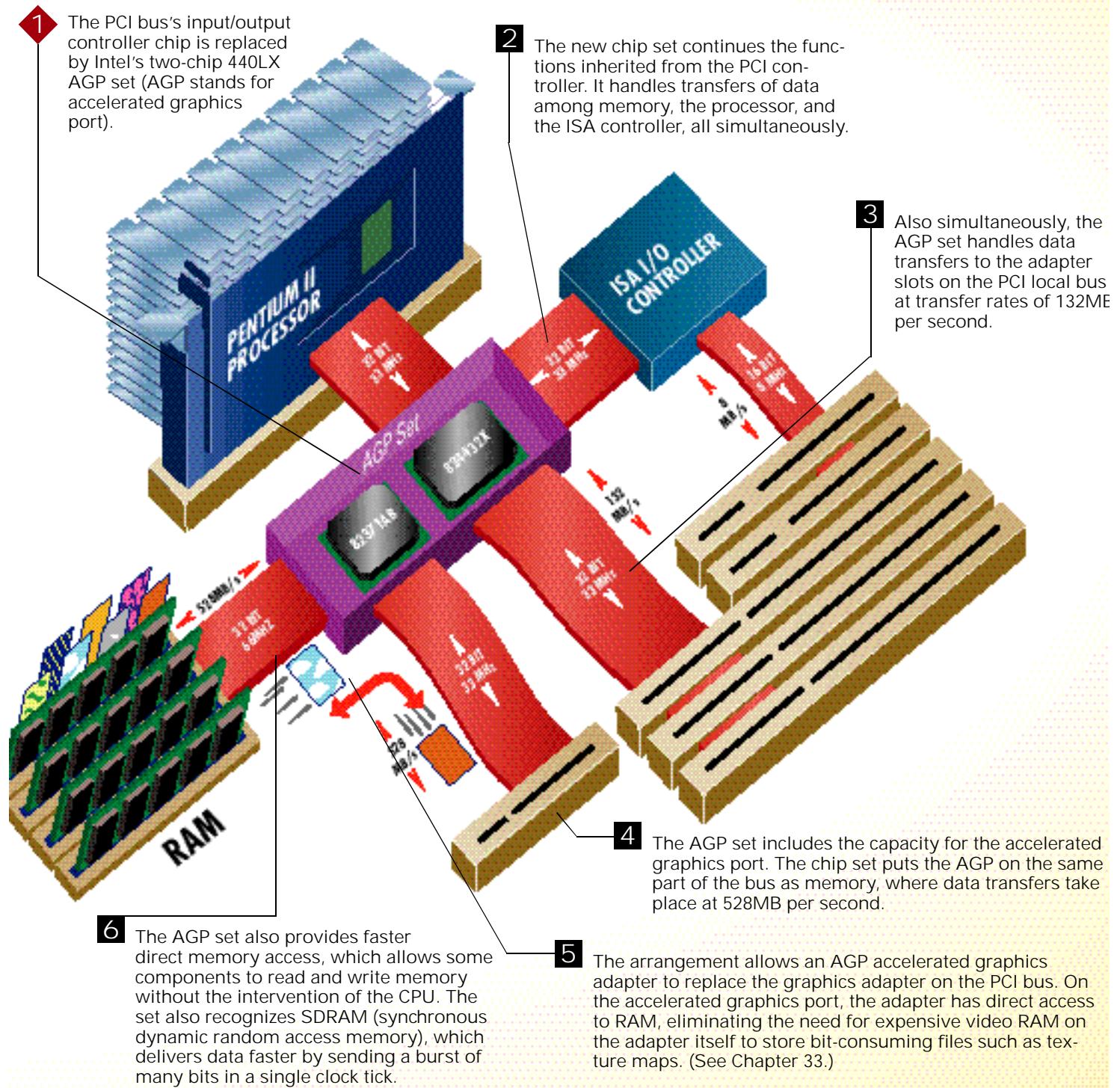
**32-Bit Accelerated Graphics Port** Despite being called a port, the AGP is an expansion slot, technically an adaptation of the PCI slot design. Unlike other expansion slots that can handle a variety of add-in cards, the AGP is designed only for a special type of video card. (This doesn't mean, however, that some enterprising engineer won't find other uses for the port.) The 44 pins on an AGP card are arranged in a fashion similar to those on an EISA board, interweaving every other connector so that some touch slot connectors along a bottom row and the remaining pins come in contact with a top row of slow connectors. For more information on accelerated graphics, see Chapter 33.



# How a PCI Local Bus Works



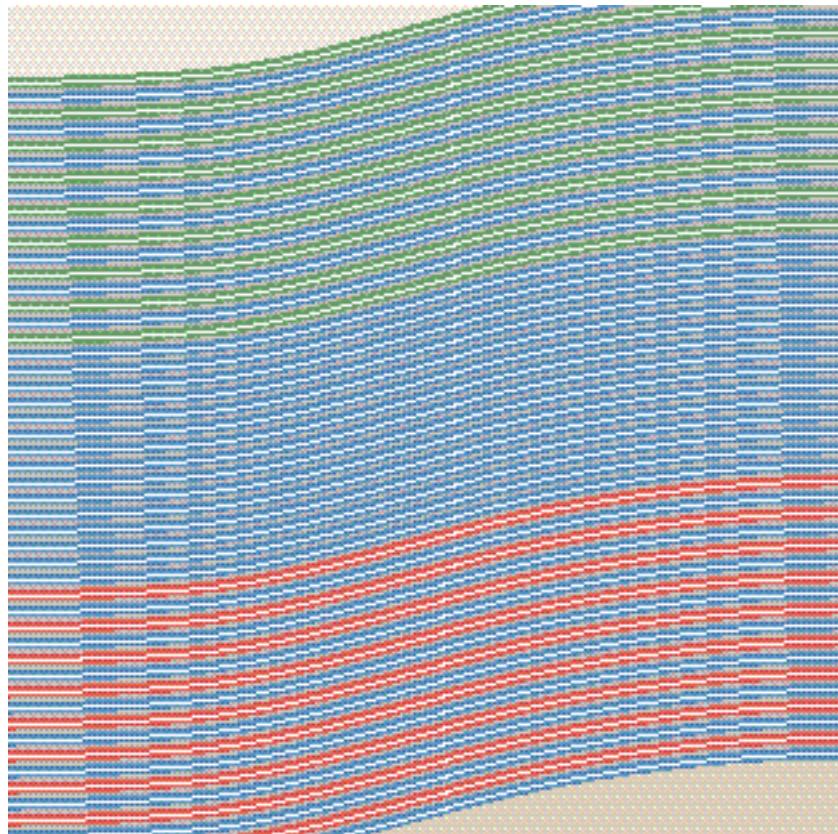
# How an Accelerated Graphics Port Works



CHAPTER

19

# How Computer Ports Work



**WITHOUT** a computer's ports—that collection of odd-shaped connectors sticking out of the back of your PC and sprouting from adapter cards, drives, printers, scanners, joysticks, or any other peripheral you hook up to a computer, much of the work that a PC accomplishes would never reach anyone other than the person sitting in front of the monitor. In fact, without some ports, the computations of your processor wouldn't even reach you. We'll first look at two of the oldest computer ports—the parallel and serial ports.

Since its introduction, the parallel port—also called a Centronics port—has been almost synonymous with printer port. Although a serial port can also be used to send data from a PC to some models of printers, the parallel port is faster. A serial port sends data one bit at a time over a single one-way wire; a parallel port can send several bits of data across eight parallel wires simultaneously. In the same time that a serial connection sends a single bit, a parallel port can send an entire byte. Another way to look at it: In the time a serial connection can send the letter A, a parallel port can send the word aardvark.

The serial port is the jack-of-all-trades among computer components. It is simple in concept: one line to send data, another line to receive data, and a few other lines to regulate how data is sent over the other two lines. Because of its simplicity, the serial port has been used at one time or another to make a PC communicate with just about any device imaginable—from commonplace modems and printers to plotters and burglar alarms. And while it may be slow compared to a parallel port, a serial port is fine for modems, which can only push signals across a telephone line serially, and for a mouse, which doesn't need to send large quantities of data.

The serial port is often referred to as an RS-232 port. RS-232 is the Electronics Industries Association's designation for a standard for how the various connectors in a serial port are to be used. The trouble is that the standard is sometimes ignored by manufacturers of peripherals and even computer makers. The fact that both 9-pin and 25-pin connectors are used as serial ports shows that we still have a long way to go before settling on exactly what constitutes an RS-232 port.

One of the drawbacks to both parallel and serial ports is that, for practical purposes, you're limited to having two of each. It's not at all hard to accumulate more external devices than parallel or serial ports. This is a limitation built into the basic input/output systems of personal computers. And the BIOS is not something you change willy-nilly. The most recent solution to this problem is the universal serial bus, which allows dozens of devices to be daisy-chained from a single port.

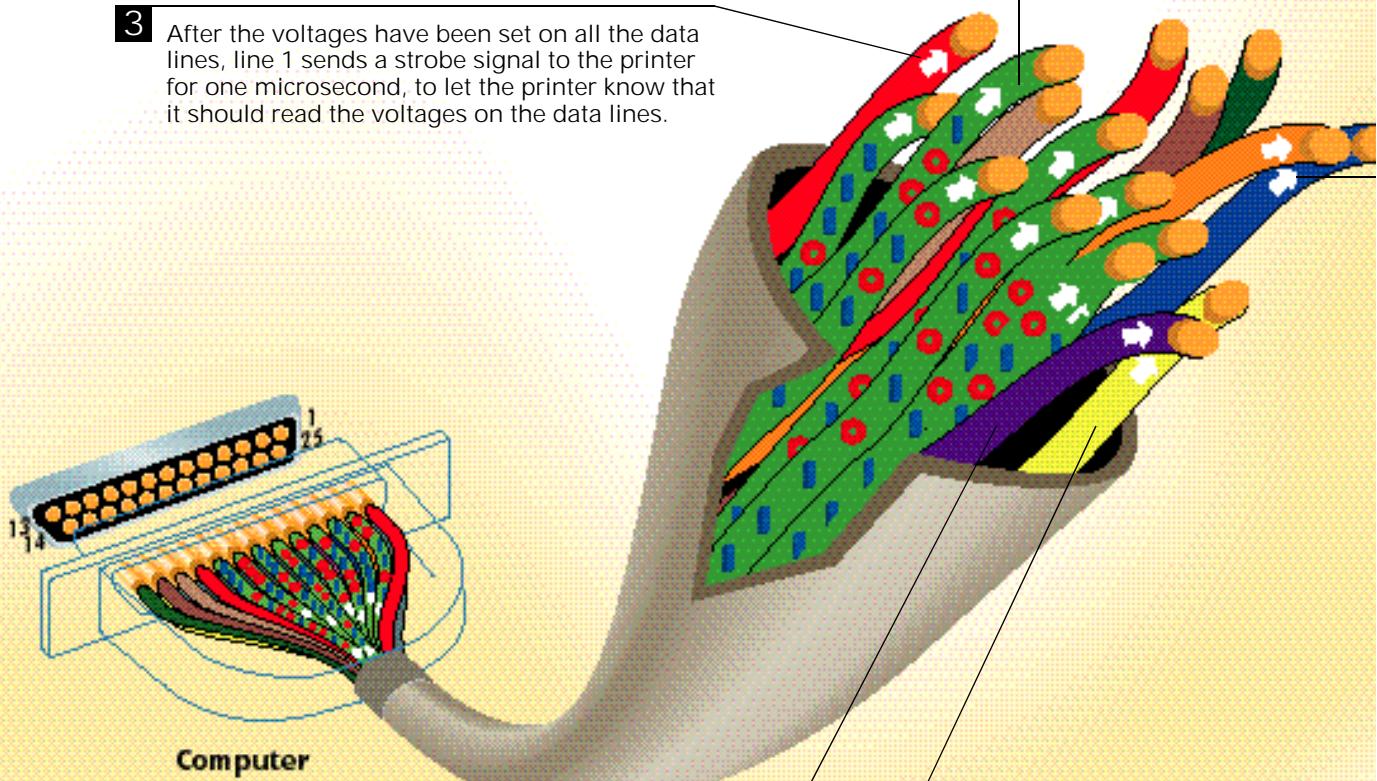
We're also looking at ports we don't usually think of as ports. The connection between your computer and its drives is a port, and there have been many designs for drive ports. We'll look at the two that have dominated in this survival of the fittest: the EIDE and SCSI ports. EIDE (Enhanced Integrated Drive Electronics) is prevalent because it is an inexpensive, dependable technology. SCSI (small computer system interface) costs more, but moves data faster and is more versatile.

# How a Parallel Port Works

**1** A signal to the PC on line 13—called the select line—from the peripheral, usually a printer, tells the computer that the printer is online and ready to receive data.

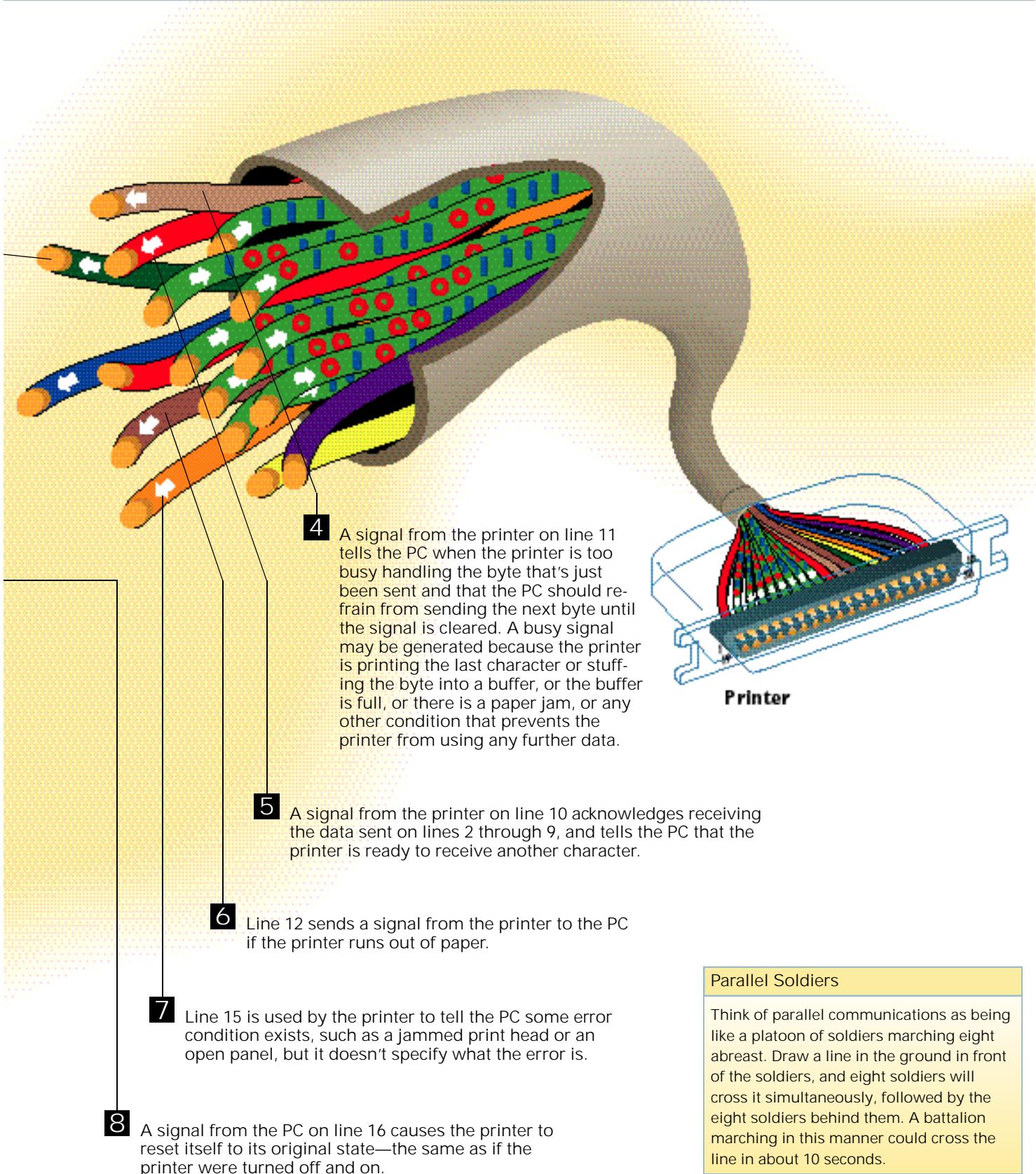
**2** Data is loaded on lines 2 through 9, shown here as green, in the form of a “high” voltage signal—actually about five volts—to signify a 1, or a low, nearly zero voltage signal to signify a 0.

**3** After the voltages have been set on all the data lines, line 1 sends a strobe signal to the printer for one microsecond, to let the printer know that it should read the voltages on the data lines.



**10** A signal from the PC on line 17 tells the printer not to accept data. This line is used only with some printers, which are designed to be switched on and off by the PC. Lines 18 through 25 are simply ground lines.

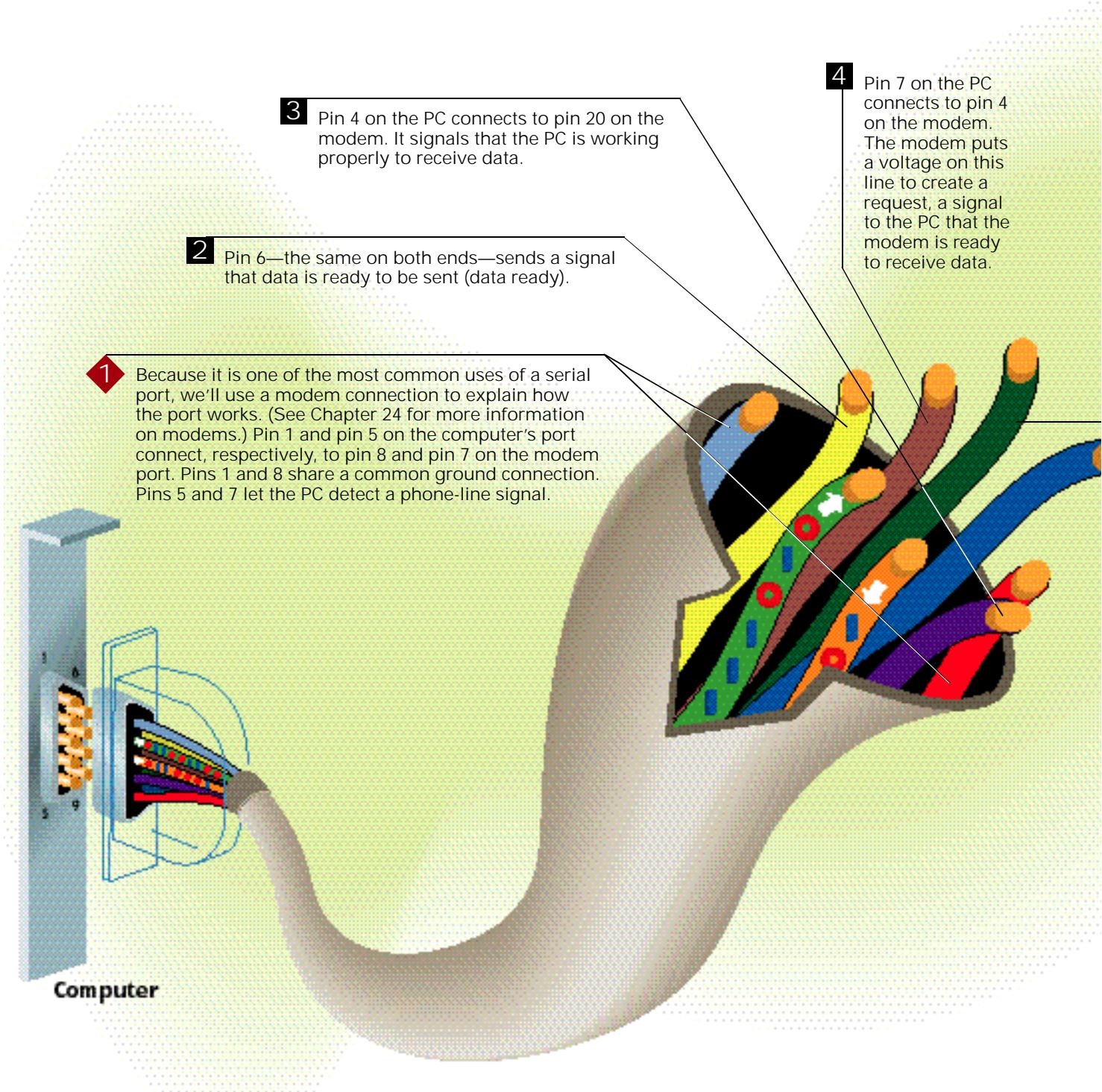
**9** A low-voltage or zero-voltage signal from the PC on line 14 tells the printer to advance the paper one line when it receives a carriage return code. A high-voltage signal tells the printer to advance the paper one line only when it receives a line-advance code from the printer.

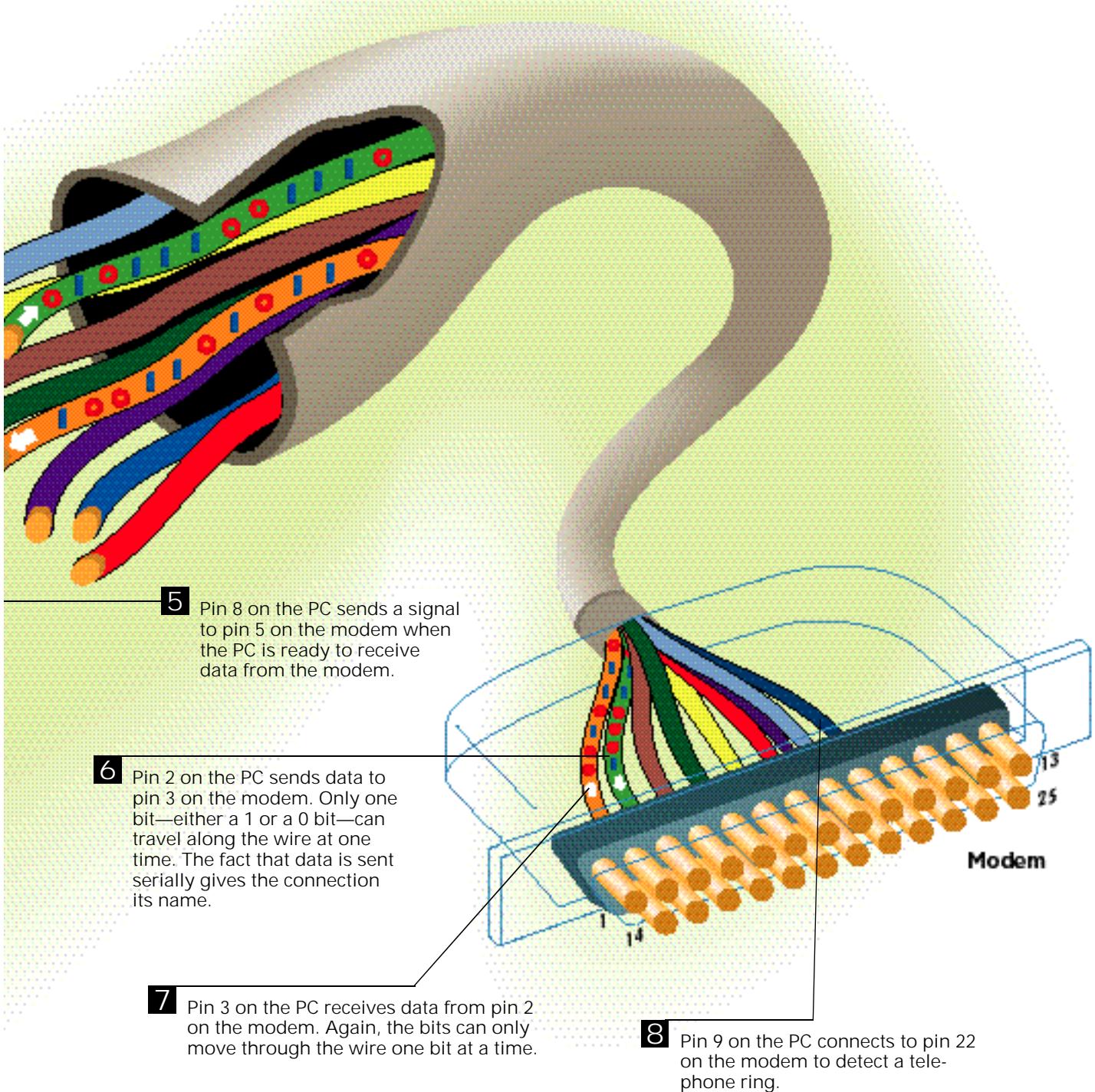


#### Parallel Soldiers

Think of parallel communications as being like a platoon of soldiers marching eight abreast. Draw a line in the ground in front of the soldiers, and eight soldiers will cross it simultaneously, followed by the eight soldiers behind them. A battalion marching in this manner could cross the line in about 10 seconds.

# How a Serial Port Works

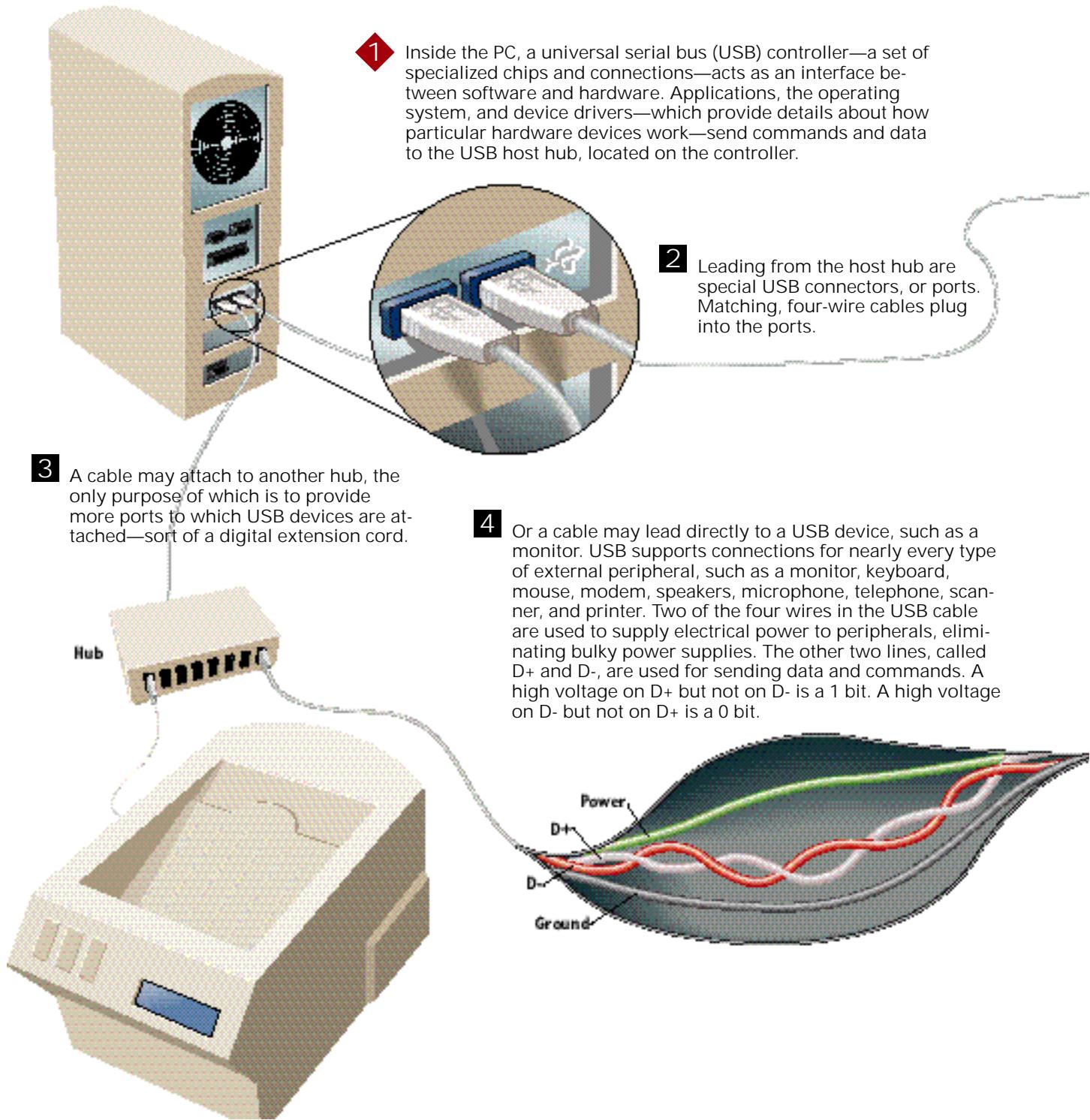




#### Soldiers All in a Row

A serial connection is comparable to soldiers lined up in a single row. Only one of them at a time would be able to cross a line drawn on the ground. A battalion would take more than a minute for all the soldiers to cross the line serially.

# How the Universal Serial Bus Works



- 5** Any USB device may also include a hub, so that a monitor, for example, provides ports into which multimedia speakers, a microphone, and a keyboard can be plugged.



- 8** Working with similar Plug and Play technology that allows automatic configuration of internal PC components, the USB host controller tells the new device to identify itself, finds out what it requires for sending and receiving data, and assigns the device an identification number.

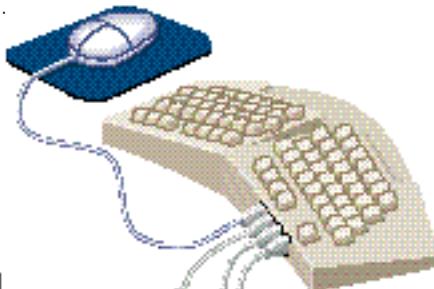
ISDN MODEM  
HIGH SPEED  
ID: DEVICE 10

- 10** USB can work with three types of data transfers, and assigns bandwidth priorities in the following order:



Highest priority—Isochronous, or real-time, where there can be no interruption in the flow of data, such as video or sound.

- 6** These devices can, in turn, provide ports for further USB hardware. For example, a mouse and a digitizing pen could attach to the keyboard, attached to the monitor, attached to the host hub. This system of branching connections lets the universal serial bus handle up to 127 devices.



- 7** When a new USB device is plugged into a port, it automatically causes a voltage change on one of the two data wires. If the voltage is applied to D+, the peripheral is saying it's a high-speed device, capable of sending 12 megabits a second, used for monitors, scanners, printers, and other devices that send a high volume of data. A voltage on D- indicates it can get by with a slow transfer speed of 1.5Mbps, say, for a keyboard or mouse. (A conventional serial port, in comparison, sends only 100 kilobits a second; a parallel port about 2.5Mbps.)

- 9** Now that the new device is an official member of the bus, it takes its place as the host controller polls the devices—to issue commands, to ask if the device is ready to send or receive data, and to apportion chunks of the bus's bandwidth (data-transmitting capacity) to each device. About a million times a second, the controller sends queries or commands downstream to all the peripherals on the USB. Each of the host's messages begins with a token that identifies which peripheral it's addressed to. The message goes to all devices on the bus, but devices that don't match the token's address simply ignore it. Devices send data upstream to the host only after the host gives them permission.



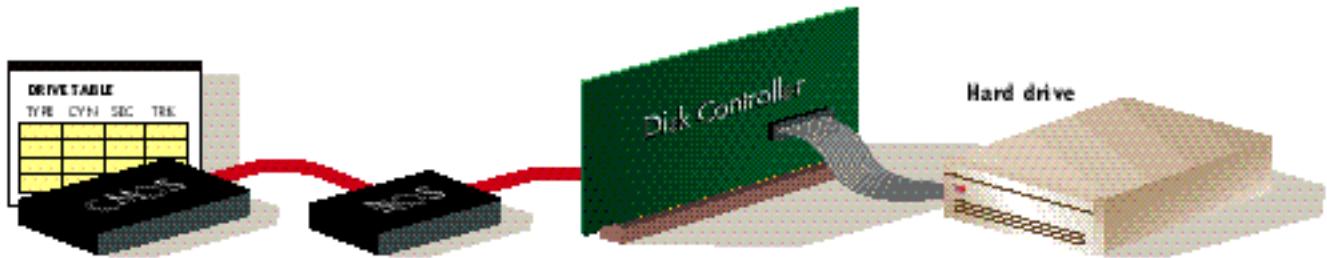
Second highest priority—Interrupt transfers, which occur only when a device, such as a keyboard or joystick, generates an occasional interrupt signal to get the processor's attention.



When time permits priority—Bulk transfers of data for printers, scanners, and digital camera in which there's a lot of data to send, but no particular hurry to get it there.

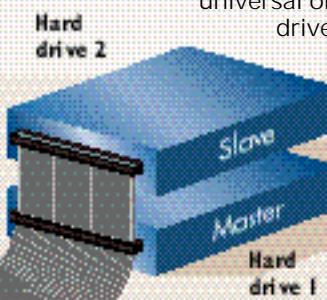
# How IDE Connections Work

- 1** Originally, personal computers knew how to handle only a few types of hard drives—14 at first, later 30—because the information on how to control each of them was embedded in the PC's CMOS. When a drive was installed, its number of tracks, sectors, and platters—its geography—had to match one of the configurations the CMOS could recognize. The BIOS uses the CMOS information to direct the movement of the write/read head and to route data between the processor and the drive. The drive controller expansion card passed the signals to the drives and handled the transfer of data between the processor and the drive.



- 2** To overcome this limitation in the evolution of hard drives, Integrated Drive Electronics (IDE) was developed to replace both the disk controller card and the CMOS-based knowledge of every possible drive's physical geometry. Today virtually all PCs come with two IDE or E-IDE (Enhanced-IDE) controllers built into the motherboard, connected in such a way that the computer sees them as a fictional ISA expansion slot 9.

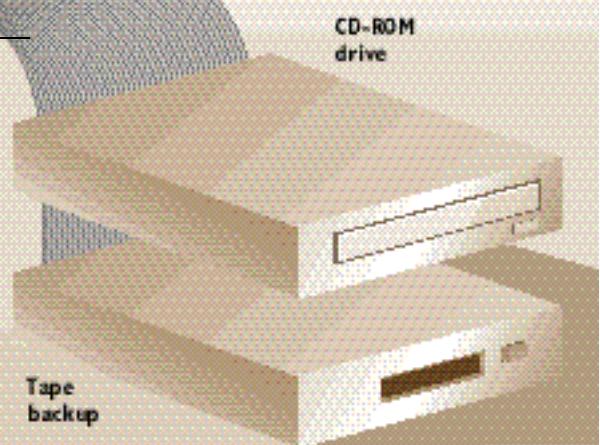
- 3** A 40-wire ribbon cable that runs from an IDE connector has two other connectors on it, each of which can connect to a hard drive. With two IDE connectors on the motherboard, a PC can have up to four hard drives. E-IDE, which is nearly universal on new PCs, can control floppy drives, CD-ROM drives, and tape drives, although they must be IDE-compatible and mounted inside the PC. For each pair of devices on the same cable, one is configured as the master and the other as the slave.



- 4** The operating system issues a generic command to one of the IDE devices as if it were one of the drives the PC knows about in the CMOS's drive table. The IDE connector does little more than pass along the signals the operating system thinks it's sending to the drive controller.

- 5** Commands and data, sent serially one bit at a time, pass through the cable to both devices connected to the cable.

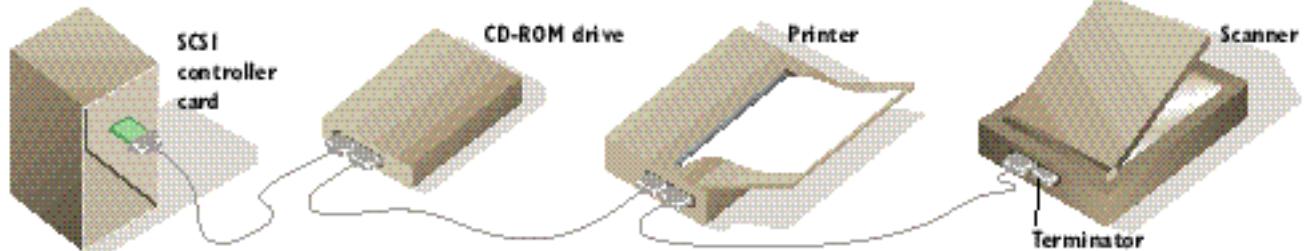
- 6** Control signals tell the master and slave which of them the commands and data are intended for.



- 7** The selected drive translates the commands for the vanilla CMOS drive so they will work with the real geometry of the drive. The drive controls all input and output and the positioning of the write/read head—functions that used to belong to a controller card.

# How SCSI Works

- 1** A SCSI (small computer system interface) connection is the fastest and most versatile way for a PC to communicate with a wide variety of peripherals—hard drives, CD-ROM drives, scanners, printers, and other devices. A single SCSI controller manages up to seven devices through a daisy-chain connection. Some versions of SCSI can run 15 devices off one SCSI host card, but for simplicity, we'll look at a version that controls no more than seven peripherals.

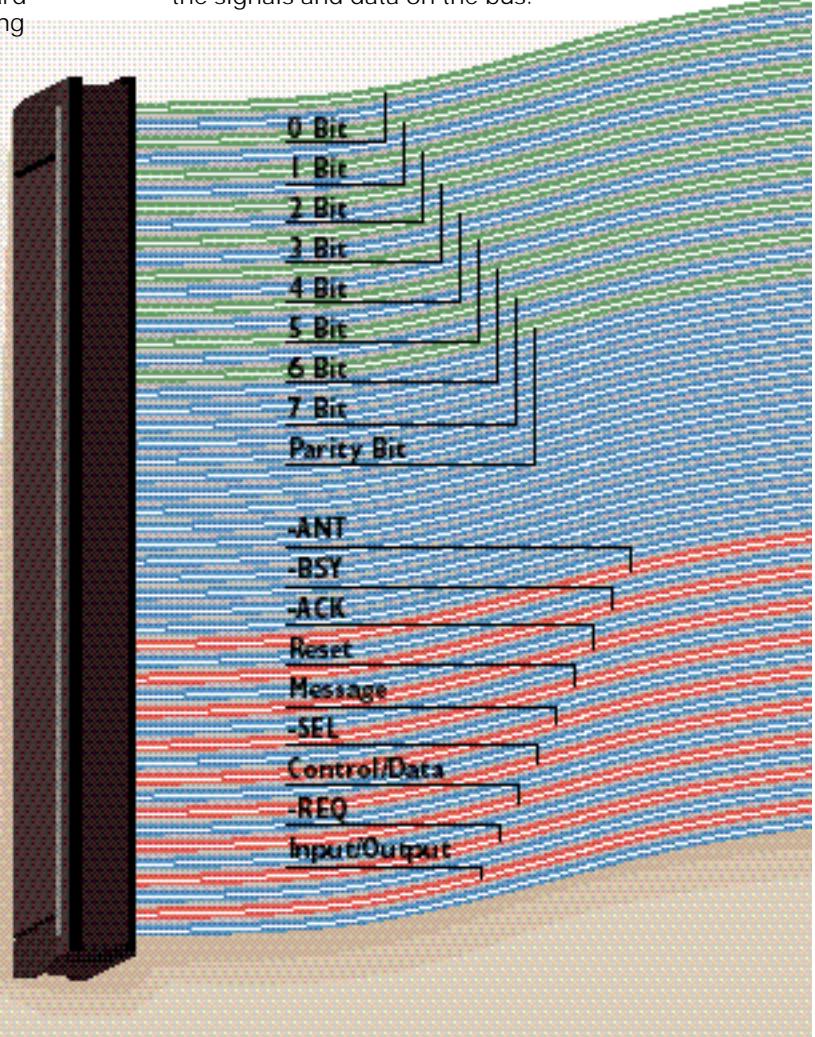


- 2** Each device on the daisy chain, called the SCSI bus, has a unique identifying number from 0-7. A device's ID number is assigned when the device is installed, by setting switches either physically or through software. The SCSI card usually assigns itself the 7 ID number, leaving 0-6 for other devices.

- 4** Signals from the controller pass from one device to another along the SCSI bus, which consists of 50 wires or lines. Of those 50 wires, SCSI uses a few to send information and control signals. The remaining wires, shown here in blue, are matched with each of the active wires and are intricately wrapped in external cables to protect the working wires from electrical interference. (For simplicity, the wires are shown here lying side by side, which usually only occurs in a ribbon cable for a SCSI device mounted inside a PC.) All devices share the same data lines, shown in green, and control lines (red). But only two devices can communicate with each other at one time. The most common forms of SCSI use 8 lines to send data at a rate of 10-20 megabytes a second. Wide-SCSI uses a 16-bit data path, boosting data transfers to as high as 40 megabytes a second. IDE drive controllers, in comparison, can transfer data at 5.5 megabytes a second under ideal conditions.

[Continued on next page.]

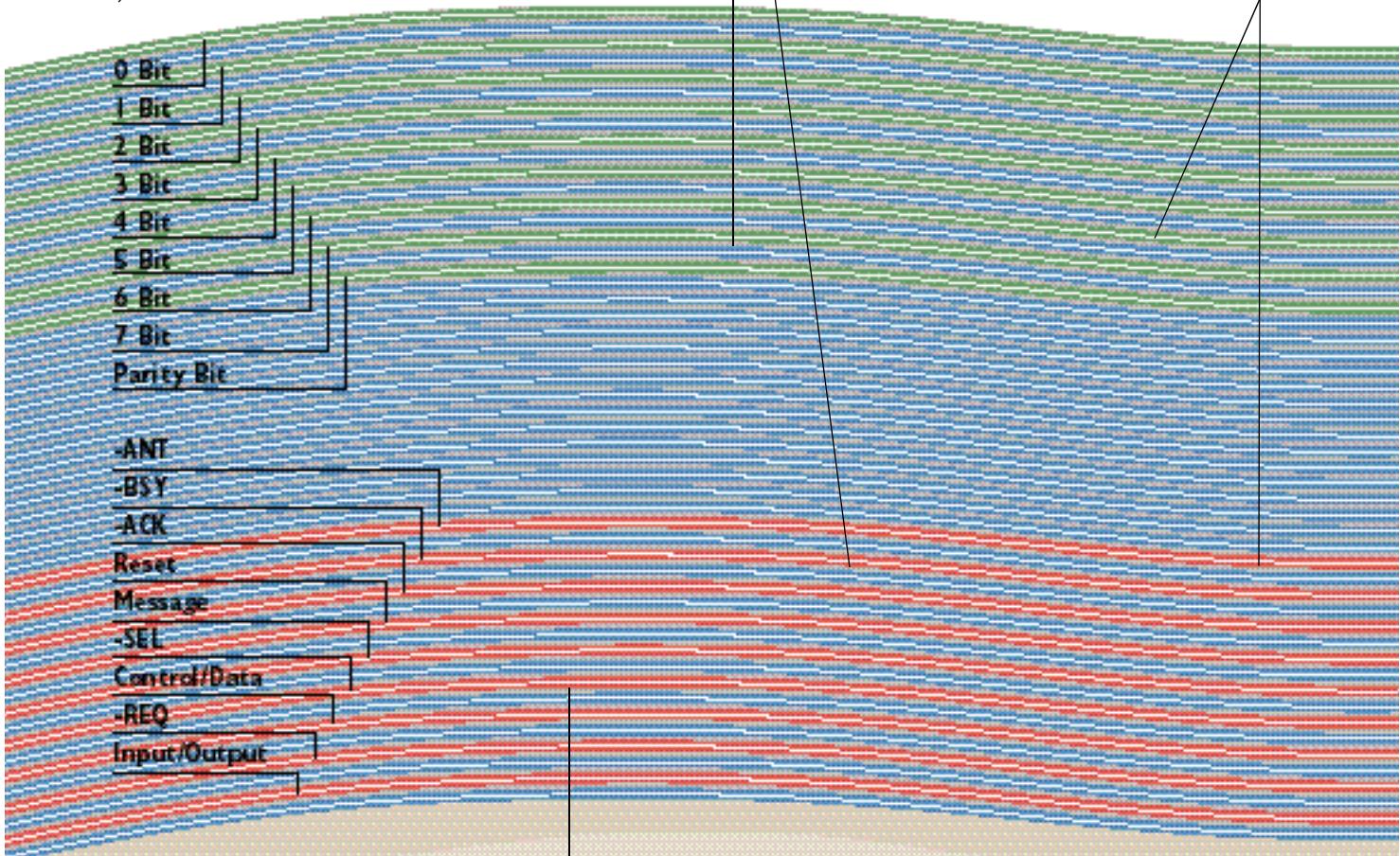
- 3** The last device on the daisy chain must have a terminator installed on the connector that's not being used. The terminator grounds the wiring in the cable to muffle spurious electromagnetic fields that could distort the signals and data on the bus.



# How SCSI Works

(Continued)

**5** When the application software and operating system tell a SCSI controller to operate one of the peripherals, the controller becomes an initiator, and the other devices become targets. A peripheral such as a hard drive may sometimes inaugurate communications. Then the drive is the initiator and the controller card becomes the target. To get control of the bus, the controller waits for a pause in the message traffic, and asserts control of the bus by sending a burst of current down the 36<sup>th</sup> wire, called the -BSY or busy line. At the same time, it sends a current along the wire that carries the seventh bit of data. This tells the bus that device 7 (the SCSI card) wants control of the bus.



**7** Next, in the selection phase, the controller says which device it wants to communicate with by turning on line 32, the -ANT or attention line. At the same time, it sends a current on the line that syncs up with the ID of the target peripheral. In this example, the host controller wants to work with the CD-ROM drive, which happens to have an ID of 5. Because SCSI makes every other line a ground wire, the wire that corresponds with ID 5 is actually line 12.

**6** If another device tries to get control at the same time, the bus goes into an arbitration phase, in which the devices compete for control by seeing which has the highest ID number. After 2.4 microseconds—two and a half millionths of a second—the device with the highest ID number asserts its dominance by sending a current along line 44—the -SEL or select line. The SCSI controller normally is assigned ID 7, to make sure it's always the boss.

**9**

Unlike older controllers that are closely matched with specific peripherals and control them directly, a SCSI controller can be relatively ignorant about the device it's managing. It doesn't need to know the details—the number of tracks, heads, and sectors, for example, of the device it's sending commands to. Instead, it places the first byte of a command descriptor block (CDB) on the 8 data lines. (Two bytes for 16-bit SCSI.) A CDB is a generic command such as "read so many sectors of a particular file." The controller doesn't have to know where the file is on the drive. That's the target device's responsibility.

**8**

The CD-ROM drive turns on the current in line 48, the request or -REQ line. This tells the host controller, in effect, "I hear you. What do you want me to do?" Importantly, the target device remembers the number of the initiator in case the target later disconnects from the controller.

**10**

The controller turns on the current in line 38, the acknowledge or -ACK line. This tells the target that the first CDB is waiting on the data lines.

**11**

The target reads the data byte and turns off the current on the -REQ line to tell the host controller it got the data.

**12**

The controller, in turn, switches off the -ACK line to acknowledge the target's acknowledgment. All this communication back and forth is to eliminate errors that could be caused by different timing within the two devices. By constantly keeping in touch with each other, both devices know their data is in sync.

**14**

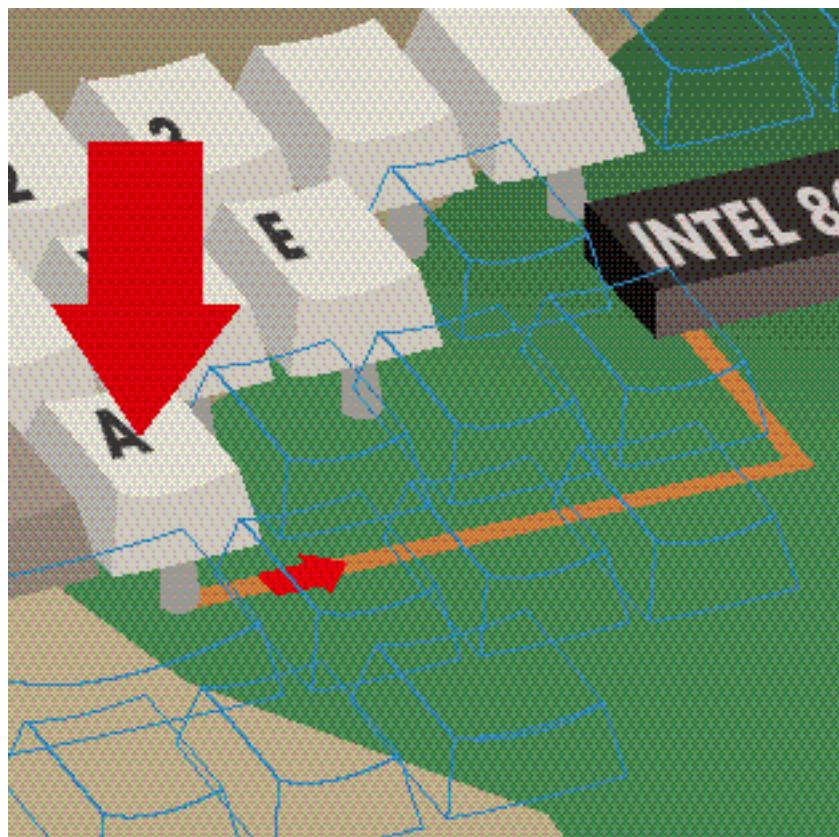
When the target has the data that the controller requested, it initiates a connection with the controller, and places the data on the bus to be read by the controller.

**13**

This type of exchange repeats itself until the target receives enough CDBs to decipher a complete, coherent command. If the target determines that the request will take a relatively long time, it signals the controller to break the connection between them so that other SCSI devices can communicate with the controller. SCSI doesn't require the CPU to direct things personally, freeing the main processor to multitask other operations.

CHAPTER  
**20**

# How a Keyboard Works



YOU come into direct contact with your PC's keyboard more than you do any other component. You may go for years without ever thinking about—much less touching—your PC's processor or hard drive, but most people pay much more attention to those components than they do to the one part of the computer that determines not how well the computer works, but how well they themselves work.

A poorly designed keyboard acts as a constant stumbling block to productivity and can even cause health problems. A well-designed keyboard is one that you never think about; your thoughts seem to flow directly from your mind to the computer's screen without you being aware of what your fingers are doing.

Despite the importance of the keyboard, most manufacturers—and too many users—pay little attention to it. In 1996, Microsoft made the biggest change in the keyboard since the function keys moved from the left side to the top: Microsoft's split-board design made a concession to ergonomics by splitting the layout in half and angling the halves so they're in line with how our arms rest naturally on a desktop. The design has been widely copied, but it's too early to tell whether the design will eventually completely replace the older arrangement.

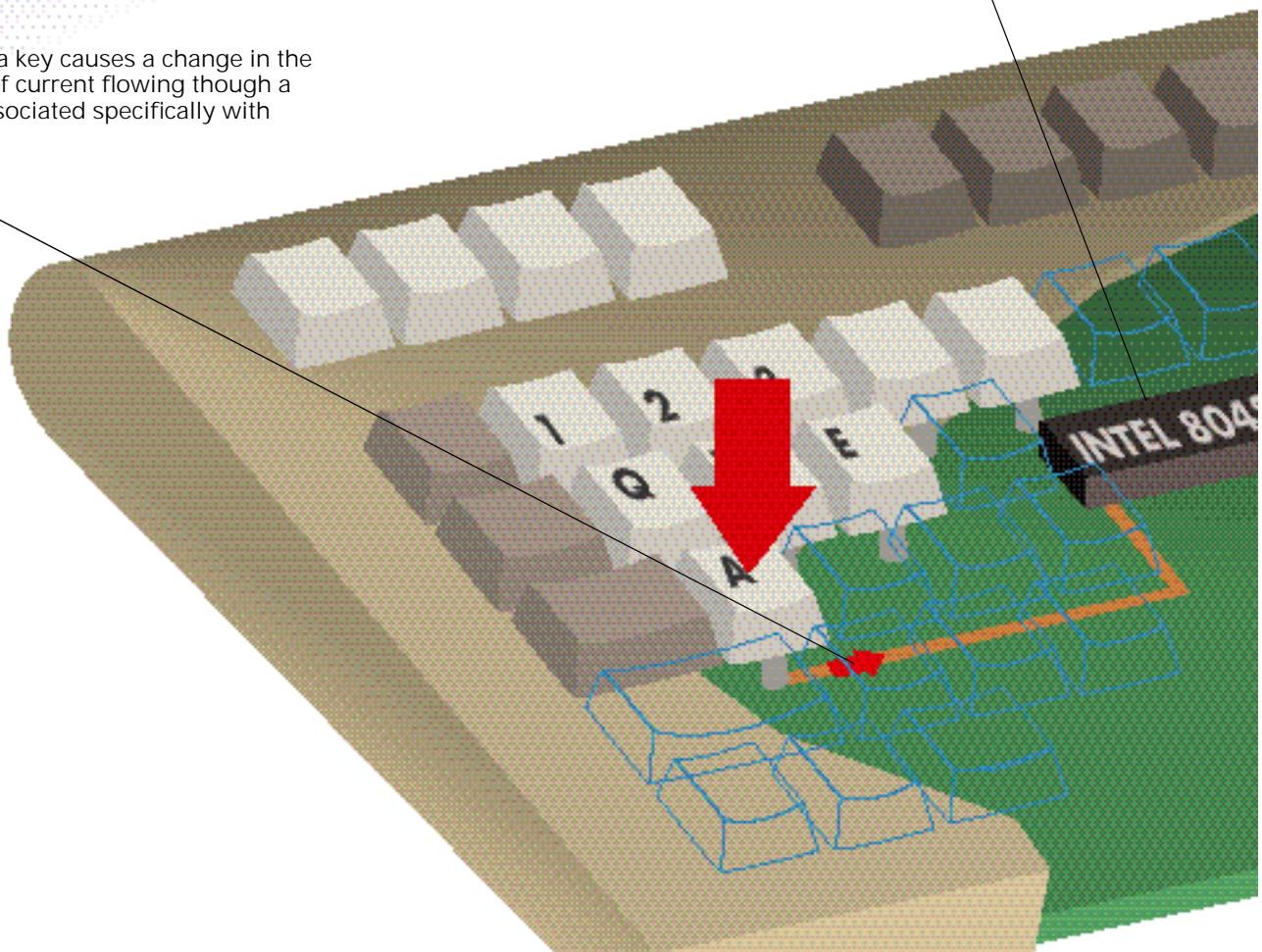
Advances have been made in keyboard design in some other clever ways. One improvement integrates a scanner to give you the choice of feeding in text by typing or scanning. Built-in pointing devices, including trackballs, touchpads, and eraserhead pointers between the G and H keys, are commonplace, if not standardized.

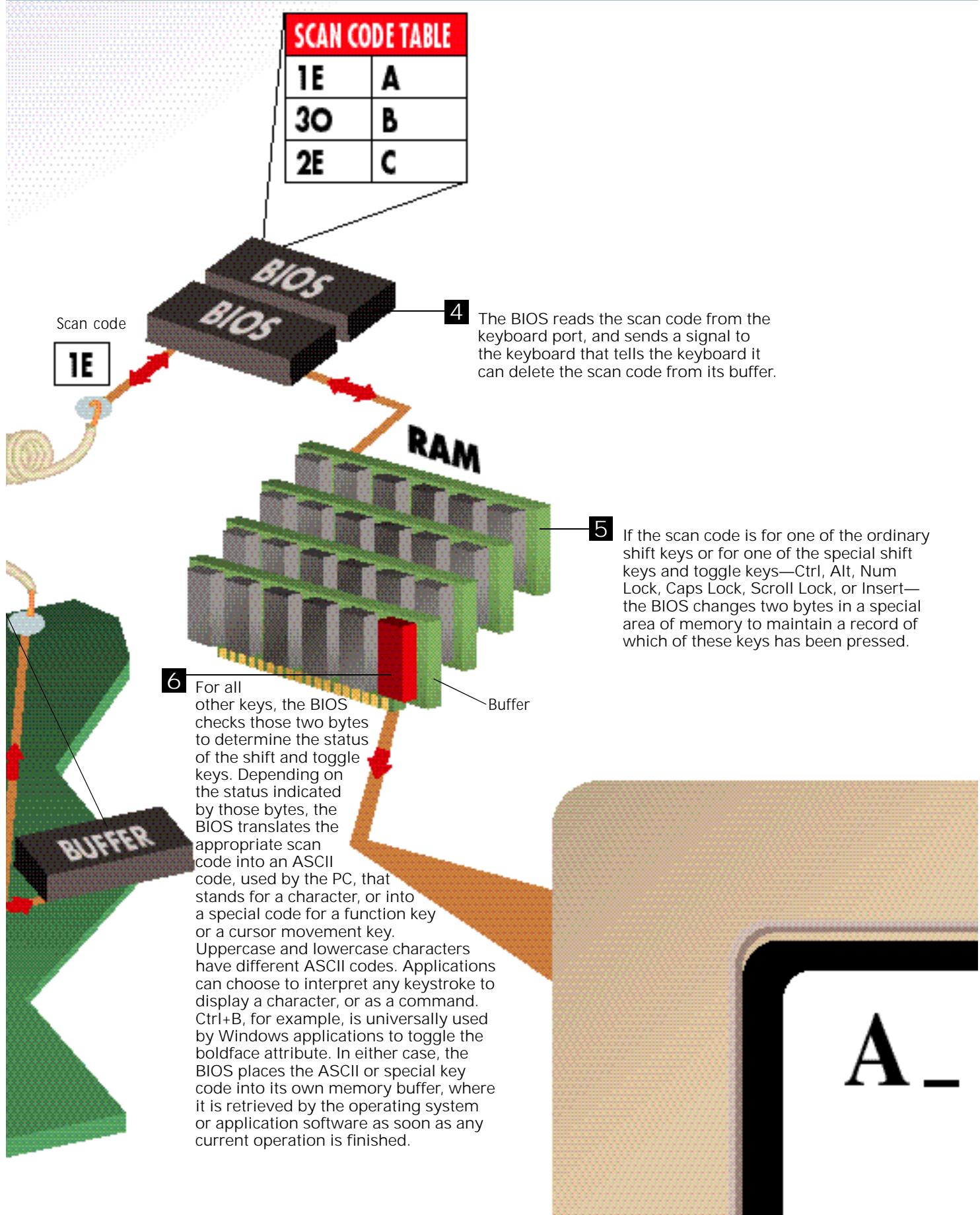
Regardless of changes in layout, the basic way a keyboard works has not changed significantly since the first IBM PC was introduced in the early 1980s.

# The Keyboard and Scan Codes

- 3 Depending on which key's circuit carries a signal to the microprocessor, the processor generates a number, called a scan code. There are two scan codes for each key, one for when the key is depressed and the other for when it's released. The processor stores the number in the keyboard's own memory buffer, and it loads the number in a port connection where it can be read by the computer's BIOS (basic input/output system). Then the processor sends an interrupt signal over the keyboard cable to tell the processor that a scan code is waiting for it. An interrupt tells the processor to drop whatever else it is doing and to divert its attention to the service requested by the interrupt. (See Chapter 3.)
- 2 A microprocessor built into the keyboard, such as the Intel 8048, constantly scans circuits leading to the keys. It detects the increase and decrease in current from the key that has been pressed. By detecting both an increase and a decrease in current, the processor can tell when a key has been pressed and when it's been released. Each key has a unique set of codes, even if, to the users, the keys seem identical. The processor can, for example, distinguish between the left and right shift keys. To distinguish between a real signal and an aberrant current fluctuation, the scan is repeated hundreds of times each second. Only signals detected for two or more scans are acted upon by the processor.

1 Pressing a key causes a change in the amount of current flowing though a circuit associated specifically with that key.

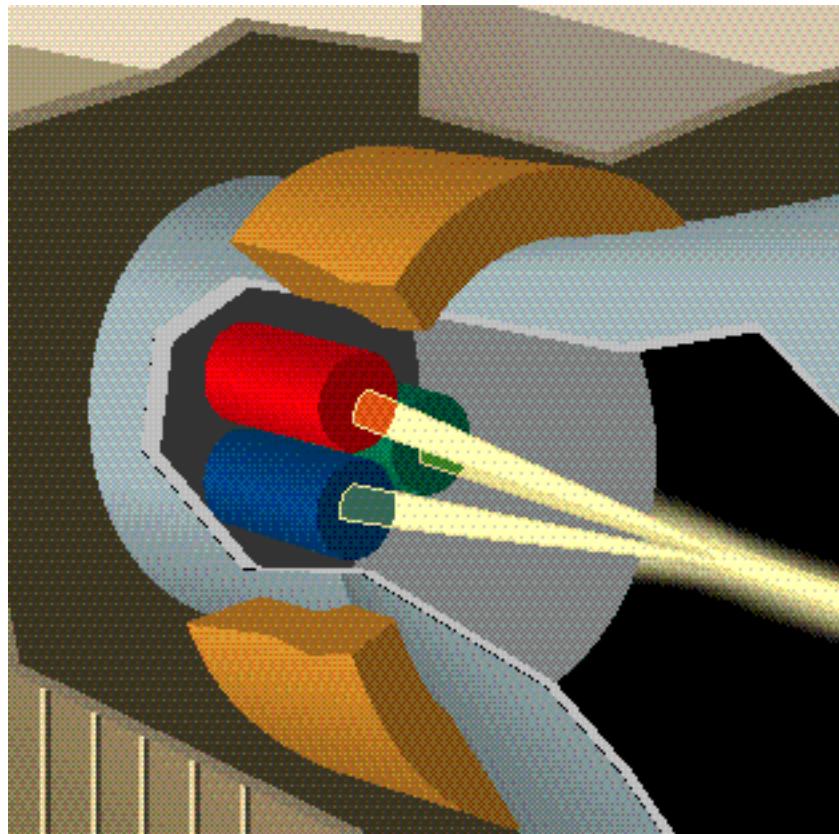




CHAPTER

21

# How a Computer Display Works



**WHEN** you read the Sunday funnies, you're looking at a hardcopy version of the way a computer displays graphics. Put a magnifying glass to the color comics and you'll see that they are made up of hundreds of dots of red, blue, yellow, and black ink. Different colors and shades are created by varying the sizes of the dots, called Ben Day dots. Large red and yellow dots and small blue dots create a shade of orange. Increase the size of the blue dots in the same area, and the color becomes brown.

If you look at a comic strip too closely, you see the dots themselves, rather than the image they are creating. But hold the comics away from you, and the dots resolve themselves into a single image.

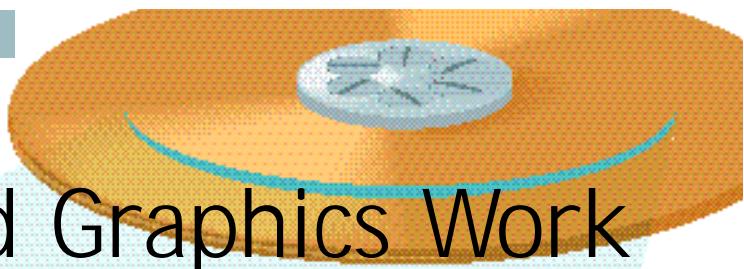
A PC monitor works the same way but uses green instead of yellow, and it's additive color as opposed to printed color's subtractive process. Glowing dots of red, green, and blue chemicals blend into millions of colors.

If you were to study a comic strip and make a meticulous record of the position, size, and color of each dot, you would in effect create a noncomputerized version of the most common form of computer graphics, a bitmap.

As its name implies, a bitmap contains a specific map of all the bits of data—location and color information—that create a computer image by changing the colors in specific pixels on a monitor. (Pixel stands for picture element, the smallest area of a monitor's screen that can be turned on or off to help create an image.) Bitmaps can be displayed quickly and they're useful when an image is static, as Windows's icons and wallpapers are.

Windows uses graphics and color, largely in the form of bitmaps, to create an interface between you and the operating system. They don't simply make Windows prettier, they convey more information than black-and-white text.

We'll see how computers store and display bitmaps and vector graphics, which can adapt themselves to size changes and the movement required by 3-D animation. We'll also look at the two most common ways of displaying output from a computer—the super-VGA monitor and the color liquid crystal display—plus a promising display of the future: gas plasma.

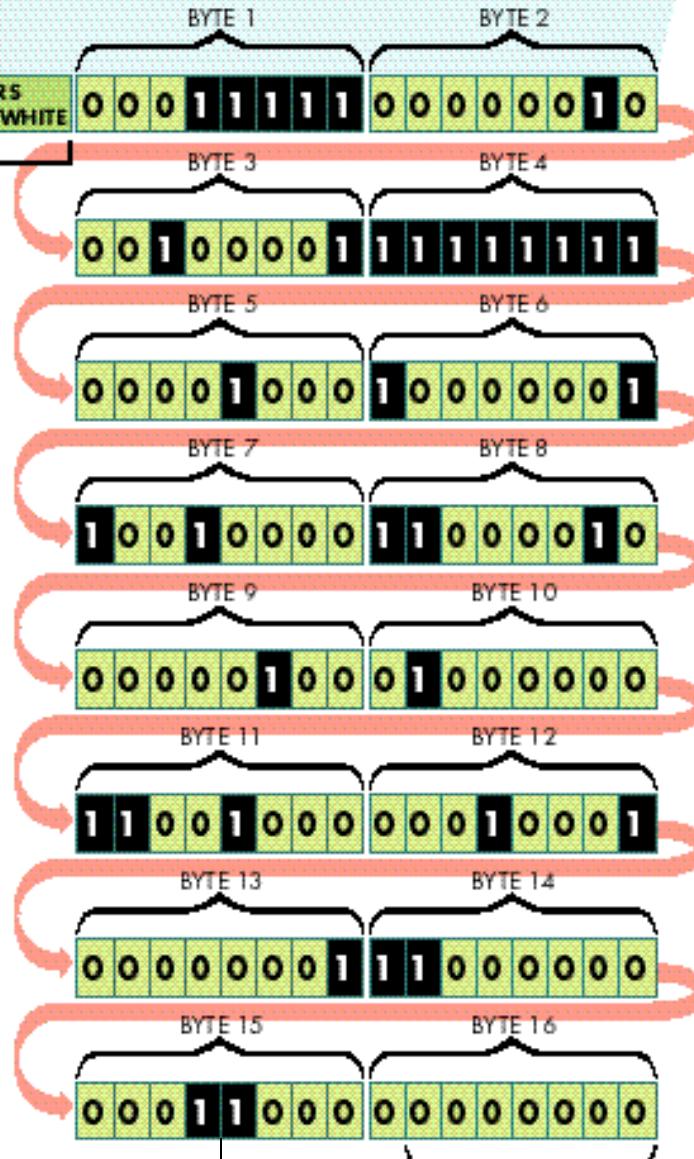
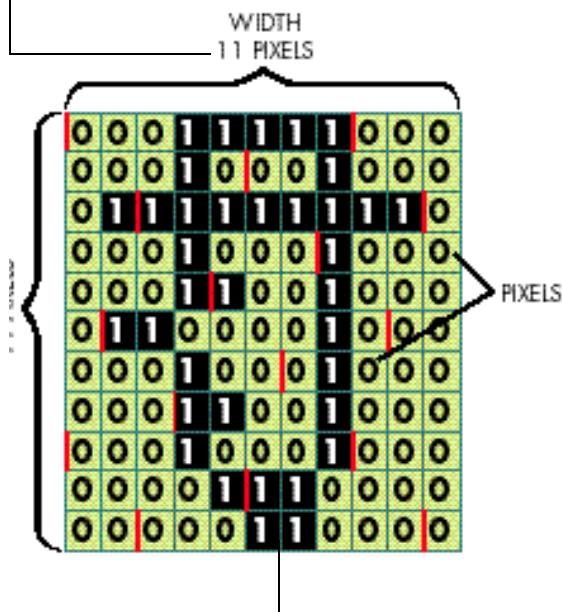


# How Bitmapped Graphics Work

- 1** When a graphics program reads a bitmapped file from a drive, it first looks at information contained in the file's header, which is several bytes at the beginning of the file that contain information the program needs to interpret the data in the rest of the file. The header begins with a signature that identifies the file as a bitmap. You don't see this signature, but you can tell that a file is a bitmap if it has an extension such as .BMP, .PCX, .TIF, or .JPG. Following the signature, the header tells the width and height of the image in pixels, which are distinct points of light, and then defines the palette (how many and which colors are used in the image).



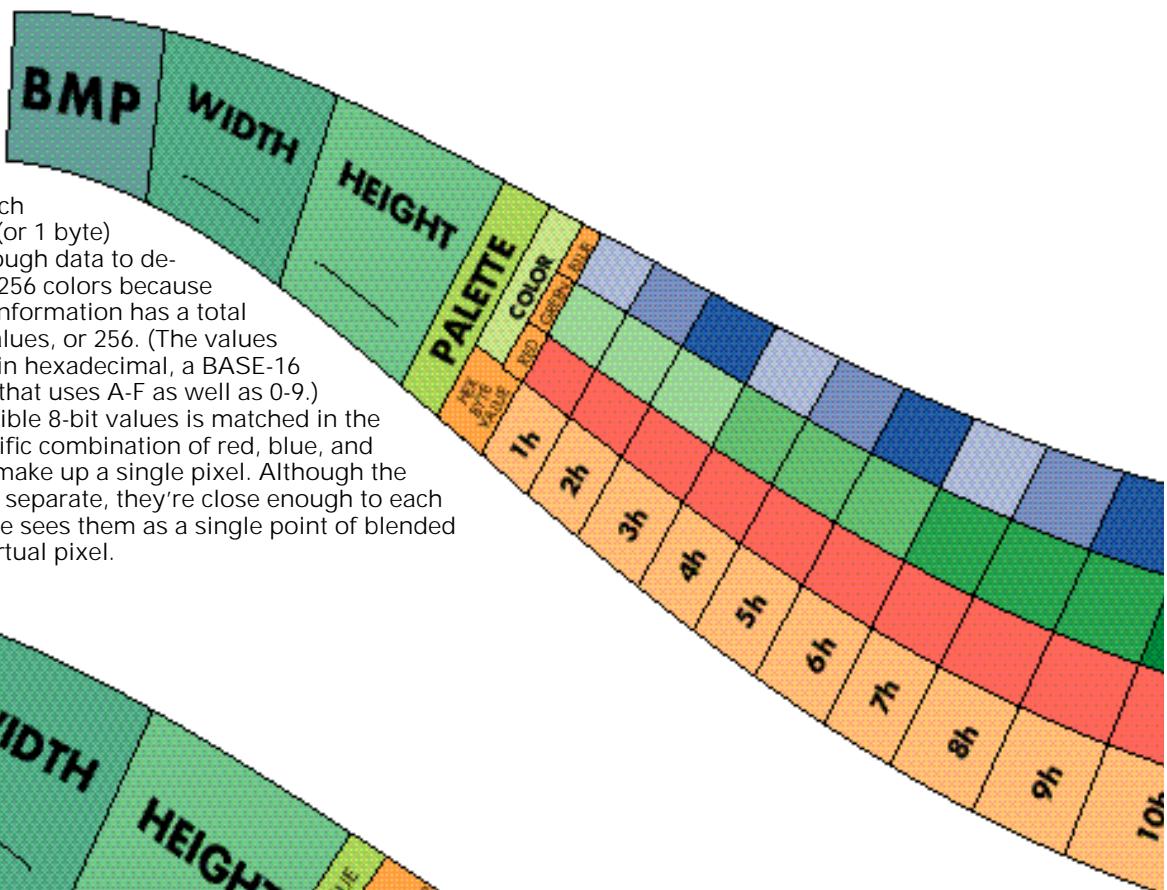
- 2** After determining the parameters of the graphic file, the program reads the bytes of data following the header that contain the image as a pattern of bits. The simplest bitmapped image is one that has only black and white pixels. For images of this type, the graphics program needs only two pieces of information: the location of a pixel and whether to turn the pixel on or off. The locations of the pixels are determined by the image's width and height as defined in the header. In this crude image of a man in a hat, the line pixels wrap every 11 bits to the next row of pixels.



The last 7 bits are discarded because they fall outside the matrix of pixels set up in the header.

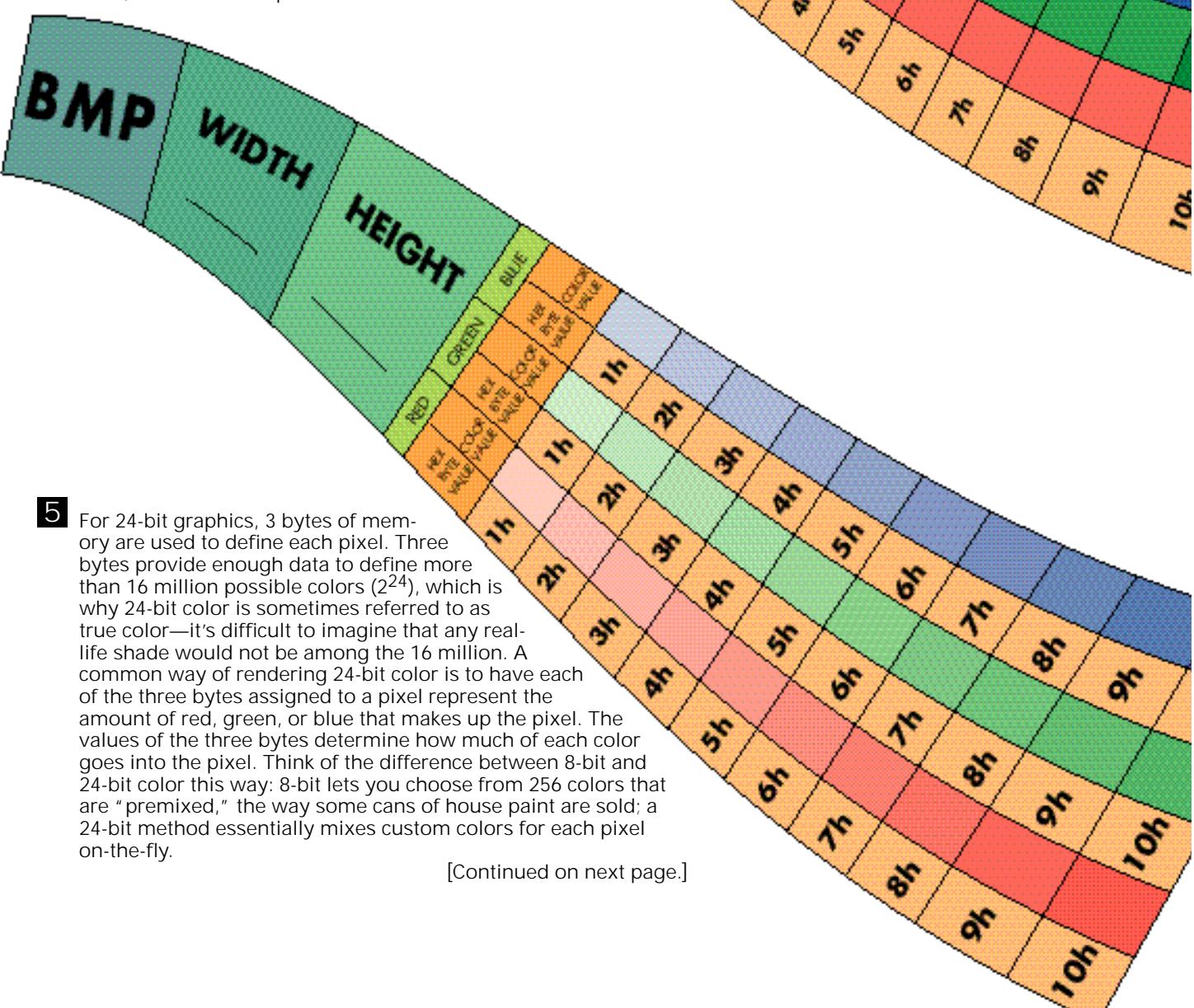
- 3** In the memory set aside for video display, the bytes that make up the black-and-white image consist of some bits set to 1 and the rest to 0. A 1 means a pixel that corresponds to that bit should be turned on. A 0 bit indicates a pixel should be turned off. The man in the hat consists of 121 pixels, which in a black-and-white image can be stored in 16 bytes.

- 4 A color bitmap requires more than 1 bit of information for each pixel. Eight bits (or 1 byte) per pixel are enough data to define a palette of 256 colors because 8 bits of binary information has a total of  $2^8$  possible values, or 256. (The values shown here are in hexadecimal, a BASE-16 number system that uses A-F as well as 0-9.) Each of the possible 8-bit values is matched in the palette to a specific combination of red, blue, and green dots that make up a single pixel. Although the dots of color are separate, they're close enough to each other that the eye sees them as a single point of blended color, called a virtual pixel.



- 5 For 24-bit graphics, 3 bytes of memory are used to define each pixel. Three bytes provide enough data to define more than 16 million possible colors ( $2^{24}$ ), which is why 24-bit color is sometimes referred to as true color—it's difficult to imagine that any real-life shade would not be among the 16 million. A common way of rendering 24-bit color is to have each of the three bytes assigned to a pixel represent the amount of red, green, or blue that makes up the pixel. The values of the three bytes determine how much of each color goes into the pixel. Think of the difference between 8-bit and 24-bit color this way: 8-bit lets you choose from 256 colors that are "premixed," the way some cans of house paint are sold; a 24-bit method essentially mixes custom colors for each pixel on-the-fly.

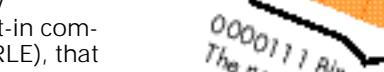
[Continued on next page.]



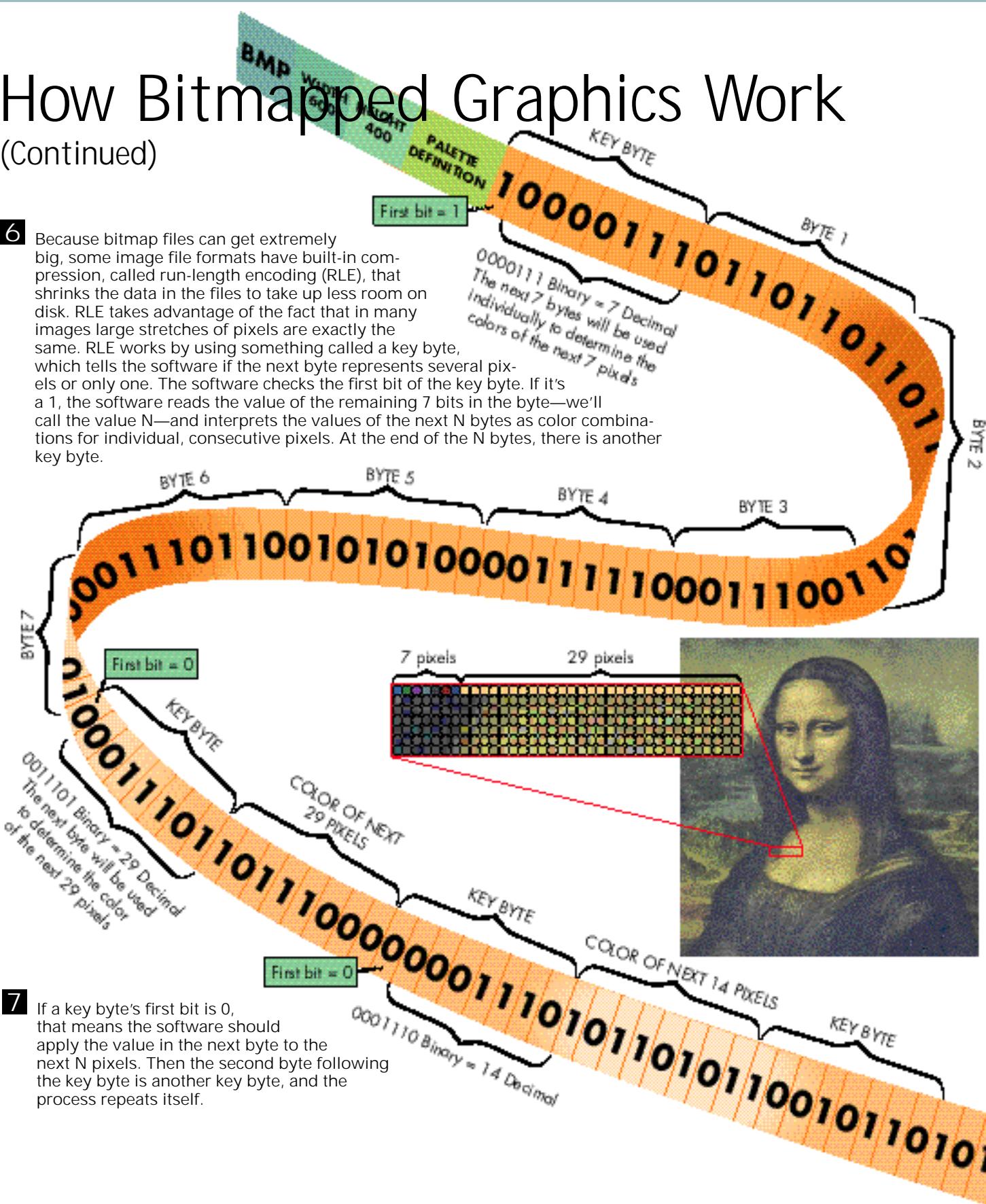
# How Bitmapped Graphics Work

(Continued)

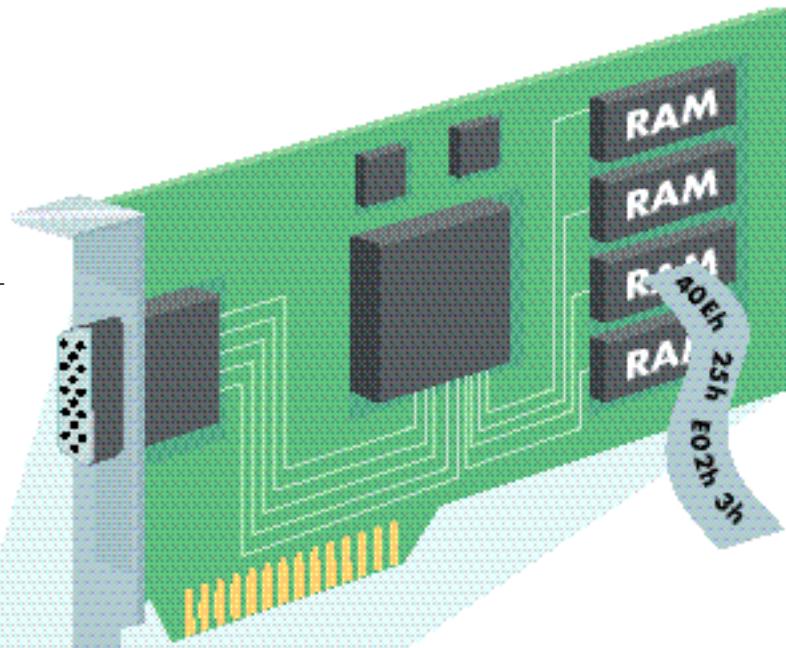
- 6 Because bitmap files can get extremely big, some image file formats have built-in compression, called run-length encoding (RLE), that shrinks the data in the files to take up less room on disk. RLE takes advantage of the fact that in many images large stretches of pixels are exactly the same. RLE works by using something called a key byte, which tells the software if the next byte represents several pixels or only one. The software checks the first bit of the key byte. If it's a 1, the software reads the value of the remaining 7 bits in the byte—we'll call the value N—and interprets the values of the next N bytes as color combinations for individual, consecutive pixels. At the end of the N bytes, there is another key byte.



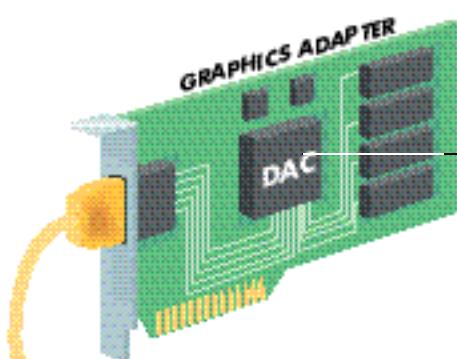
The diagram illustrates the RLE process. It shows a sequence of binary digits: 0000111. A callout box explains that this represents 7 decimal digits. Below, it says "The next 7 bytes will be used individually to determine the colors of the next 7 pixels".



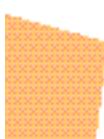
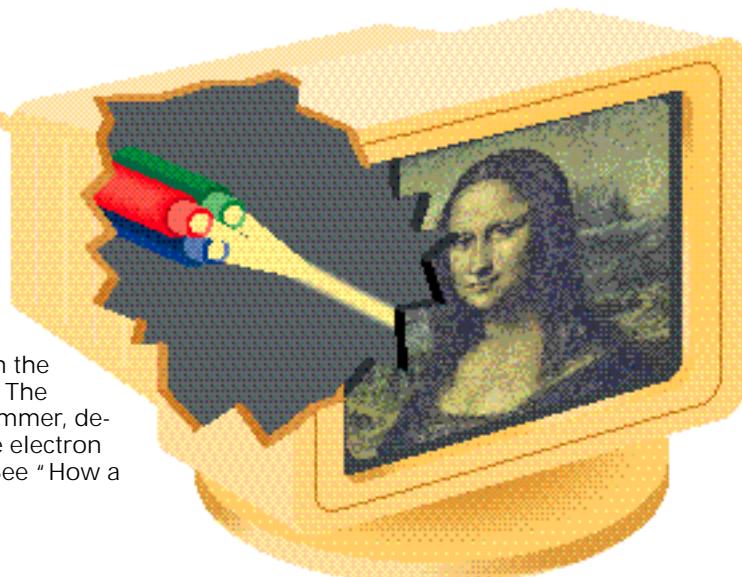
- 8 Regardless of the method used, after interpreting the bitmap file, the graphics software puts the values for the pixels into a section of the video adapter's memory called the frame buffer, which holds a value for each pixel on the screen. The adapter, in turn, uses that data in memory to adjust the electronic signals that set the intensity of each of the dots of red, green, and blue that make up the pixel.



- 9 A digital-to-analog converter (DAC) converts the digital value to corresponding analog values that determine how much electrical current is sent to the monitor.

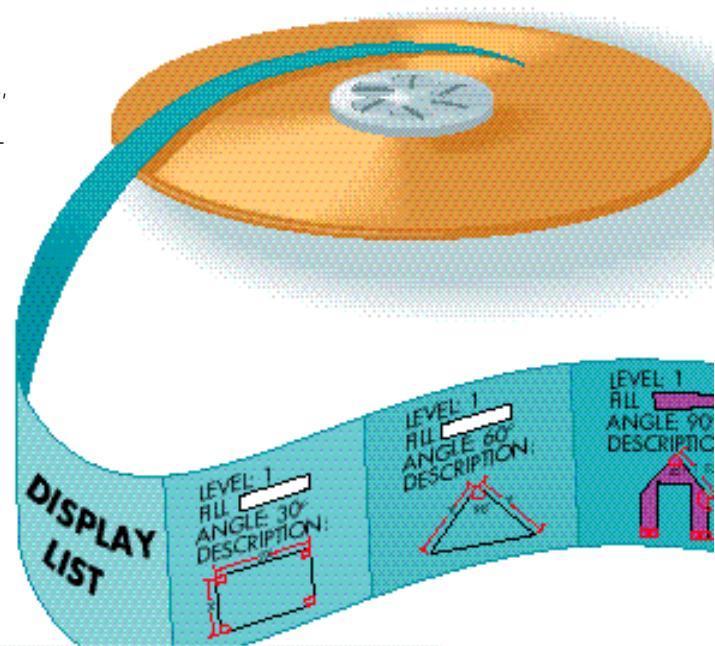


- 10 Three different electrical currents are sent to the monitor. The currents generate three electron streams of varying intensities to different phosphors—one phosphor for each of the three colors—painted on the inside of the monitor's screen. The phosphors glow brighter or dimmer, depending on the intensity of the electron stream that energizes them. (See "How a SuperVGA Display Works.")

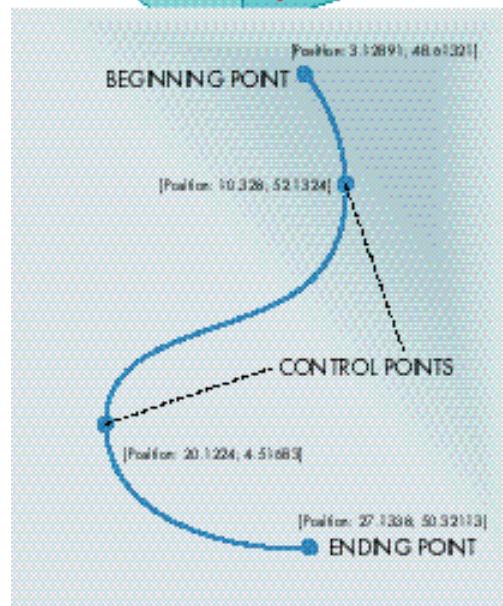


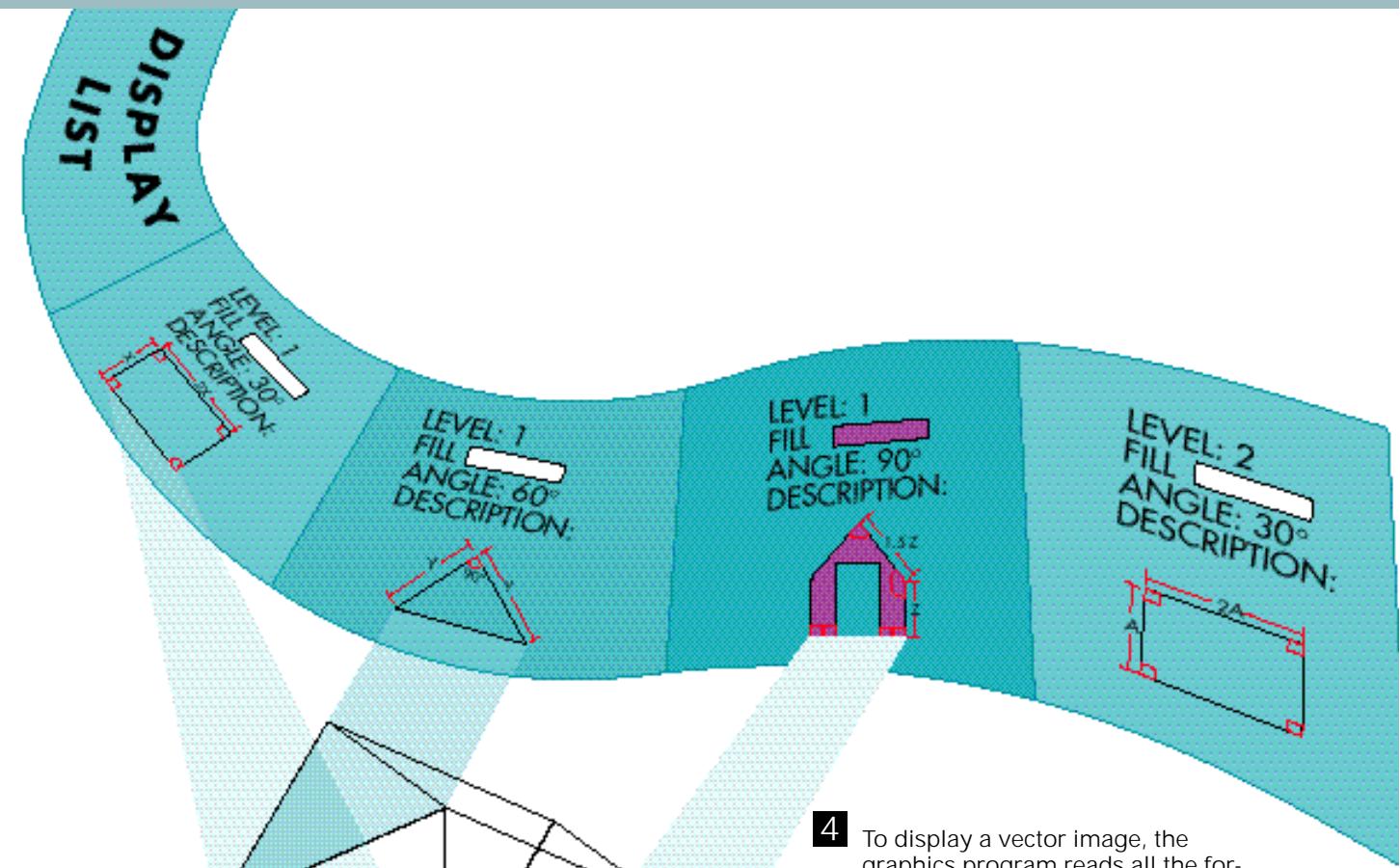
# How Vector Graphics Work

- 1** A vector-based graphic image is stored in a file as a display list that describes in mathematical terms every shape, or object, in the image along with its location and properties, such as line width and fill (the color pattern that fills a shape). The display list also specifies the hierarchy of objects—which ones should be drawn first and which are on top of others.



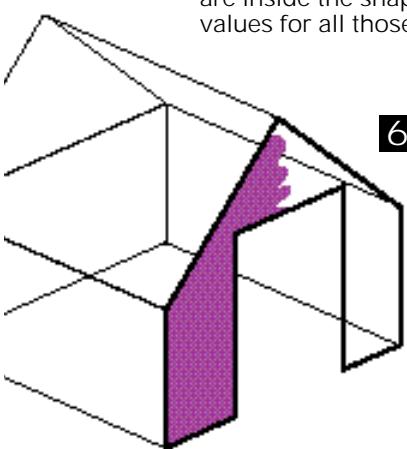
- 2** To draw an object, the program needs to know the locations of only a few points. The formula for a Bézier curve, for example, needs only four points: the beginning point, the ending point, and two control points that determine how far the curve is “pulled” away from a straight line.
- 3** The points are each defined by two numbers—one for the point’s vertical position and the other for the horizontal position. Each of those numbers is stored to a much higher degree of precision than could be specified by the pixel locations alone. This precision allows the software to draw the curve accurately, no matter how much its size is increased or decreased, or the curve is moved or otherwise manipulated.



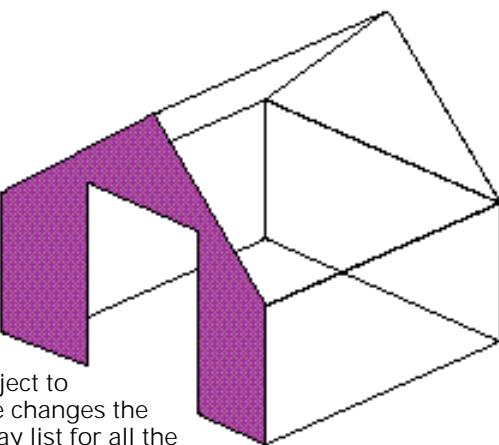


- 4 To display a vector image, the graphics program reads all the formulas and their accompanying data from the display list and uses them to compute a bitmapped image. This process is called viewing transform.

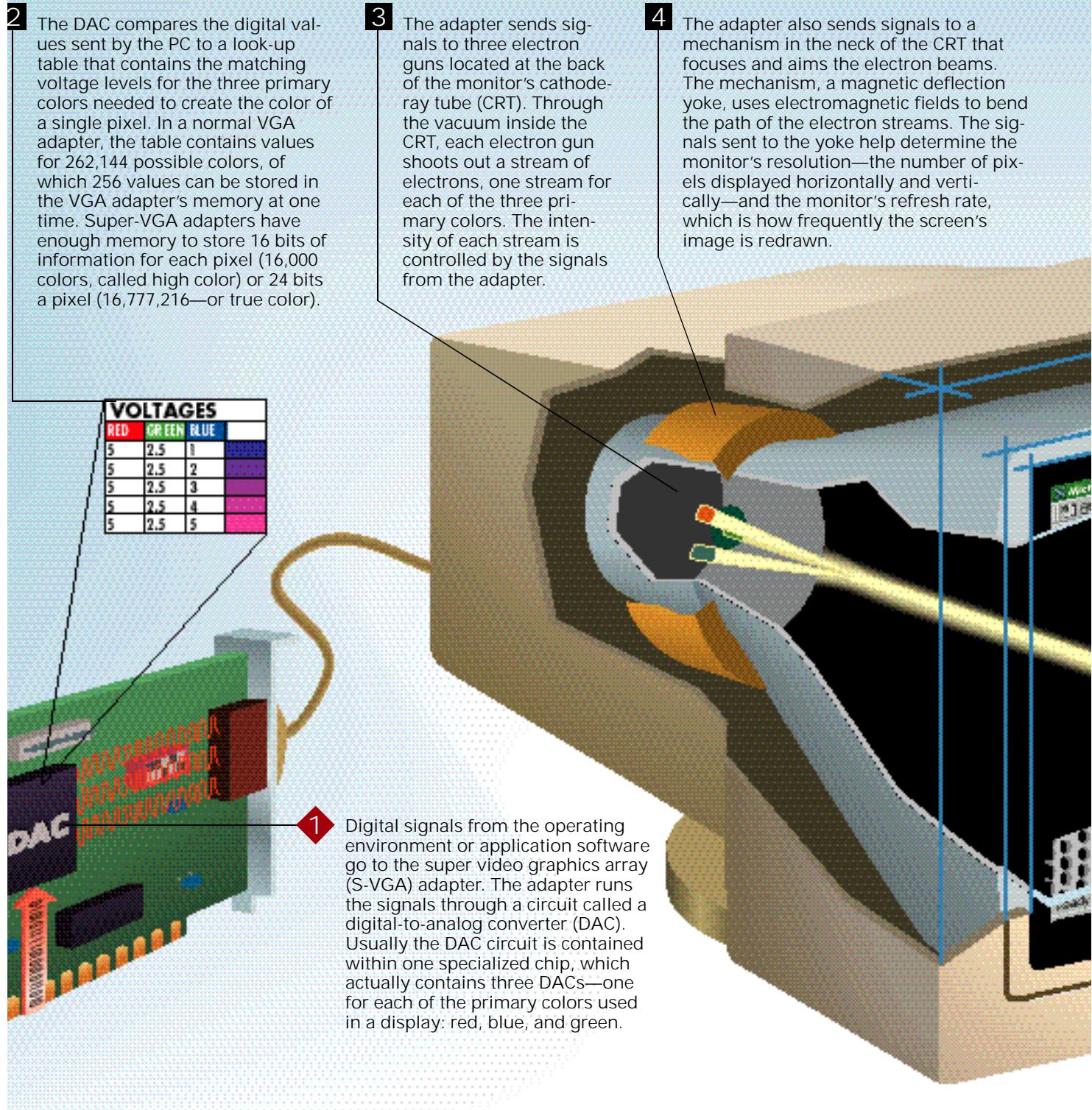
5 To display a fill, the program computes a mathematical formula that determines the locations of all the pixels that make up the edge of the shape. With that information, the program can determine which pixels are inside the shape and change the color values for all those pixels to match the fill.

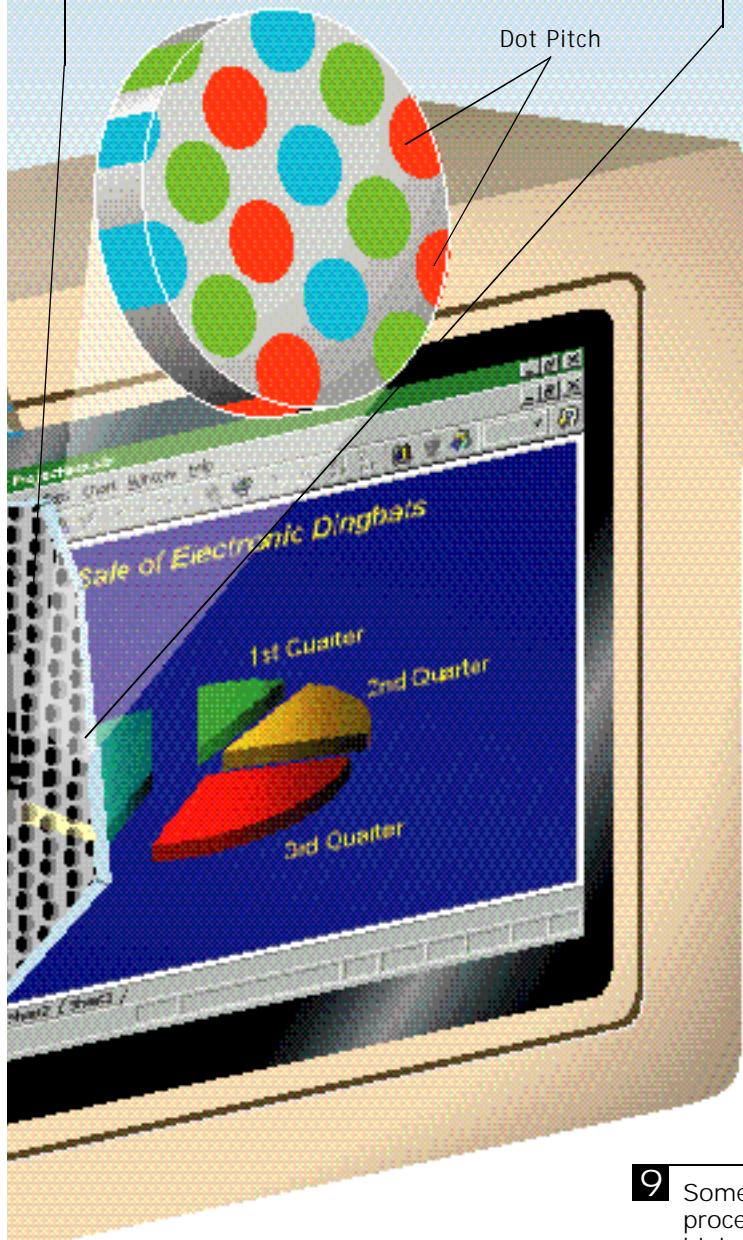


- 6 Anytime you change or move a shape, change a shape's attributes (such as its color), or add a new object to the image, the software changes the data stored in the display list for all the affected objects. Data for any object not changed is left unmodified. Viewing transform then recomputes the display bitmap to update the screen.



# How a Super-VGA Display Works

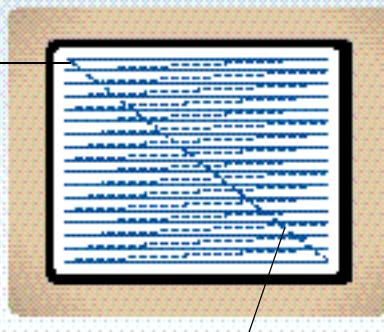




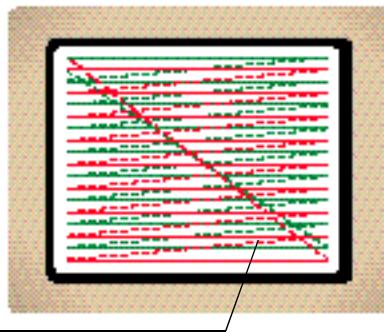
**5** The beams pass through holes in a metal plate called a shadow mask. The purpose of the mask is to keep the electron beams precisely aligned with their targets on the inside of the CRT's screen. The CRT's dot pitch is the measurement of how close the holes are to each other; the closer the holes, the smaller the dot pitch. This, in turn, creates a sharper image. The holes in most shadow masks are arranged in triangles, with the important exception of those of the Sony Trinitron CRT used by many monitor manufacturers. The Trinitron's holes are arranged as parallel slots.

**6** The electrons strike the phosphors coating the inside of the screen. Phosphors are materials that glow when they are struck by electrons. Three different phosphor materials are used—one each for red, blue, and green. The stronger the electron beam that hits a phosphor, the more light the phosphor emits. If each red, green, and blue dot in an arrangement is struck by equally intense electron beams, the result is a dot of white light. To create different colors, the intensity of each of the three beams is varied. After a beam leaves a phosphor dot, the phosphor continues to glow briefly, a condition called persistence. For an image to remain stable, the phosphors must be reactivated by repeated scans of the electron beams before the persistence fades away.

**7** After the beams make one horizontal sweep across the screen, the electron streams are turned off as the electron guns refocus the path of the beams back to the left edge of the screen at a point just below the previous scan line. This process is called raster scanning.



**8** The magnetic deflection yoke continually changes the angles at which the electron beams are bent so that they sweep across the entire screen surface from the upper-left corner of the screen to the lower-right corner. A complete sweep of the screen is called a field. Upon completing a field, the beams return to the upper-left corner to begin a new field. The screen is normally redrawn, or refreshed, about 60 times a second.



**9** Some display adapters scan only every other line with each field, a process called interlacing. Interlacing allows the adapter to create higher resolutions—that is, to scan more lines—with less expensive components. But the fading of the phosphors between each pass can be noticeable, causing the screen to flicker.

# How a Color Liquid Crystal Display (LCD) Works

**2** A polarizing filter in front of the light panel lets through only the light waves that are vibrating more or less horizontally. The fact that the polarizing filter is not entirely precise allows the display to create different hues.

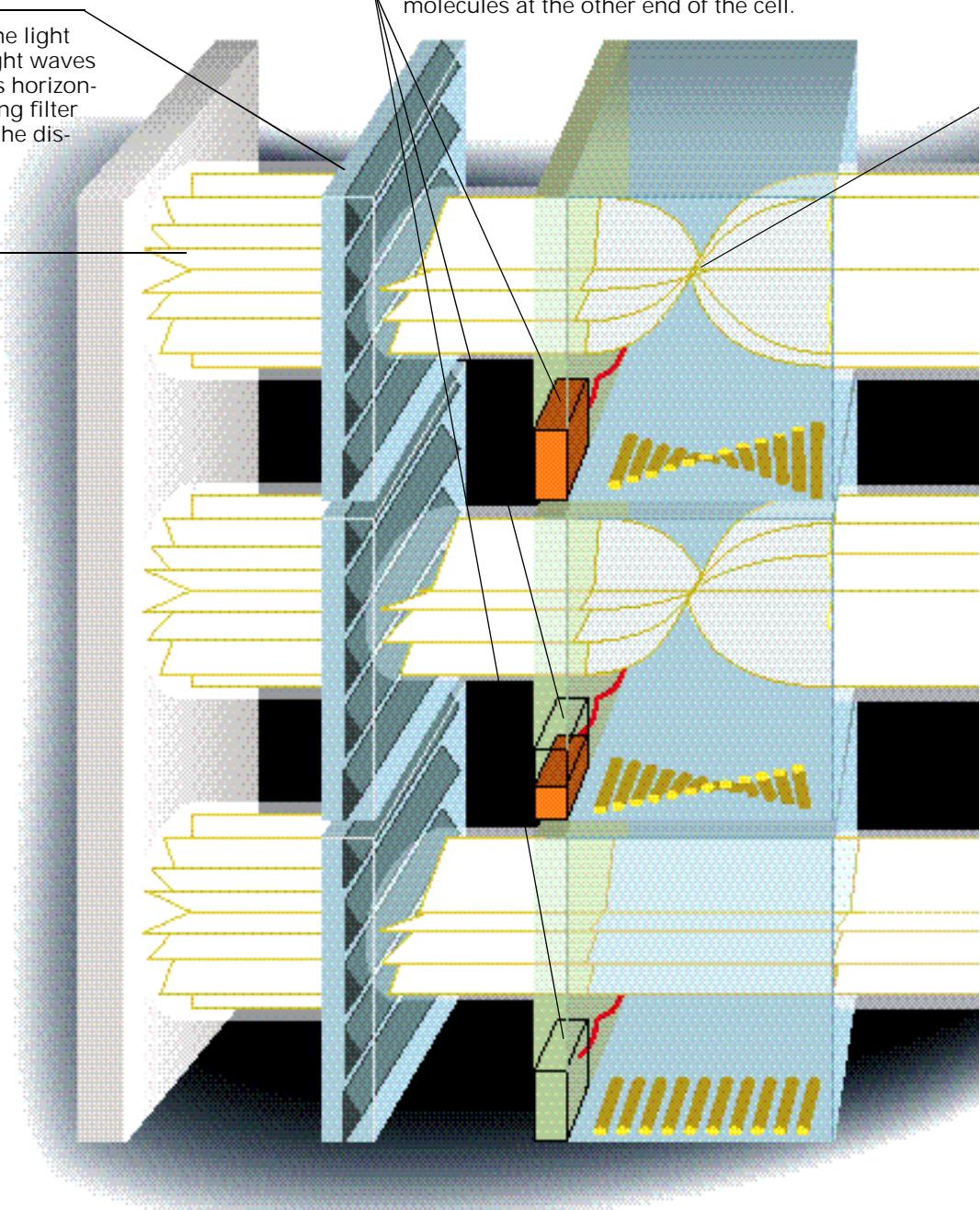
**1** Light—emanating from a fluorescent panel behind a portable computer's display panel—spreads out in waves that vibrate in all directions.

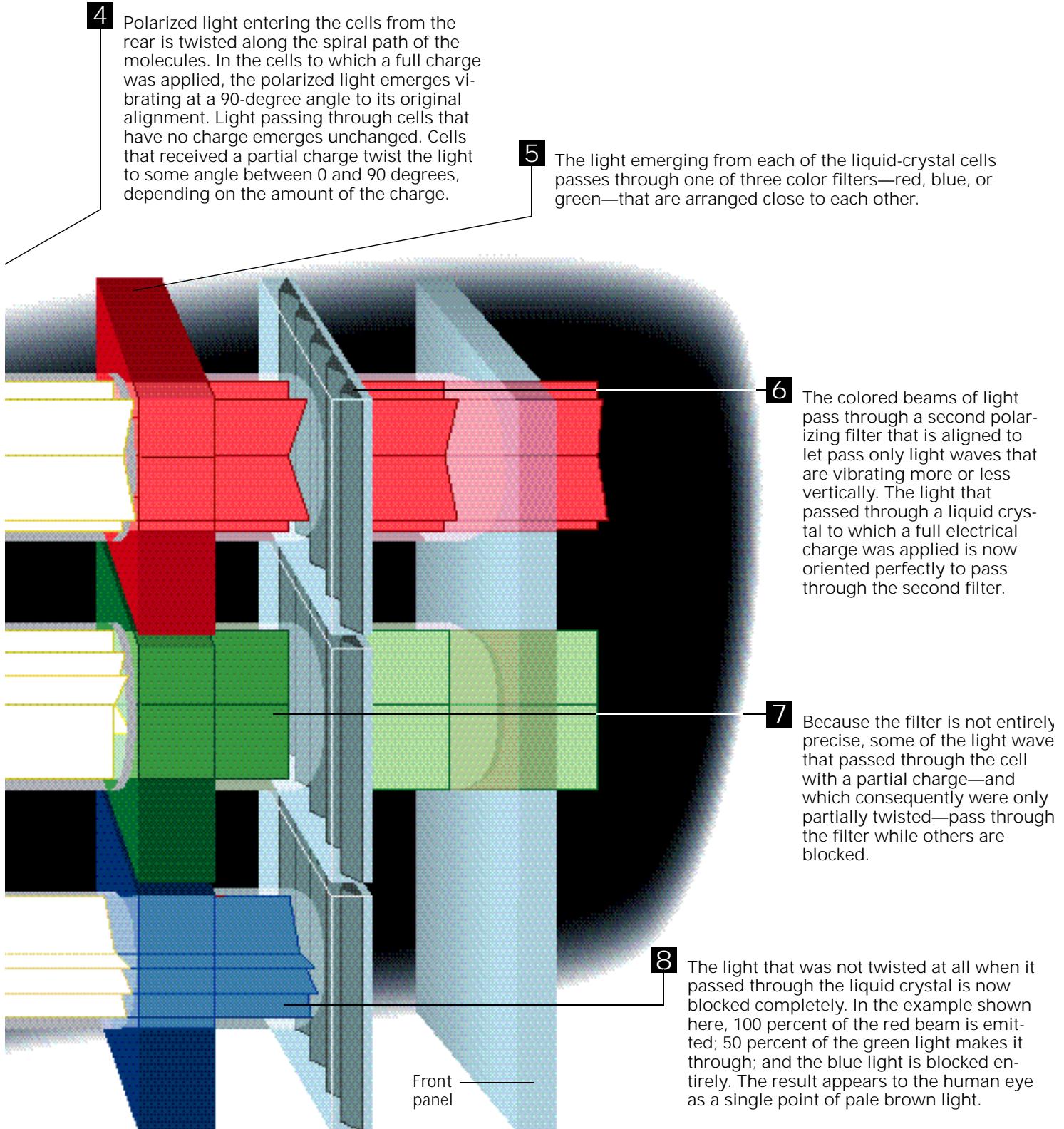
**3**

In a layer of liquid-crystal cells—one for each of the three colors that make up a pixel—the built-in graphics adapter of the portable PC applies a varying electrical charge to some of the cells and no charge at all to other cells. In cells to which current is applied, the long, rod-shaped molecules that make up the liquid-crystal material react to the charge by forming a spiral. The greater the charge, the more the molecules twist. With the strongest charge, the molecules at one end of the cell wind up at an angle 90 degrees from the orientation of the molecules at the other end of the cell.

## A Twist on LCD

The model shown here is only one way in which liquid crystals and polarizers can manipulate light. Some LCD panels use two polarizers with the same alignment so that a charge applied to a liquid crystal cell results in light that's blocked because it's twisted. Also, two methods are used to apply charges to liquid crystal cells. Passive matrix displays use only a relatively few electrodes arranged along the edges of the liquid crystal layer and rely on timing to make sure the correct cells are charged. The charges in passive matrix cells fade quickly, causing the colors to look faded. Active matrix displays, such as the one shown here, have individual transistors for each of the cells. The individual transistors provide a more precise and stronger charge, creating colors that are more vivid.





# How a Gas Plasma Display Works

1

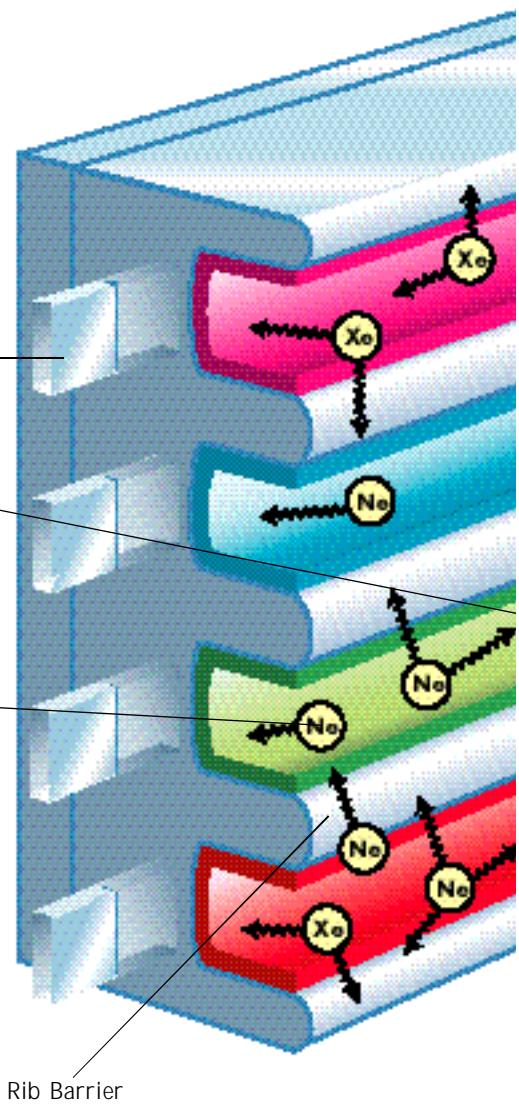
An electrical current is applied to two electrodes. One, an address electrode in the rear of the gas plasma display, identifies which of the display's cells are to be affected by another electrical current passing through transparent electrodes in the front of the display.

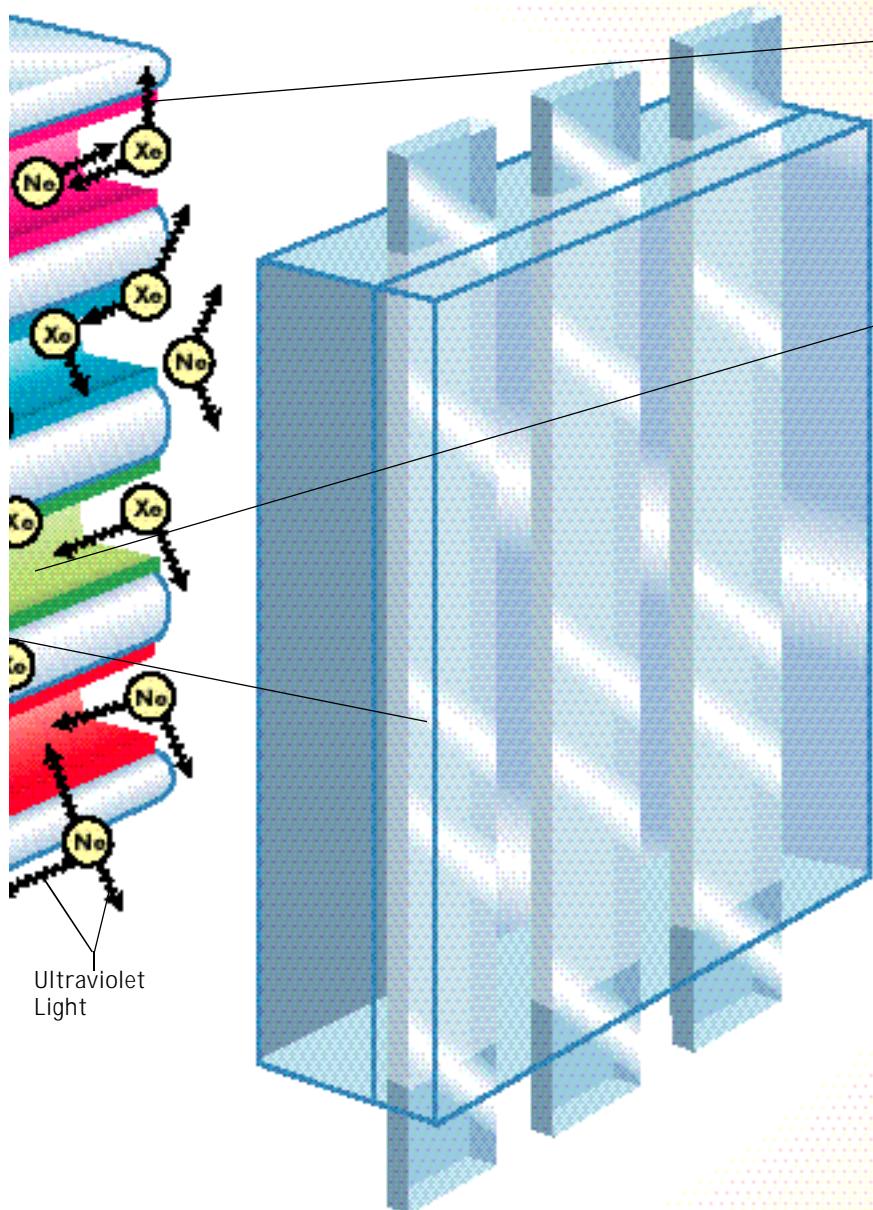
2

The current flows between the electrodes through a cell filled with a mixture of neon and xenon gases. Ridges called rib barriers separate the long cells that cross the width of the display.

## The Perfect Display?

Gas plasma promises to be the all-around display of choice in the future. Like an LCD screen, a plasma display is less than two inches thick. You can hang it on the wall like a painting. But unlike LCD, plasma is brighter and can be seen from a wide angle—160 degrees—because light does not pass through polarizing filters. Polarization by its nature absorbs most of the light that strikes the filters, and the light that does pass through travels mostly at a right angle to the surface of the screen, requiring you to view the screen straight-on for best brightness and contrast. Gas plasma displays can easily be larger than 40 inches diagonally, a size difficult to achieve in LCD screens. The giant size makes gas plasma perfect for presentations and for television displays. Finally, its simplified circuitry and less-expensive materials lend itself to low-cost, high-volume production.

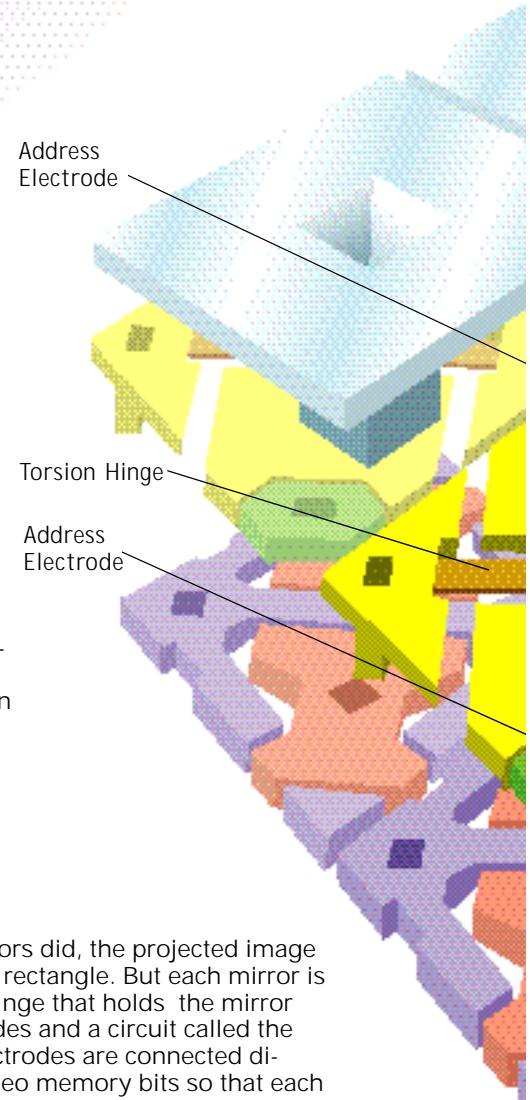
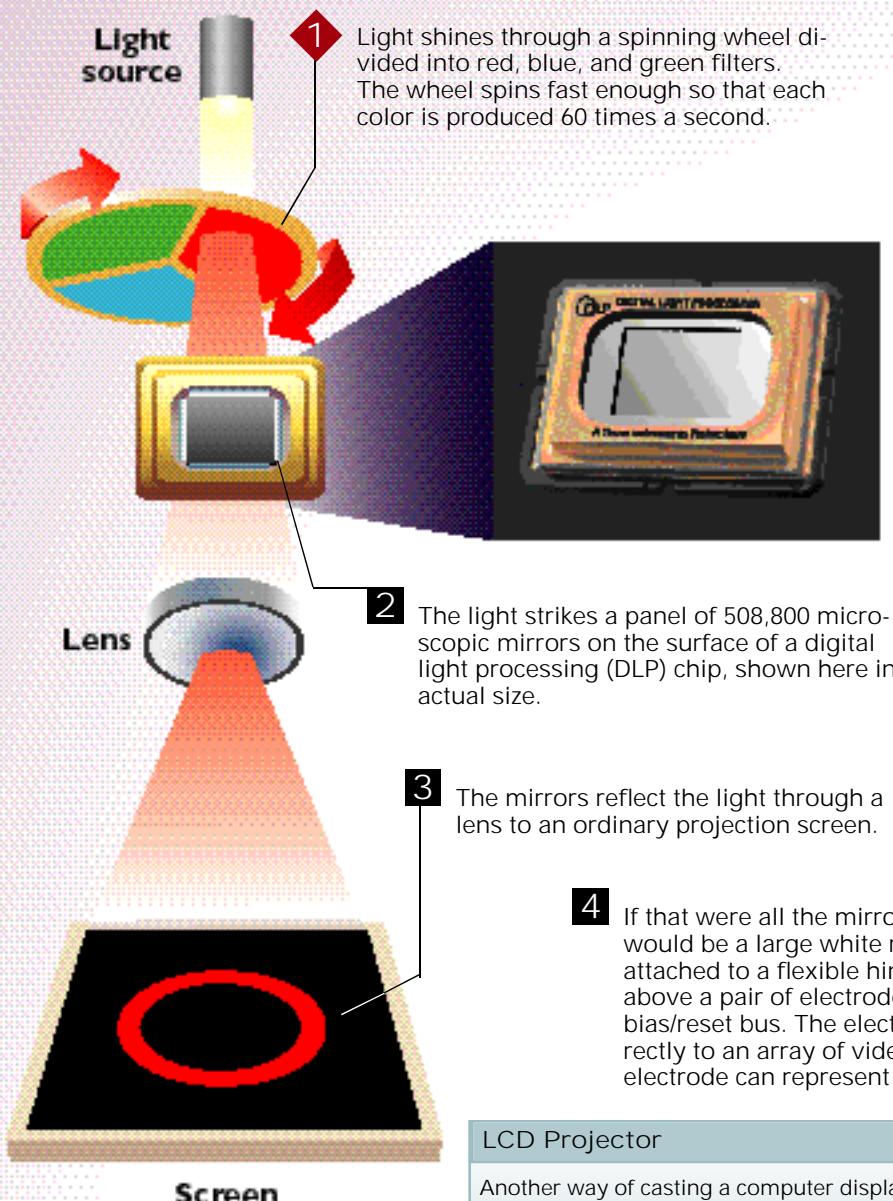




**3** The energy from the electrical current excites the atoms in the gas mixture, causing the gas molecules to throw off ultraviolet light waves. The higher the current level passing through the gas, the more intense the ultraviolet light.

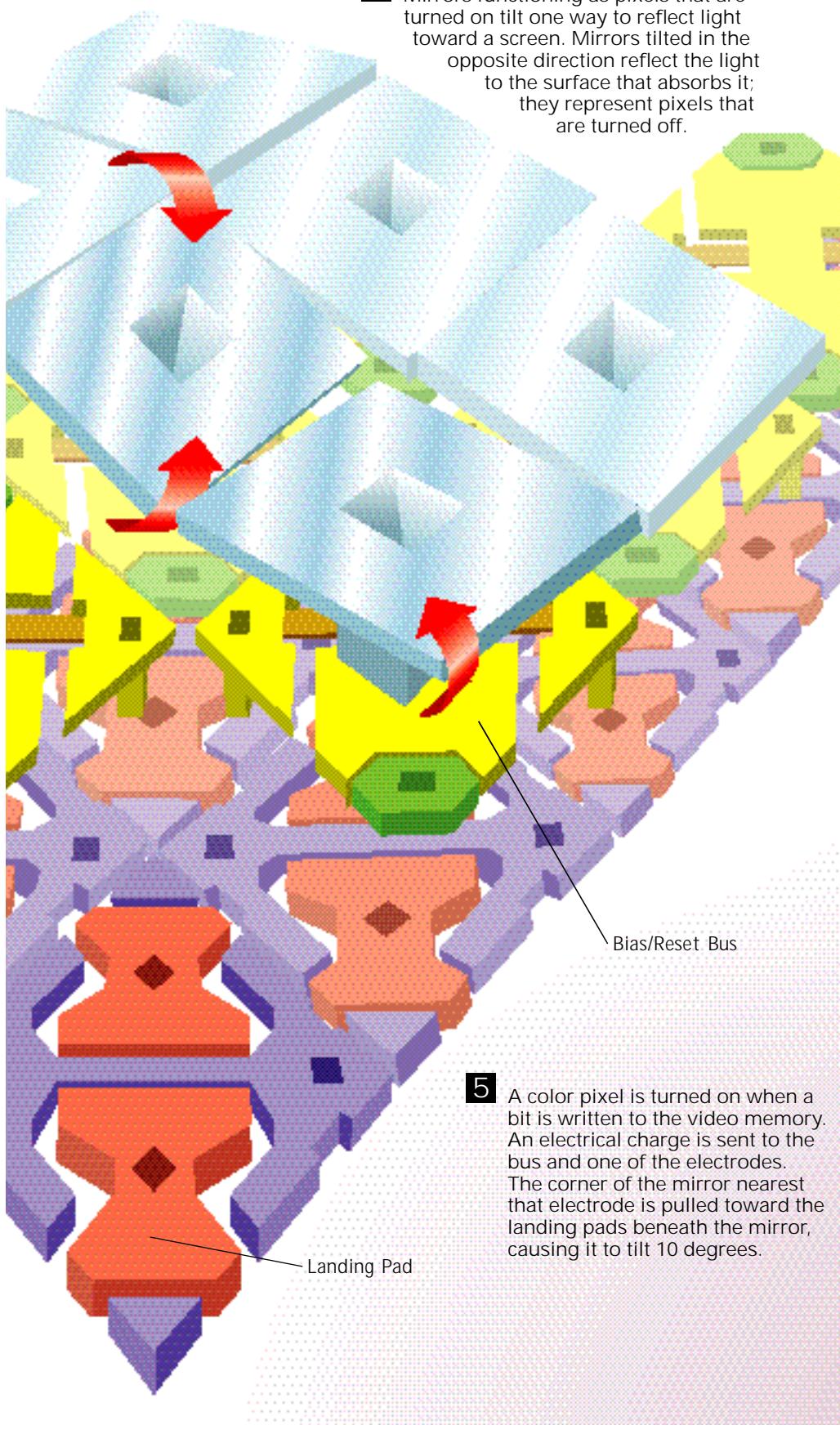
**4** The ultraviolet waves strike phosphors that have been spray-painted inside each cell. The ultraviolet lights cause the phosphors to glow—similar to the process in an everyday cathode-ray tube monitor. Different phosphor materials that glow either red, blue, or green coat the inside of three adjacent cells to create a single logical pixel. By varying the amount of current going to each cell, the display changes the mix of the three primary colors to create different hues.

# How Digital Light Processing Works

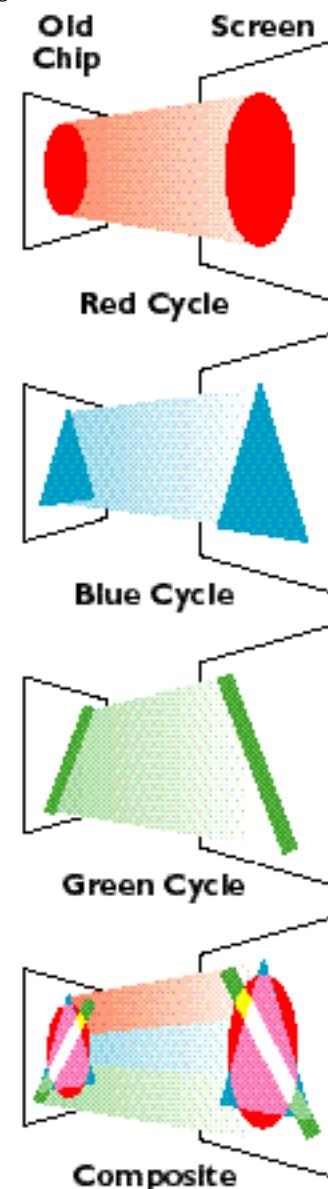


## LCD Projector

Another way of casting a computer display image is with a liquid crystal display projector. Colors and images are created the same way as for a laptop computer's LCD display. (See "How a Color Liquid Crystal Display (LCD) Works," page 152.) The essential differences are that the projector's light source is much stronger than a laptop's, and the image is focused through a lens. But the nature of LCD means that much of the light is absorbed so that an LCD image is dimmer than a DSP projection. In addition, the gaps between the mirrors in a DSP display are smaller than those between the cells that make up the LCD, making the DSP image sharper.

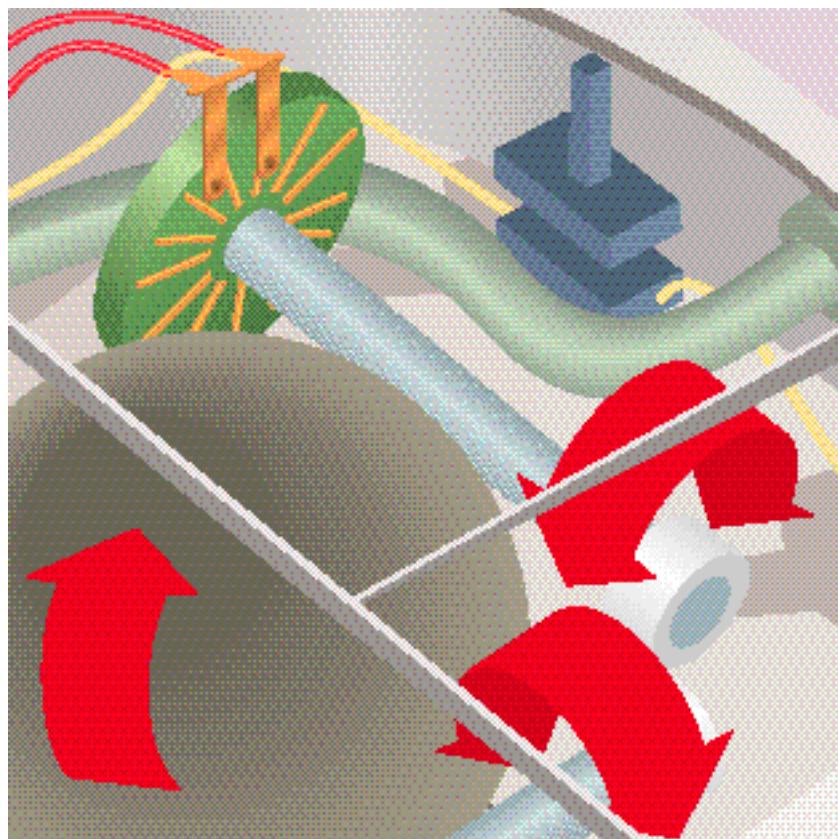


7 Hues of colors are created by how often the source light is turned on and off while the three filters pass in front of it. The color wheel and the RGB data are synced so that the time a mirror reflects any one of the three primary colors is in proportion to the amount of that color in the mixture that is perceived by the human eye. Each mirror can tilt on or off in less than 20 milliseconds, which means that each mirror can generate 16.7 million colors.



CHAPTER  
**22**

# How Pointing Devices Work



**THERE** is nothing natural or intuitive about a keyboard. No child is born knowing how to type, and even when the skill is learned, there's little sense to it—no one can give a sensible explanation of why the alphanumeric keys are arranged the way they are.

For many, the keyboard is actually a barrier to learning how to use a computer. Even for the experienced typist, there's nothing instinctive in pressing F5 to print a file. Engineers—not one of them touch typists, I'll bet—at Xerox Corporation's Palo Alto Research Center (PARC) developed a concept first explored by Douglas C. Engelbert of the Stanford Research Center. The concept was a pointing device, something a computer user could move by hand, causing a corresponding movement onscreen. Because of its size and tail-like cable, the device was named for the mouse. Apple Computer made the mouse a standard feature of its Macintosh computers, and Windows has made a mouse standard equipment on PCs, as well.

The mouse is not the only pointing device that's been invented. The joystick used with games essentially accomplishes the same task, but doesn't feel quite right in all situations. Digitizing tablets are popular with artists and engineers who must translate precise movements of a pen into lines on the screen. Touch screens, using your finger or a special light pen to control the software, are too tiring to use for any length of time. The most successful of pointing innovations have been "eraserhead" pointing devices, so called because they look like a pencil's eraser stuck between the G and H key; the touch pad, which is a digitizing table without the precision; and trackballs. All three are popular on laptops, used where there's no space for a conventional mouse.

The mouse and its cousins can never replace the keyboard, but they can supplement the keys by doing tasks such as moving and pointing to onscreen objects, tasks for which the cursor keys are ill-suited. We're just starting to reach the point where we control our PCs simply by speaking to them.

Mice are still an integral part of our systems. The mechanical mouse is the most popular pointing device for graphic interfaces represented by Windows and the Macintosh OS. With the mouse, you control your PC by pointing to images instead of typing commands. Here's how the mouse translates the movements of your hand into the actions onscreen.

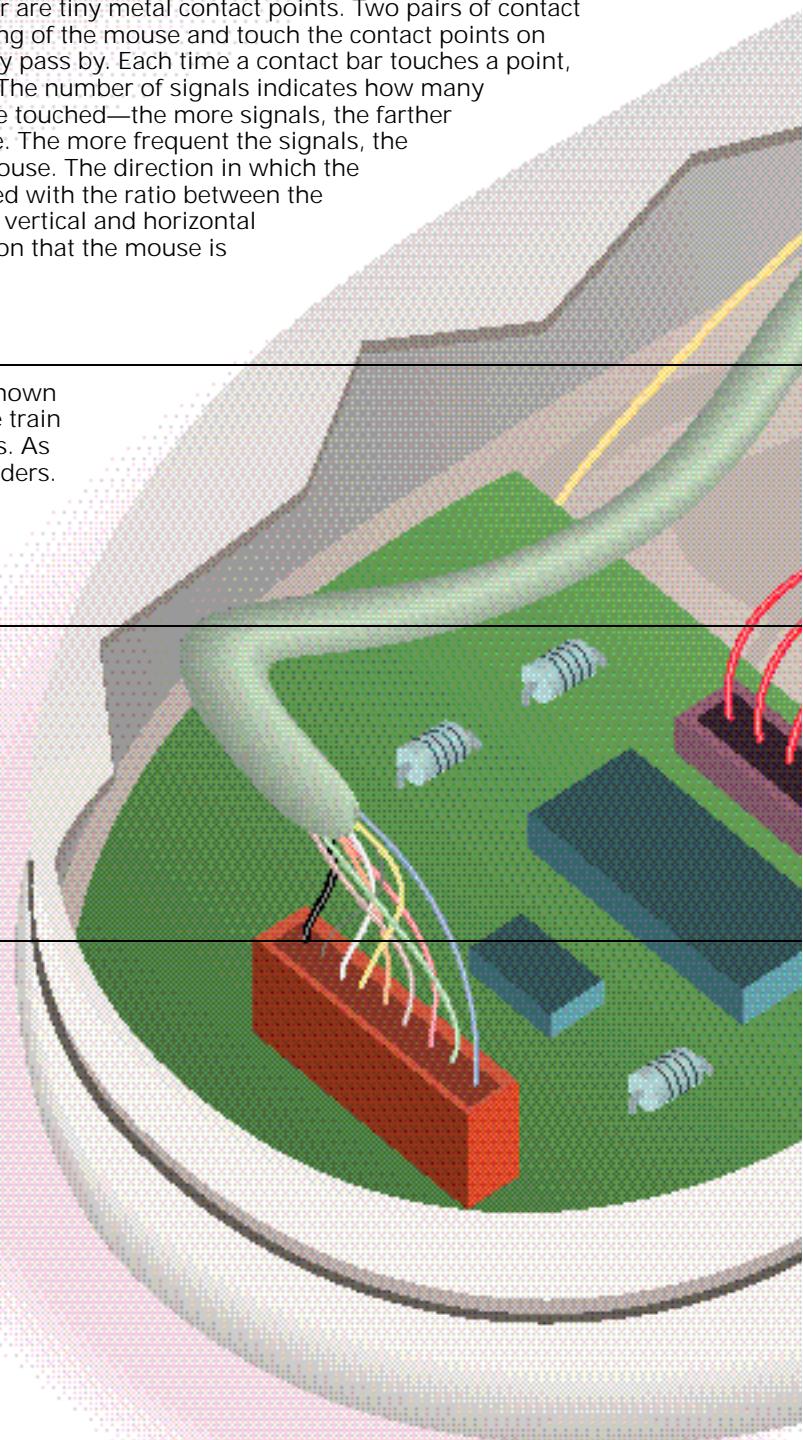
# The Mechanical Mouse

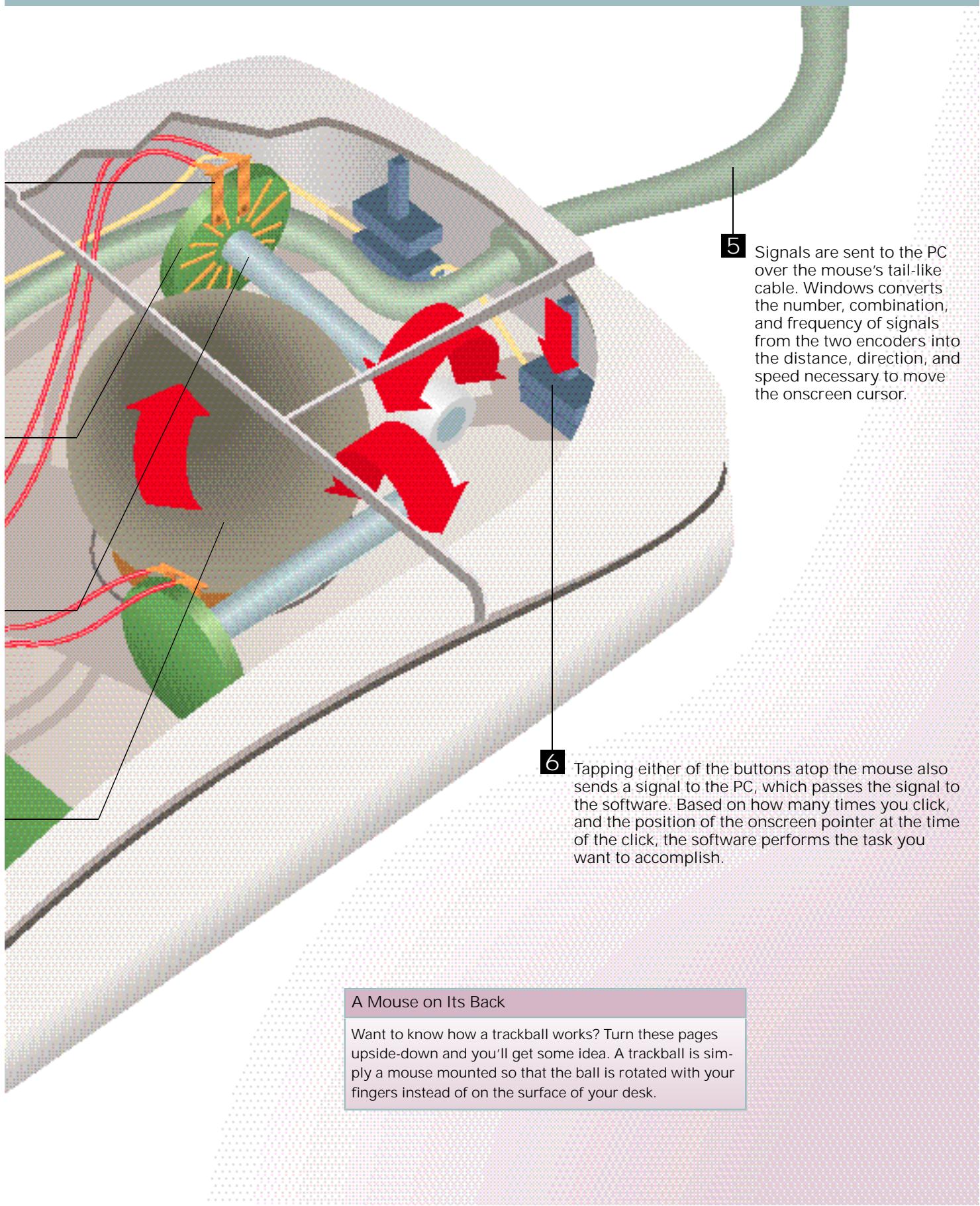
**4** On the rims of each encoder are tiny metal contact points. Two pairs of contact bars extend from the housing of the mouse and touch the contact points on each of the encoders as they pass by. Each time a contact bar touches a point, an electrical signal results. The number of signals indicates how many points the contact bars have touched—the more signals, the farther you have moved the mouse. The more frequent the signals, the faster you're moving the mouse. The direction in which the rollers are turning, combined with the ratio between the number of signals from the vertical and horizontal rollers, indicates the direction that the mouse is moving.

**3** Each roller is attached to a wheel, known as an encoder, much as a car's drive train is attached by its axles to the wheels. As the rollers turn, they rotate the encoders.

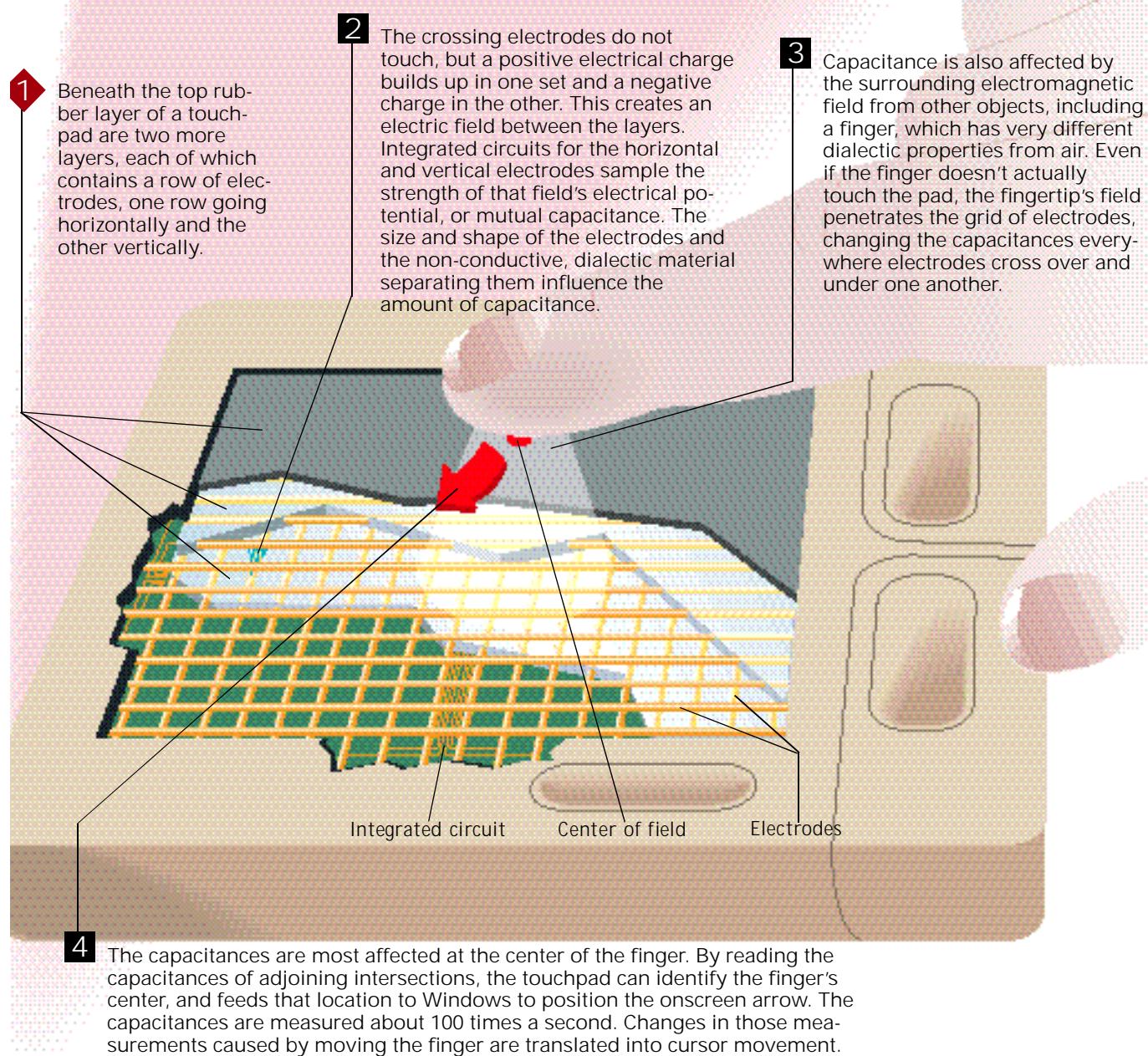
**2** As the ball rotates, it touches and turns two rollers mounted at a 90-degree angle to each other. One roller responds to back-and-forth movements of the mouse, which correspond to vertical movements onscreen. The other roller senses sideways movements, which correspond to side-to-side movements onscreen.

**1** As you move a mechanical mouse by dragging it across a flat surface, a ball—made of rubber or rubber over steel—protruding from the underside of the mouse turns in the direction of the movement.

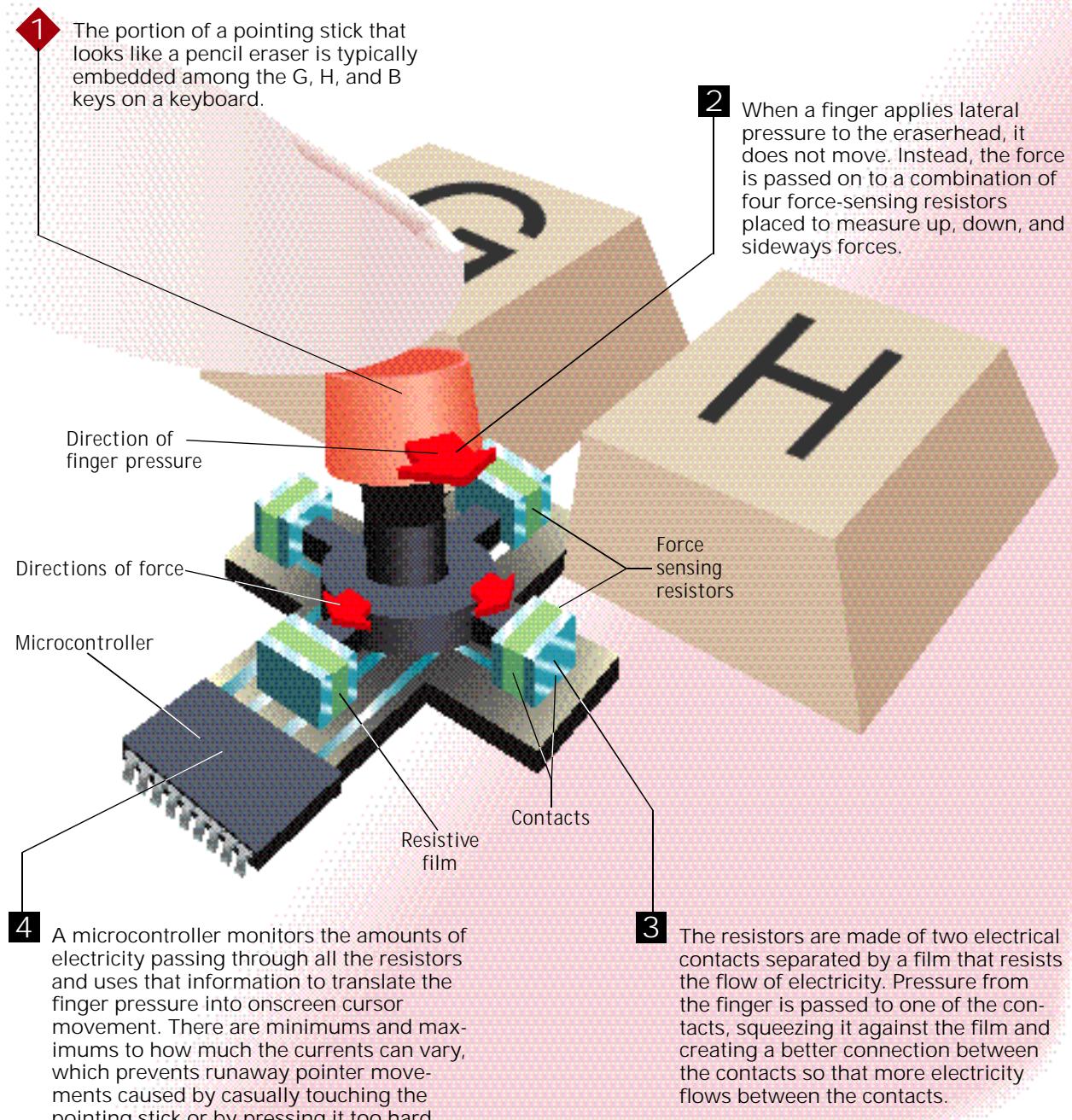




# How a Touchpad Works



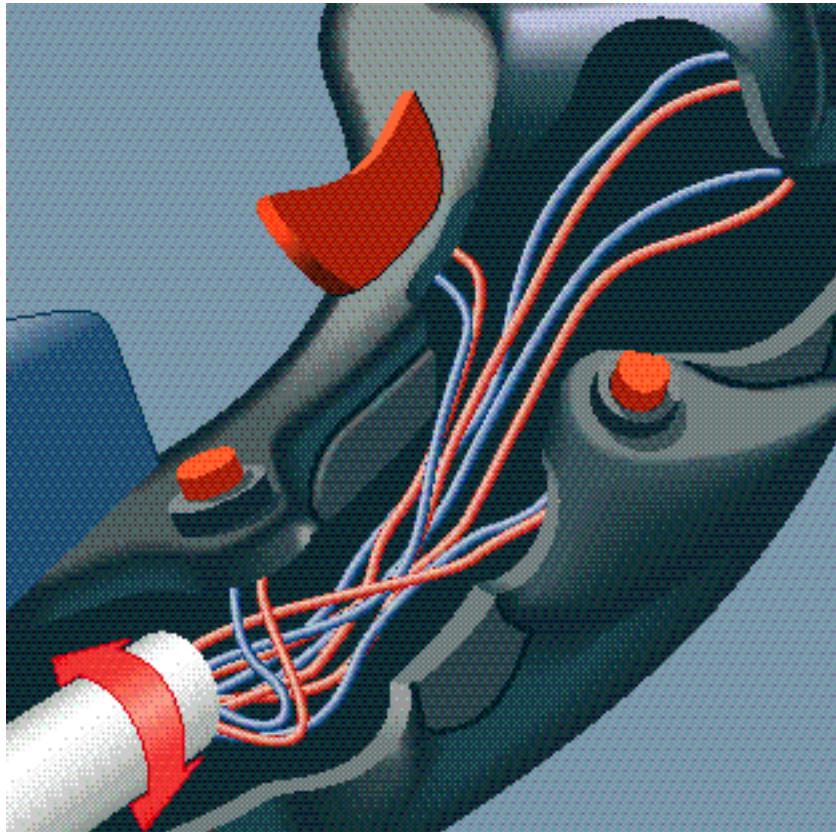
# How a Pointing Stick Works



CHAPTER

23

# How Game Controllers Work



**THERE** are times when a keyboard just won't do. If you're being pursued by a two-ton Belaguisian Swamp Thumper intent on making you its dinner, typing "My regrets. I can't attend." isn't going to help. Or if your F44X jet fighter is plunging toward the rocky ground, you can type "Up, up, up!" all you want and you're still going to turn into part of the landscape.

That's why there are joysticks and other game controllers. In all the talk about natural interface and ease of use in the Windows and Mac environments, no one seems to mention that language is not a natural interface. It's a learned ability. If you want a true natural interface, there's nothing better than a joystick. Want to go forward? Push the stick forward. Want to turn to the right? Push the stick to the right.

There have been tremendous improvements in game controllers in the last decade, while changes to keyboards have been few and of dubious worth. The game controller has evolved from a small box with a tiny rod sticking out of it to controls that look as if they've been ripped from the cockpit of a real fighter jet. At the same time, specialized game controllers have evolved for action software for which the joystick just isn't right. For racing games, we now have steering wheels and gas and brake pedals. A clever device called the Cyberpuck is particularly well-suited for first-person shooter games because it incorporates more movement directions than most joysticks can, all in a package that fits in your palm. PCs have stolen game pads from the video game console, despite the fact that it's been proven they are totally incomprehensible to anyone over the age of 16.

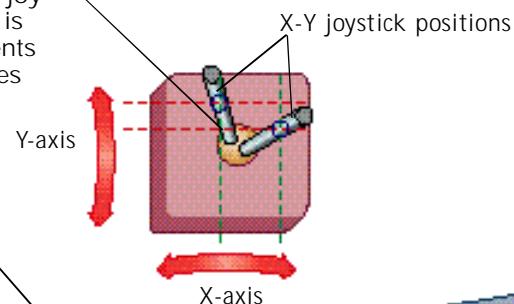
In this chapter, we're sticking to joysticks. All the other game controllers, despite how different they may look, work essentially as a joystick does: Pushing some button or lever creates an electrical signal that is transformed into onscreen movement.

We do take a look at a new, breakthrough joystick, the force-feedback joystick. Until the debut of feedback from several manufacturers in 1997, game controllers were based on eye-hand coordination. A feedback joystick adds hand-hand coordination. Rather than simply judging the effect of a joystick movement by watching the screen, you can feel resistance, wind, brick walls, push-back forces, and even the slipperiness of ice with your hand.

Now, if only they could come up with a keyboard that pushes back whenever I misspell a word.

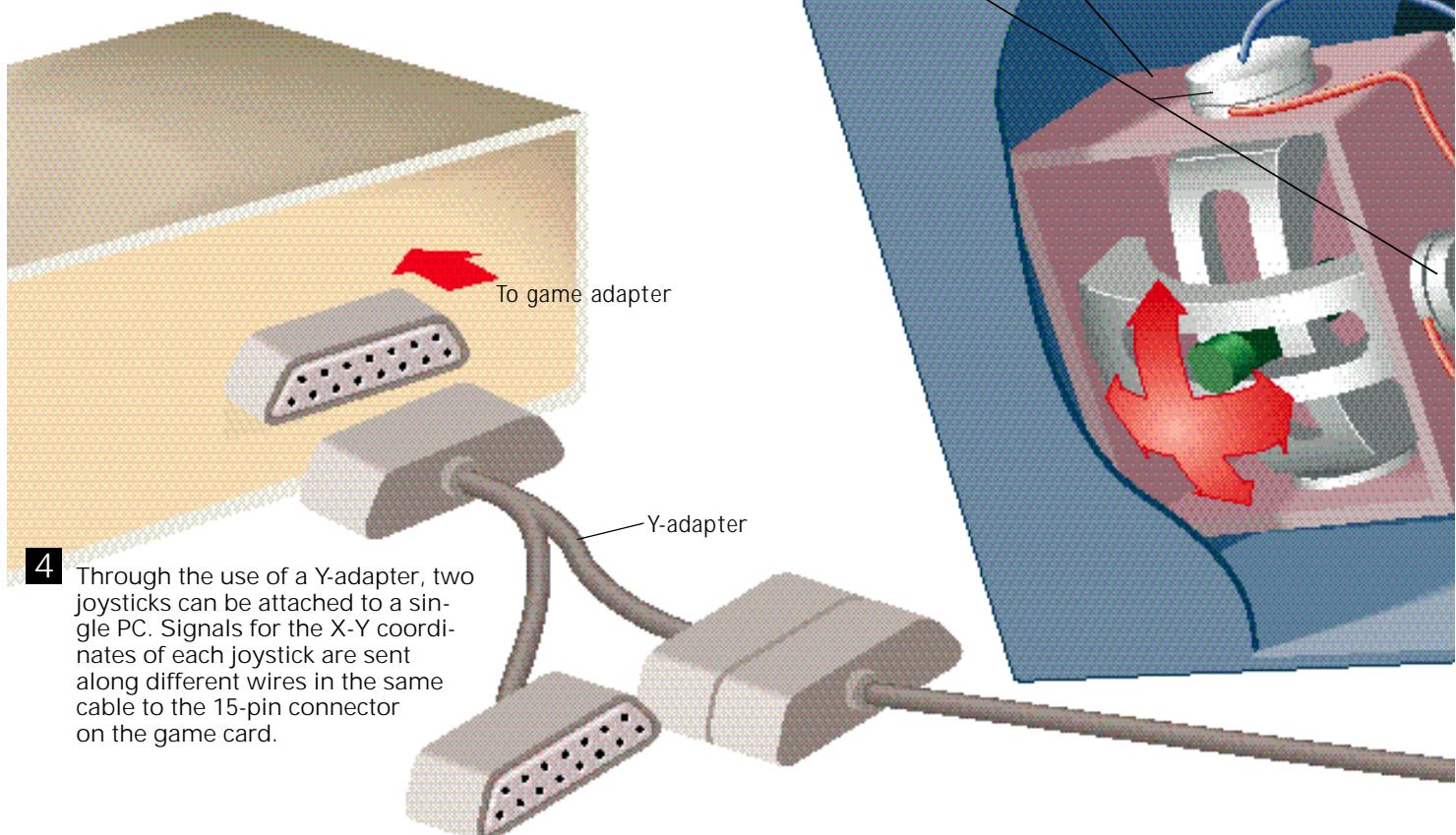
# How a Joystick Works

- 1** All joysticks are designed to tell a computer how the handle of the joystick is positioned at any time. To do this, all the joystick has to do is provide the X-Y axis coordinates of the handle. The X-axis represents the side-to-side position. The path of the Y-axis is shifted 90 degrees from that of the X-axis and represents the forward-back position.



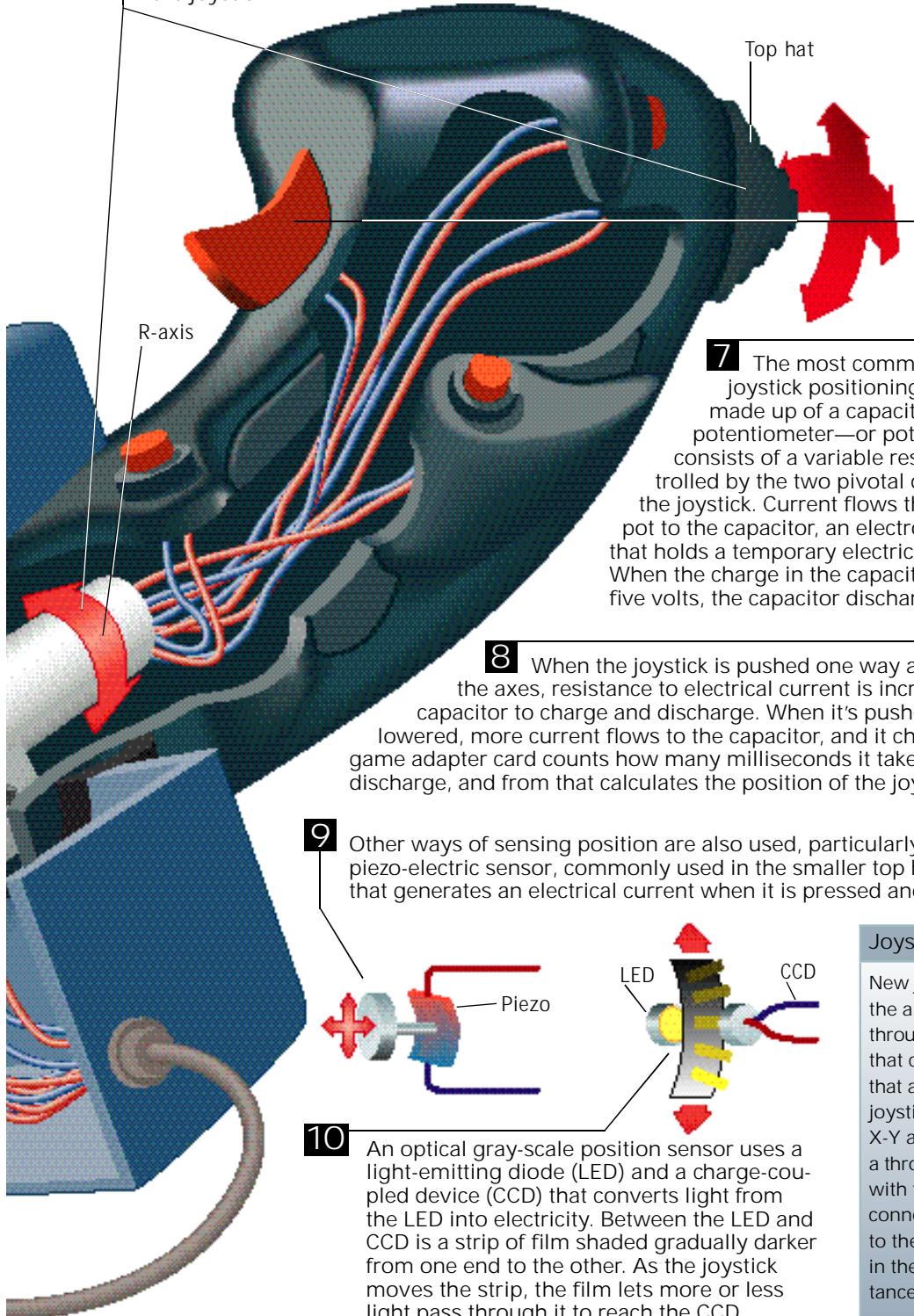
- 2** The base of the handle is connected to a yoke with pivots that allow the joystick to move freely in any direction. Other types of game controllers, such as steering wheels and game pads, may look different from a joystick, but they produce signals using the same devices and connections a joystick does.

- 3** Position sensors attached to each axis of the joystick respond to the joystick's X-Y coordinates and send signals to the game adapter card that the software uses to interpret the position of the game controller.



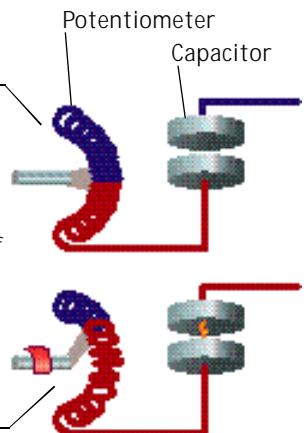
- 4** Through the use of a Y-adapter, two joysticks can be attached to a single PC. Signals for the X-Y coordinates of each joystick are sent along different wires in the same cable to the 15-pin connector on the game card.

**5** More advanced joysticks use both sets of X-Y axis signals—one set for communicating the joystick position, and the other to communicate the position of the top hat, an additional control that's maneuvered with the thumb. Some controllers also track the rotational movement (R-axis) of the joystick.



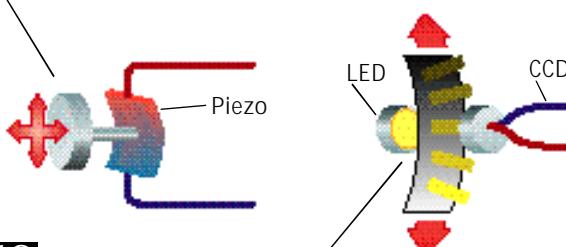
**6** Simple contact switches are used for the trigger and buttons on the joystick. If a button is pressed, the switch sends an electrical signal down a matching wire to the game card. The adapter card, in turn, writes a 1 bit to a specific memory address if the button is not pushed and a 0 bit if it is pushed. What pressing the trigger or buttons does depends on how the game software chooses to use them.

**7** The most common type of joystick positioning sensor is made up of a capacitor and a potentiometer—or pot—which consists of a variable resistor controlled by the two pivotal directions of the joystick. Current flows through the pot to the capacitor, an electronic device that holds a temporary electrical charge. When the charge in the capacitor reaches five volts, the capacitor discharges.



**8** When the joystick is pushed one way along either of the axes, resistance to electrical current is increased, and it takes longer for the capacitor to charge and discharge. When it's pushed the other way, resistance is lowered, more current flows to the capacitor, and it charges and discharges faster. The game adapter card counts how many milliseconds it takes the capacitors to charge and discharge, and from that calculates the position of the joystick on both axes.

**9** Other ways of sensing position are also used, particularly in new digital joysticks. A piezo-electric sensor, commonly used in the smaller top hat control, uses a crystal that generates an electrical current when it is pressed and distorted.



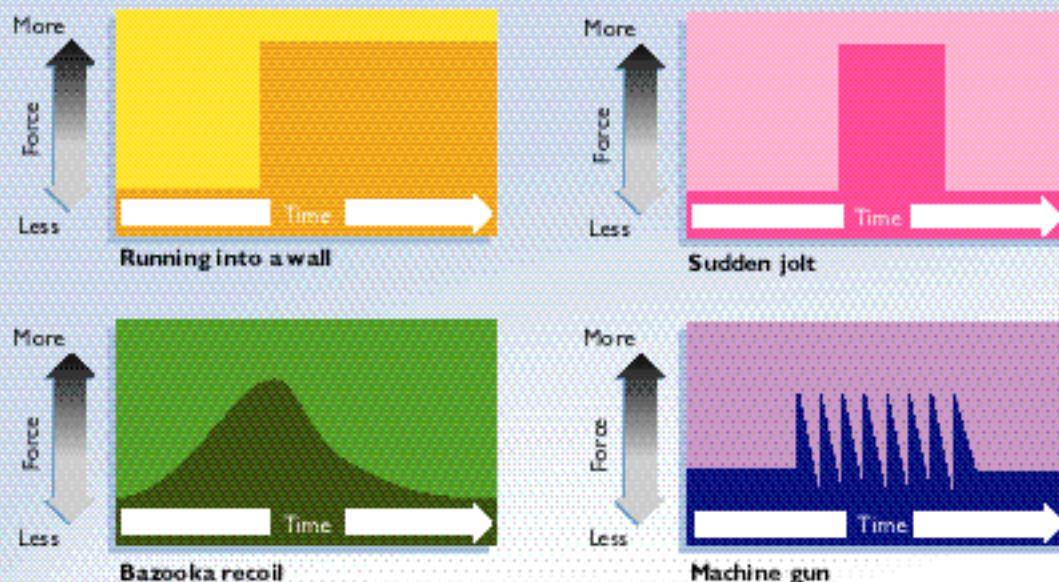
**10** An optical gray-scale position sensor uses a light-emitting diode (LED) and a charge-coupled device (CCD) that converts light from the LED into electricity. Between the LED and CCD is a strip of film shaded gradually darker from one end to the other. As the joystick moves the strip, the film lets more or less light pass through it to reach the CCD.

#### Joysticks of the Future

New joysticks use similar position sensors, but the amount of current produced by them is fed through an analog-to-digital converter (ADC) chip that converts varying voltages into digital values that are then interpreted by the software. Digital joysticks allow the use of more than two pairs of X-Y axes. A Z-axis, for example, may be added as a throttle control. Likely to become more common with the spreading use of universal serial bus connections are controllers that provide feedback to the user so that someone can feel the bumps in the road in a racing game or the wind resistance in a flight simulator.

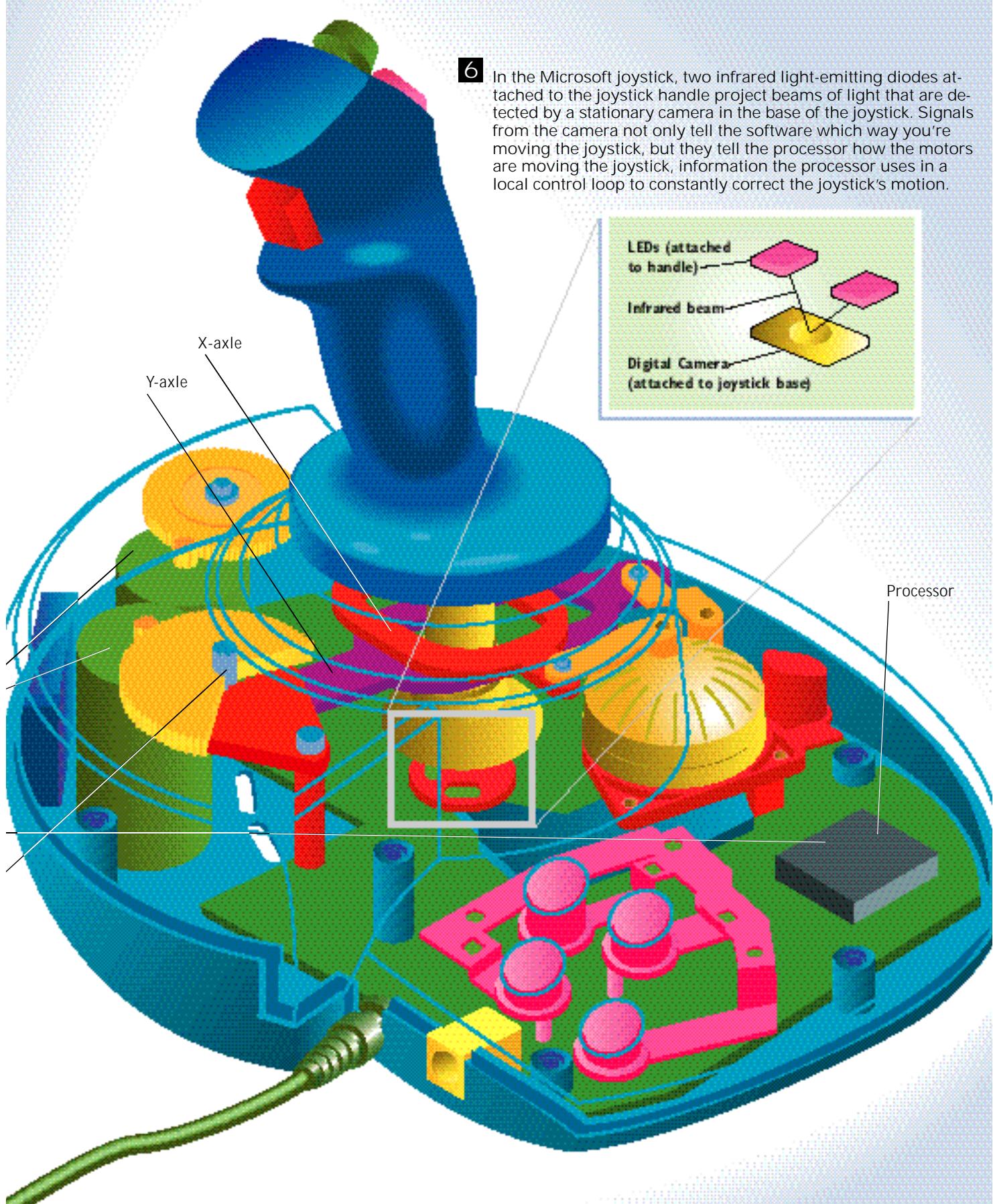
# How Force-Feedback Joysticks Work

- 1** A force-feedback joystick, such as the Microsoft Sidewinder Force Feedback Pro shown here, responds to events in a game by moving the joystick on its own. The instructions to make a joystick move are created in software as wave forms, varying degrees of force graphed over periods of time. Here are examples of waveforms of different feedback forces.



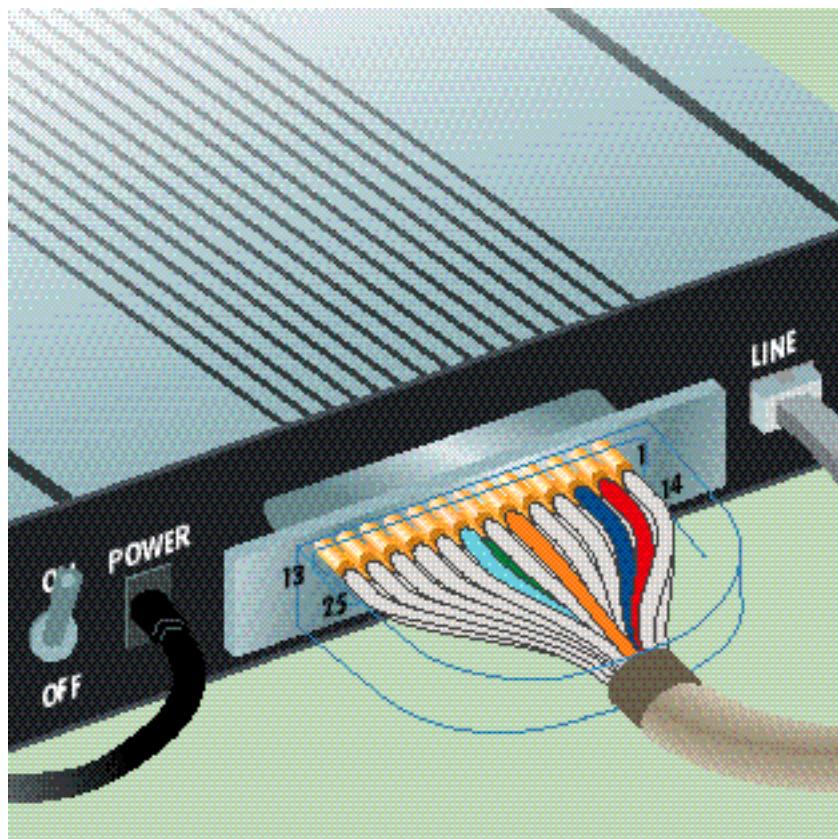
- 2** The game software doesn't have to send the complete instructions for movements to the joystick. Instead, the software sends a token, a much shorter burst of data that identifies the complete wave form the joystick should execute.
- 
- 3** In the Microsoft joystick, a 16-bit, 25MHz microprocessor receives the token and looks up the wave form among 32 movement effects permanently stored on a ROM chip. The software can also download its own original wave forms to a 2K RAM chip for the processor to use.
- 
- 4** The processor follows the instruction of the wave form indicated by the token and sends electronic signals to two motors—one each for the joystick's X-axis and Y-axis.
- 
- 5** The motors convey their precise movements to a gear train that transmits the forces to the joystick's two axles, causing the joystick itself to exert pressure.

- 6 In the Microsoft joystick, two infrared light-emitting diodes attached to the joystick handle project beams of light that are detected by a stationary camera in the base of the joystick. Signals from the camera not only tell the software which way you're moving the joystick, but they tell the processor how the motors are moving the joystick, information the processor uses in a local control loop to constantly correct the joystick's motion.



CHAPTER  
**24**

# How a Modem Works



YOUR PC is a digital device. It accomplishes most of its tasks by turning on or off a series of electronic switches. A binary 0 represents a switch that is turned off; a binary 1 indicates that the switch is on. There is no in-between designation. A graph of digital code would look like this:



The telephone system is an analog device, designed—at a time when digital electronics was unknown—to transmit the diverse sounds and tones of the human voice. Those sounds are conveyed electronically in an analog signal as a continuous electronic current that smoothly varies its frequency and strength. It can be depicted on an oscilloscope as a wavy line, such as this:



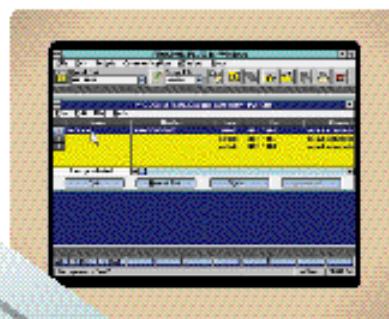
A modem is the bridge between digital and analog signals. It converts on and off digital data into an analog signal by varying, or modulating, the frequency of an electronic wave, a process similar to that used by FM radio stations. On the receiving end of a phone connection, a modem does just the opposite: It demodulates the analog signals back into digital code. The two terms modulate and demodulate combine to give the modem its name.

There are now modems that aren't really modems at all because they work entirely with digital signals, but we call them modems out of habit. One is the ISDN (integrated services digital network) modem. More accurately described as a terminal adapter, an ISDN device takes advantage of special wiring a local phone company can install that sends voice and data as digital, rather than analog, signals. Cable modems, which transmit data over the same cables used to send television signals, are the newest technology for connecting to the Internet.

Modem communications involves three of the least standardized elements of personal computing—serial ports, modem commands, and communications software. The inconsistencies make it impossible to describe one universal way in which all modems work, but the operations discussed here accurately describe two common types of modems. One is the traditional modem, which manages data transfers up to 28.8 kilobits a second. The other is a scheme that involves both analog and pure digital communications for transfer rates nearly twice as fast—in one direction.

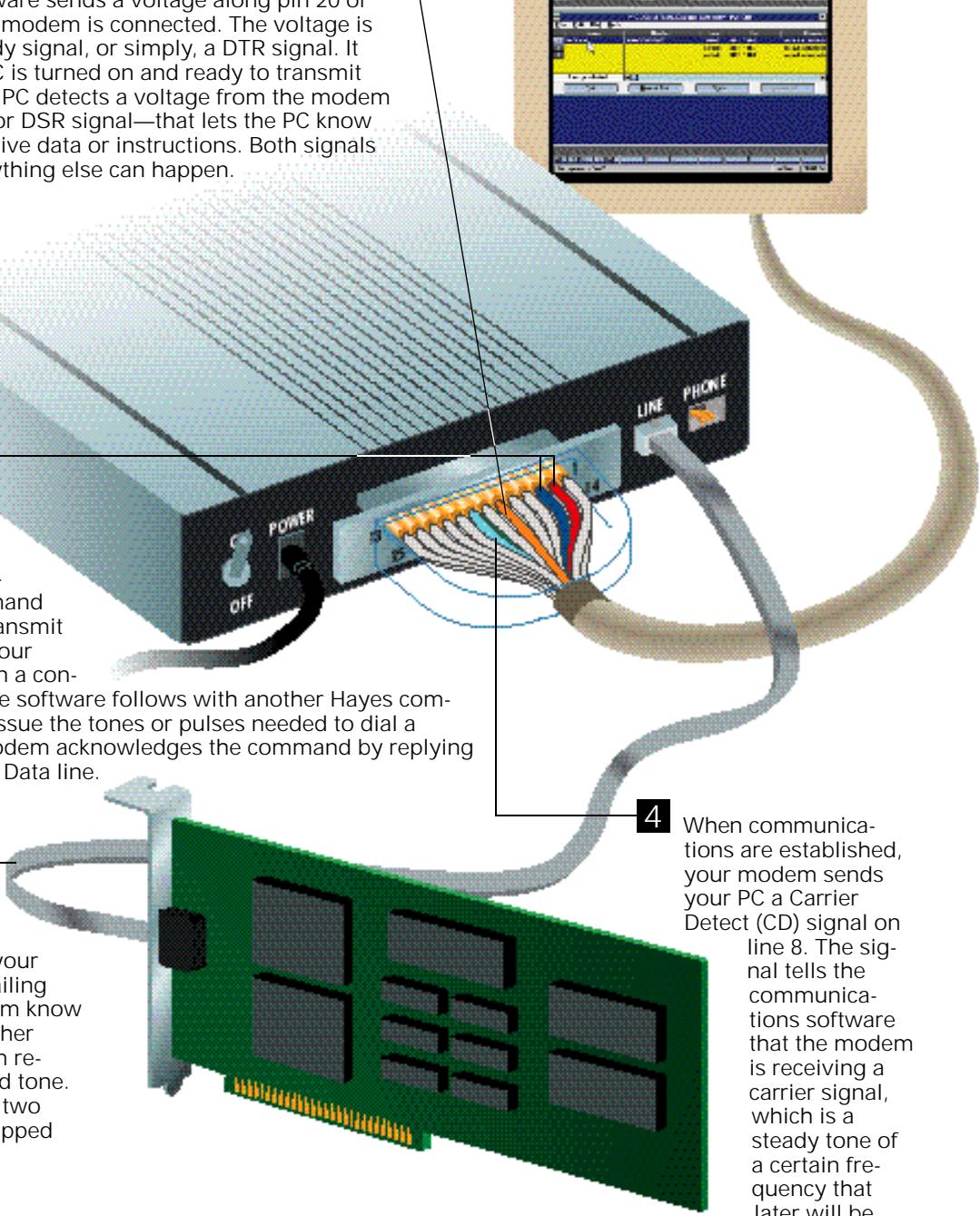
# How a Modem Works

**1** Your communications software sends a voltage along pin 20 of the serial port to which the modem is connected. The voltage is called a Data Terminal Ready signal, or simply, a DTR signal. It tells the modem that the PC is turned on and ready to transmit data. At the same time, the PC detects a voltage from the modem on pin 6—Data Set Ready, or DSR signal—that lets the PC know the modem is ready to receive data or instructions. Both signals must be present before anything else can happen.



**2** Using a standard command language named after the Hayes modems on which it was first popularized, the communications software sends a command to the modem via line 2, the Transmit Data line. The command tells your modem to go off hook—to open a connection with the phone line. The software follows with another Hayes command that tells the modem to issue the tones or pulses needed to dial a specific phone number. The modem acknowledges the command by replying to the PC on line 3, the Receive Data line.

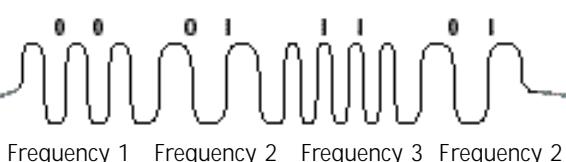
**3** When the modem on the other end of the phone connection—the remote modem—answers the call, your local modem sends out a hailing tone to let the remote modem know that it's being called by another modem. The remote modem responds with a higher-pitched tone. (You can ordinarily hear the two tones if your modem is equipped with a speaker.)



**4** When communications are established, your modem sends your PC a Carrier Detect (CD) signal on line 8. The signal tells the communications software that the modem is receiving a carrier signal, which is a steady tone of a certain frequency that later will be modulated to transmit data.

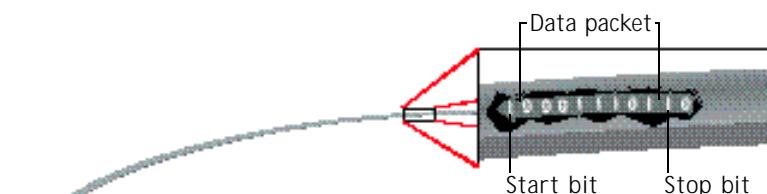


- 5** The two modems exchange information about how they'll send data to each other, a process called a handshake. The two modems must agree on the transfer speed, the number of bits that make up a data packet, how many bits will signal the beginning and end of a packet, whether the modems will use a parity bit for error checking, and whether they will operate at half-duplex or full duplex. If the local and remote systems do not use the same settings, either they'll wind up sending characters that make no sense or they'll refuse to communicate at all.



**Transmission Speed** Although transmission speeds are often expressed in baud—the number of frequency changes occurring during one second—that term is outdated and bits per second is more accurate today. The transmission rate on early modems of 300 bits per second was achieved by sending one frequency to indicate a 0 bit and a different frequency to indicate a 1 bit. The analog signal of a phone line, however, can't change frequencies more than 600 times a second. This is a serious limitation that affects the transmission rate. It has necessitated different schemes to increase the rate at which data is sent.

Group Coding permits different frequencies to stand for more than one bit at a time. For 1,200 bits-per-second transmissions, for example, signals are actually sent at 600 baud, but four different frequencies are used to represent the four different possible pairs of binary bits: 0 and 0, 0 and 1, 1 and 0, and 1 and 1. A similar scheme matches more frequencies with more binary combinations to achieve 2,400 bits per second. For still faster transmission rates, the two modems must both use the same method of compressing data by recognizing frequently repeated patterns of zeros and ones and using shorter codes to stand for those patterns.



**Start/Stop Bits** Each data packet uses a single bit to signal the start of a character and either one bit or two bits to signal the end of a character. The example here uses one stop bit.

**Data Bits** Communications systems may use either seven bits or eight bits to represent a data packet. In this example, eight data bits are used.

**Parity Bit** As a form of error correction, the two systems may agree to use even parity, odd parity, or no parity at all. If they agree on even or odd parity, both systems add up the bits contained in the character and then add another bit called the parity bit. It may be either a 0 bit or a 1 bit, whichever is needed, to make the total either an even number or an odd number, depending on the parity that the systems agree on. The parity bits are used for error checking.

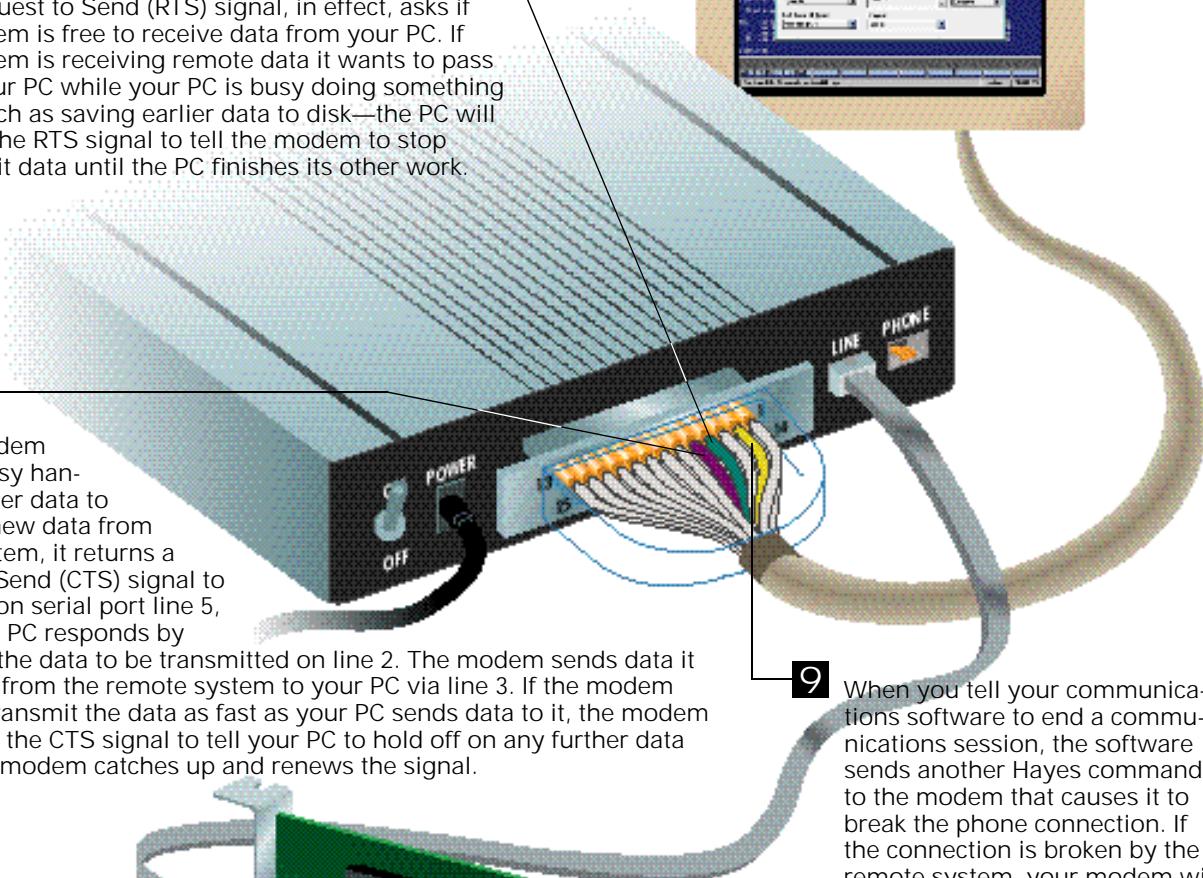
**Half/Full Duplex** The two systems must agree which is responsible for displaying text on the local computer. One system must be set for full duplex and the other set for half-duplex. The system using full duplex is responsible for displaying text on both systems and echoes any text sent to it by the half-duplex system. If the two systems don't use complementary duplex settings, either no text will appear on the local system or each character will appear twice.

[Continued on next page.]

# How a Modem Works

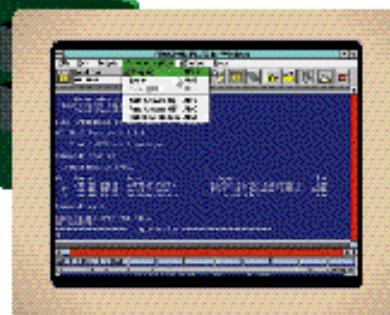
(Continued)

- 6** When the communications software wants to send data, it first sends voltage to line 4 on the serial port. This Request to Send (RTS) signal, in effect, asks if the modem is free to receive data from your PC. If the modem is receiving remote data it wants to pass on to your PC while your PC is busy doing something else—such as saving earlier data to disk—the PC will turn off the RTS signal to tell the modem to stop sending it data until the PC finishes its other work.



- 7** Unless your modem is too busy handling other data to receive new data from your system, it returns a Clear to Send (CTS) signal to your PC on serial port line 5, and your PC responds by sending the data to be transmitted on line 2. The modem sends data it received from the remote system to your PC via line 3. If the modem cannot transmit the data as fast as your PC sends data to it, the modem will drop the CTS signal to tell your PC to hold off on any further data until the modem catches up and renews the signal.

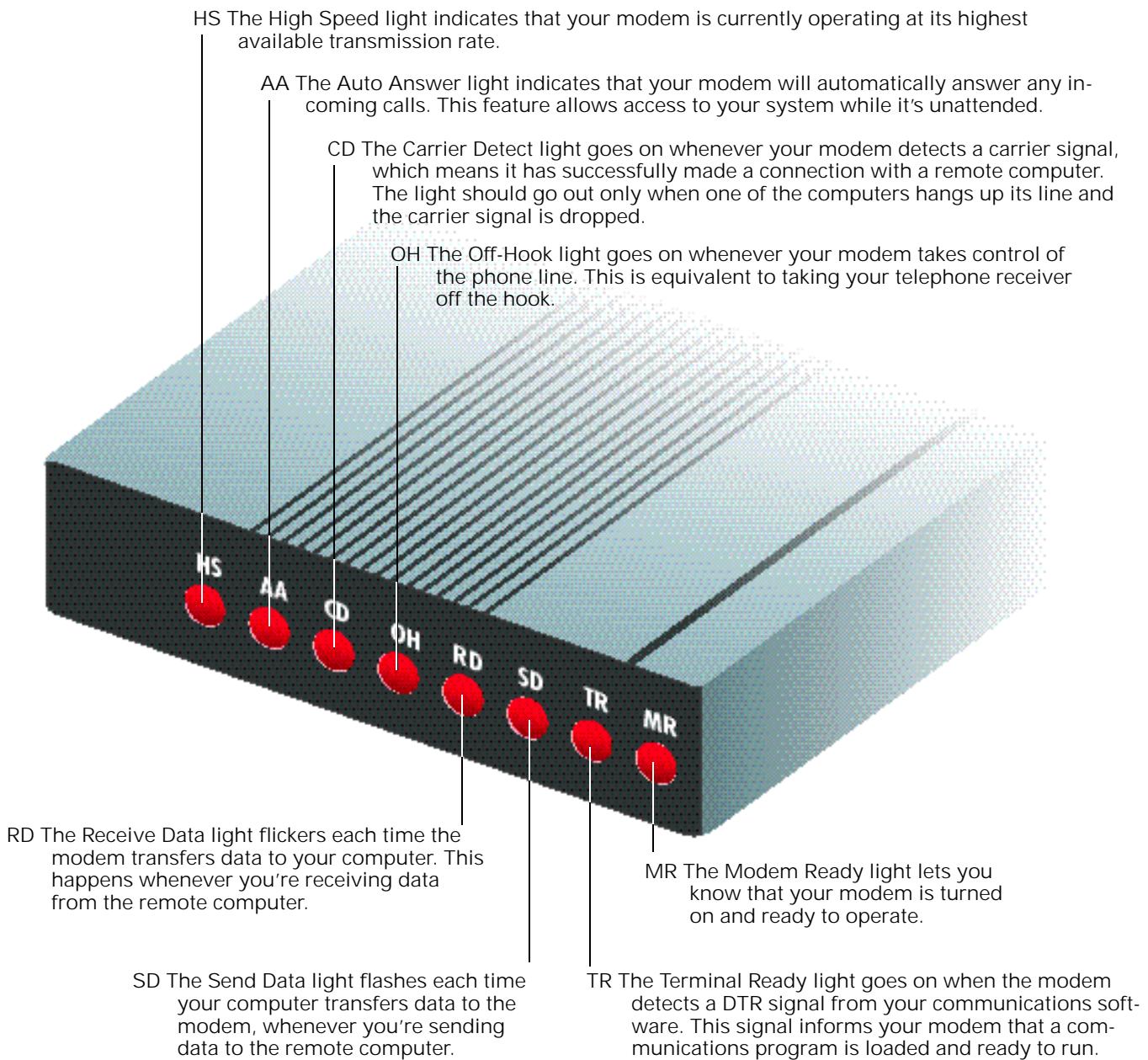
- 8** At the other end of the phone line, the remote modem hears incoming data as a series of tones with different frequencies. It demodulates these tones back into digital signals and sends them to the receiving computer. Actually, both computers can send signals back and forth at the same time because the use of a standard system of tones allows modems on either end to distinguish between incoming and outgoing signals.



- 9** When you tell your communications software to end a communications session, the software sends another Hayes command to the modem that causes it to break the phone connection. If the connection is broken by the remote system, your modem will drop the Carrier Detect signal to your PC to tell the software that communications are broken.

## Reading Your Modem Lights

The indicator lights on the front of an external modem tell you what's happening during your communications session. The exact locations of the lights and the order in which they appear varies from modem to modem. But they are usually labeled with two-character abbreviations. Here's what the most common ones mean.

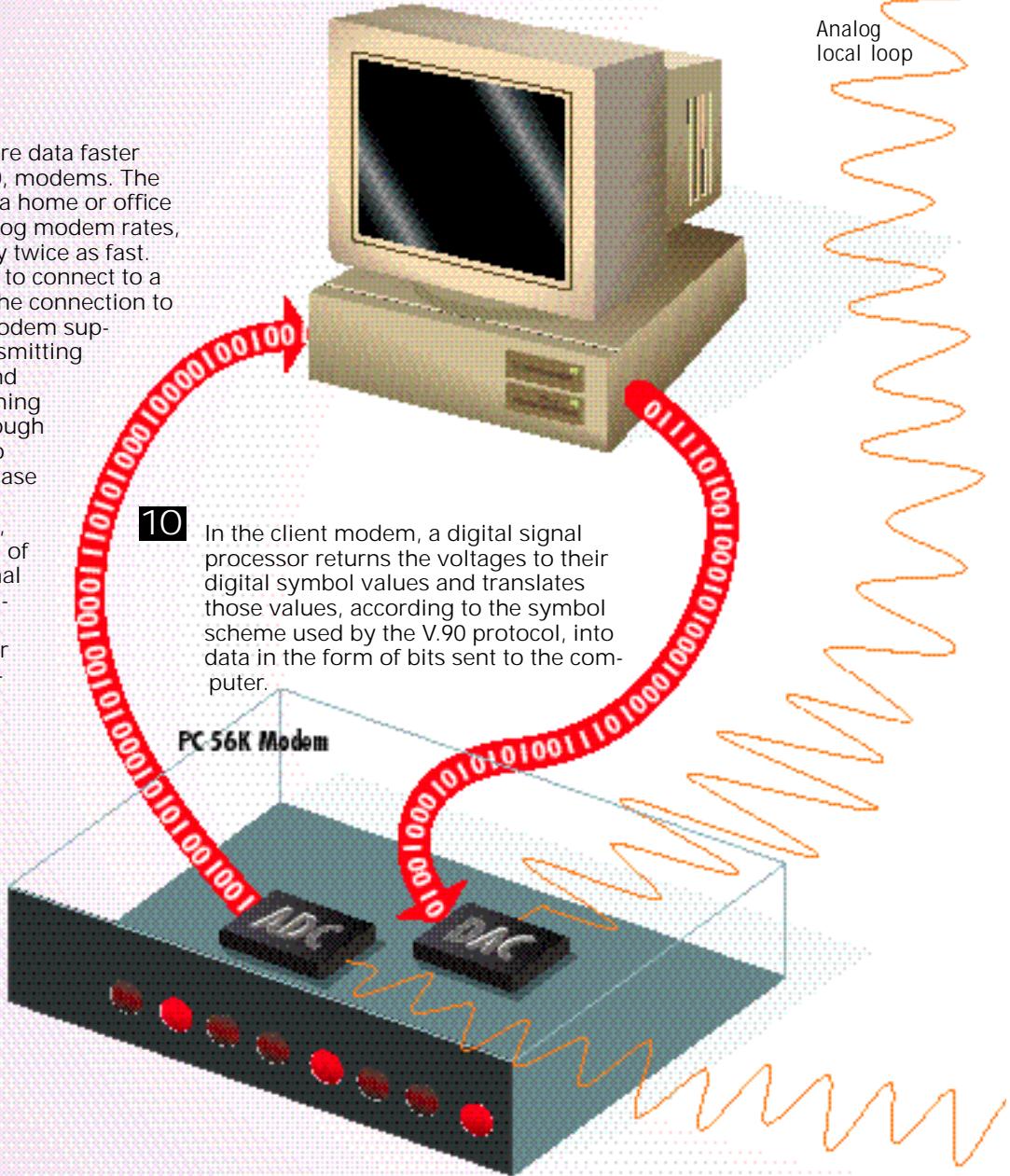


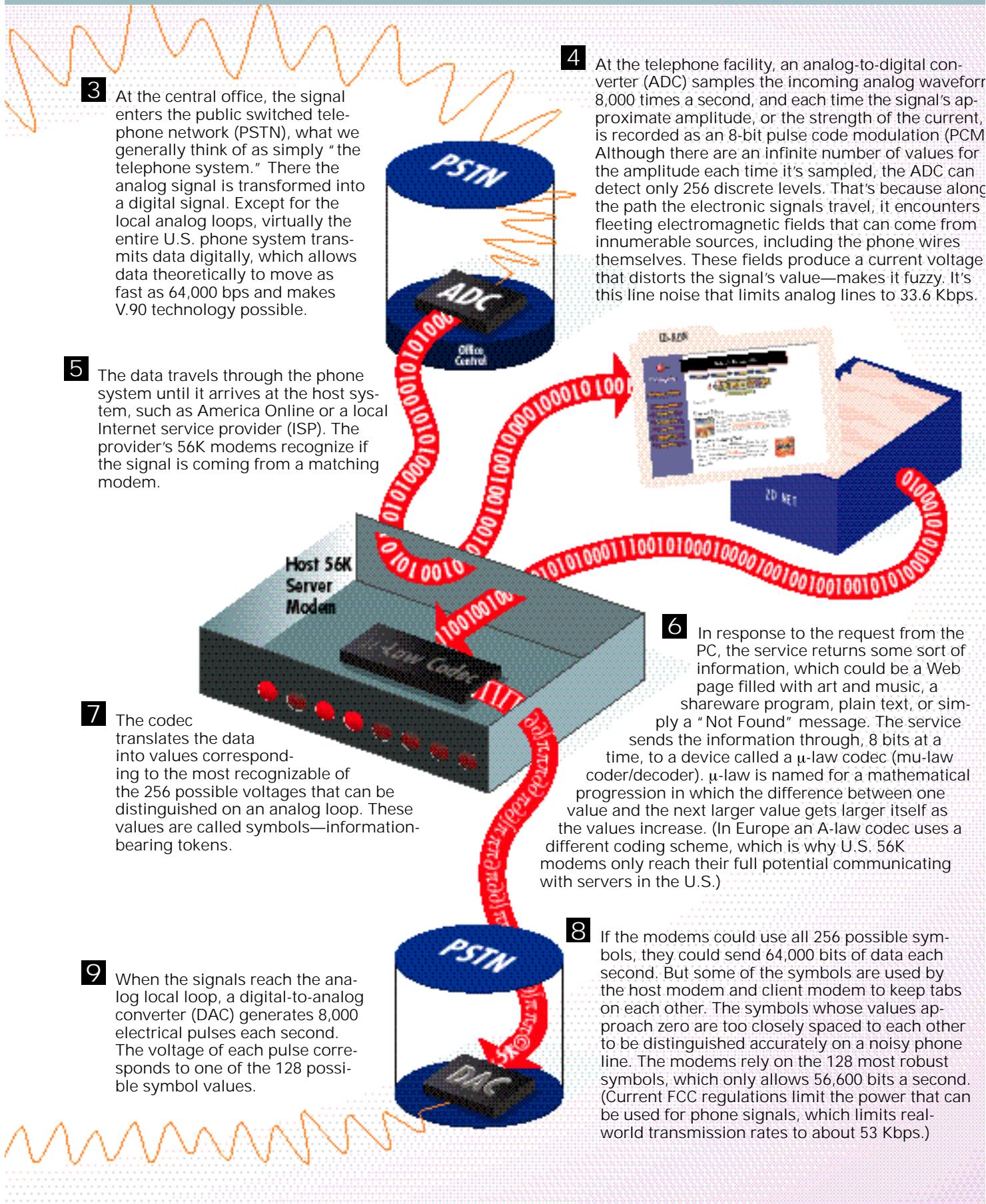
# How a 56K Modem Works



**1** We have a new way to pull more data faster from the Internet—56K, or V.90, modems. The technology allows a person at a home or office PC to send data at normal analog modem rates, but get information back nearly twice as fast. When a PC uses a 56K modem to connect to a host, the modem first probes the connection to determine whether the host modem supports the V.90 standard of transmitting 56,600 bits per second (bps) and whether any of the signals coming from the host to the PC go through any conversions from digital to analog to digital again. In the case of multiple conversions, 56K transmissions are not possible, and the modems on either end of the connection default to normal transmissions. But if a 56K connection is possible, the PC and host modems synchronize their timing to help ensure the accuracy of the downloaded data.

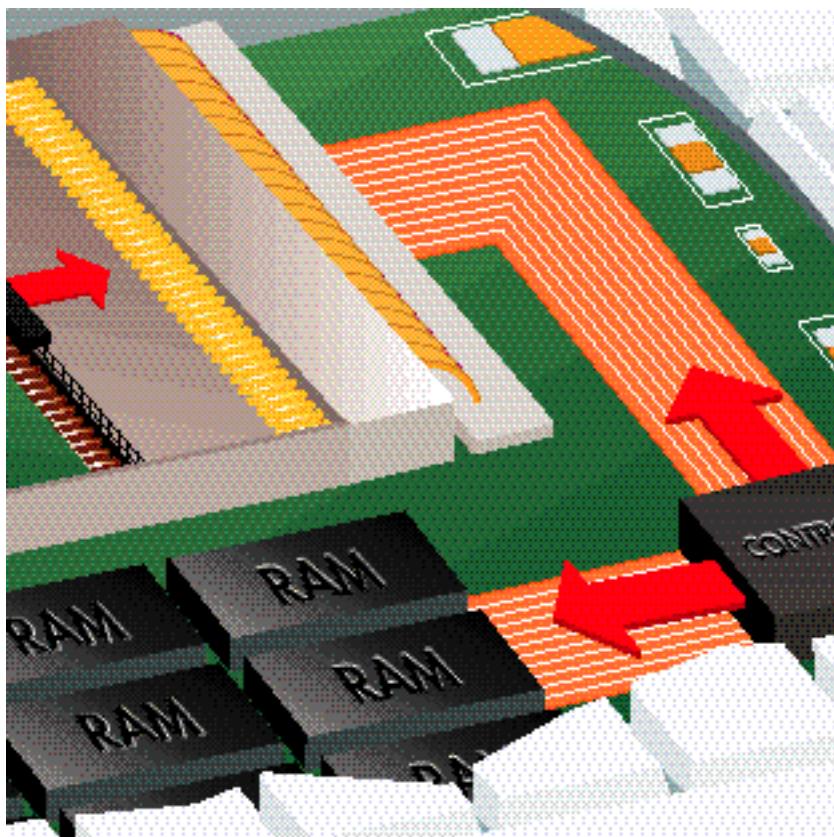
**2** The PC sends a message or a request, which begins as digital data in the PC. The modem converts the bits to an ordinary telephone analog signal. The request travels from the PC's modem along two twisted copper wires that lead to a local telephone company central office. This stretch of wire is called the analog local loop and the outgoing signals that travel on it, under ideal conditions and with data compression, are limited to 36,000 bps.





CHAPTER  
**25**

# How a PC Card (PCMCIA) Works



**BARELY** bigger than a credit card, the PC Card is a miracle of microcomputing miniaturization. These cards contain anything from a modem or network adapter to a solid-state hard disk. They're all packed into a tiny package and made to work with your portable PC, increasing the versatility of portables while increasing their weight by only an ounce or so. Plug in one card and your notebook has a new built-in modem. Plug in another and it has a network adapter or a scanner.

The most common place to find a slot for a PC Card is on a portable computer, naturally. But they're not limited to pint-sized PCs. A PC Card slot on a notebook and another on a desktop mean you can use the same PC Card on either. The advantage: When you take a trip, you can slip that hard drive PC Card out of your desktop PC and into your portable. You take all your work with you without the hassle of feeding floppies into both computers or running a file-transfer cable between their parallel ports.

Originally, PC Cards were called PCMCIA Cards, one of the more unlovely of computer acronyms. It was supposed to stand for Personal Computer Memory Card Interface Association, but others thought of a different meaning for it: People Can't Master Computer Industry Acronyms. It's a shame the device began life with such a terrible name, and truthfully "PC Card" isn't much better because it sounds too much like "Expansion Card."

But name snafus aside, PC Cards are terrific inventions because one of the ultimate goals of computing is to make the buggers so tiny and lightweight that they'll be like Dick Tracy's wrist radio. Digital signal processing—which lets the same device process different kinds of communications—may in the future allow a single card to function as different types of devices. We've still got a way to go to witness those miracles, but the important thing to remember right now is that if you're buying a portable computer, make sure it has at least one PC Card slot—two are even better. There are different types of slots; some are made to accommodate a slightly thicker card. Before buying any PC Card notebook, check out what cards are available for the type of slot the computer is sporting.

# How a PC Card Works

1 At the factory, most of the circuitry in the PC Card may be installed in the form of RAM, in which case the card is designed for use as a solid-state drive, such as the one shown here. Some PC Cards actually contain a real, spinning hard drive. Alternatively, the circuitry may function as a modem, network adapter, SCSI adapter, or other peripheral.

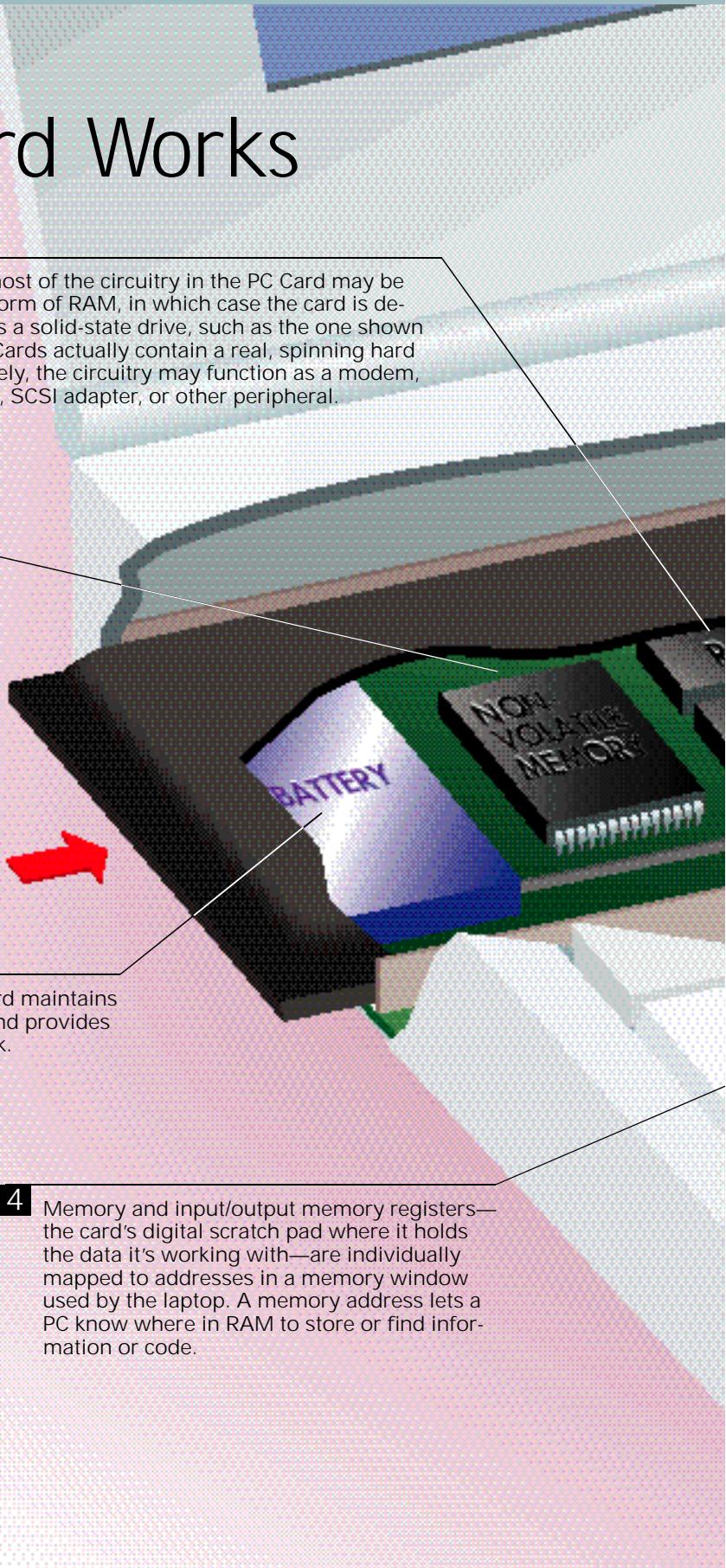
2 In nonvolatile memory—which doesn't lose its contents when power is turned off—the PC Card stores configuration information about itself so that no jumpers or DIP switches need to be changed to set it up. This information determines how the laptop accesses the card. The data is coordinated with a device driver that's run on the laptop during boot-up so the portable PC can have access to the card. A device driver is a software extension to the operating system that lets DOS or Windows know how to communicate with a specific piece of hardware.

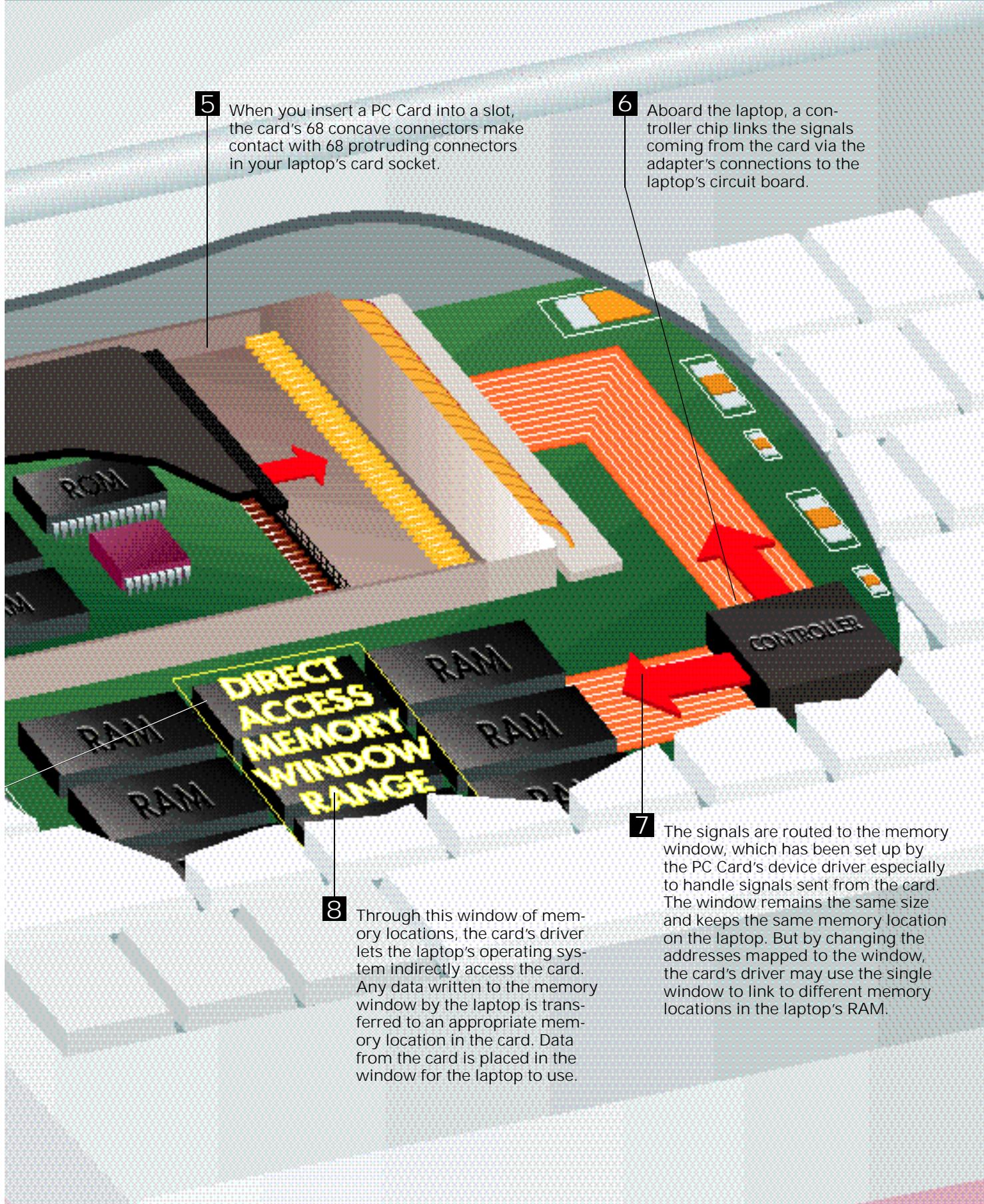
3 A battery built into the card maintains data stored in any RAM and provides power for the card to work.

4 Memory and input/output memory registers—the card's digital scratch pad where it holds the data it's working with—are individually mapped to addresses in a memory window used by the laptop. A memory address lets a PC know where in RAM to store or find information or code.

## Pick a Card, Any Card

Theoretically, systems that support the PC Card standard can have up to 255 adapter plugs, each capable of connecting to as many as 16 card sockets. That means, hypothetically, a single system could have up to 4,080 PC Cards attached to it.

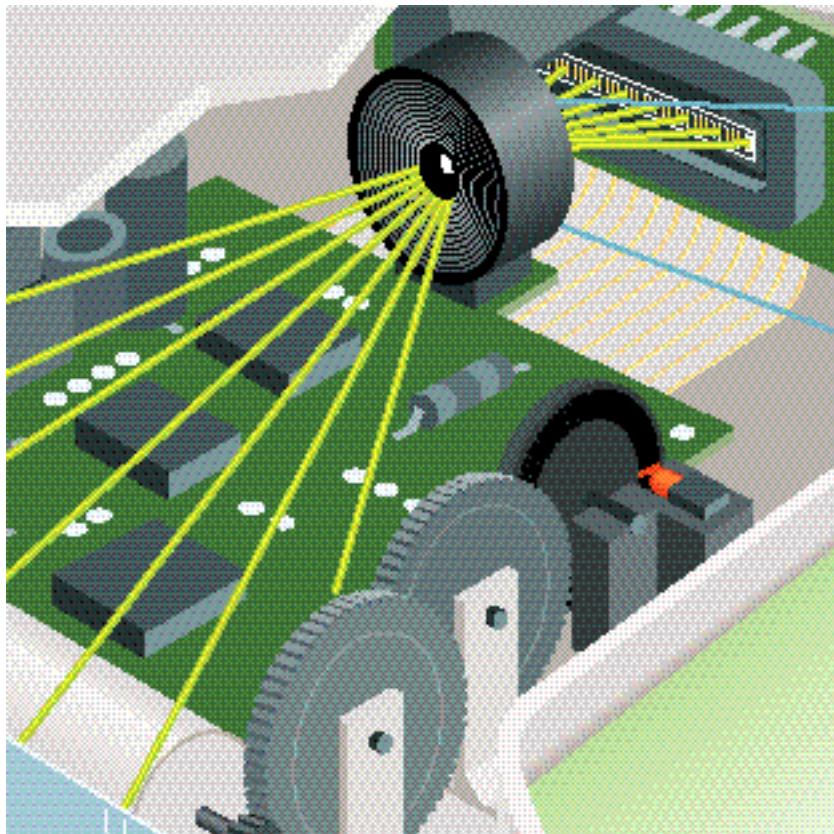




CHAPTER

**26**

# How Scanners and Optical Character Recognition Work



**SCANNERS** are the eyes of your personal computer. They allow a PC to convert a drawing or photograph into code that a graphics or desktop publishing program can use to both display the image on the screen and reproduce the image with a graphics printer. Or a scanner can let you convert printed type into editable text with the help of optical character recognition (OCR) software.

The three basic types of scanners differ primarily in the way that the page containing the image and the scan head that reads the image move past each other. In a sheet-fed scanner, mechanical rollers move the paper past the scan head. In a flatbed scanner, the page is stationary behind a glass window while the head moves past the page, similar to the way a copying machine works. Hand-held scanners rely on the human hand to move the scan head.

Each method has its advantages and disadvantages. The flatbed scanner requires a series of mirrors to keep the image that is picked up by the moving scan head focused on the lens that feeds the image to a bank of sensors. Because no mirror is perfect, the image undergoes some degradation each time it is reflected. But the advantage of a flatbed scanner is that it can scan oversized or thick documents, such as a book. With a sheet-fed scanner, the image is captured more accurately, but you're limited to scanning single, ordinary-sized sheets.

A hand-held scanner is a compromise. It can scan pages in books, but often the scanning head is not as wide as that in either a flatbed or a sheet-fed scanner. Most hand-scanner software automatically combines two half-page scans into a single image. The hand-held scanner incorporates mechanisms to compensate for the unsteadiness of most hands, and it's generally less expensive than other types of scanners because it doesn't require a mechanism to move the scan head or paper.

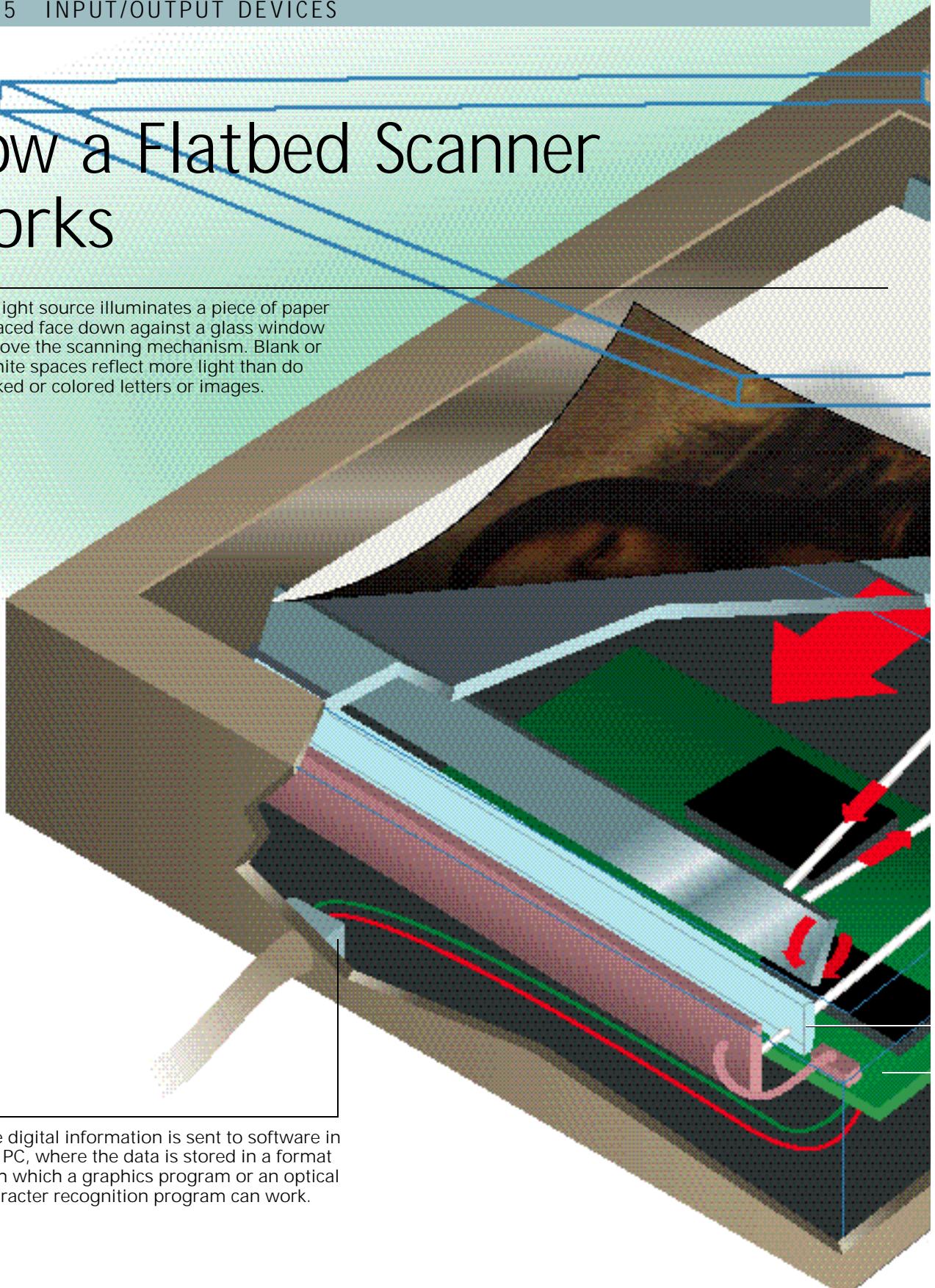
A scanner's sophistication lies in its ability to translate an unlimited range of analog voltage levels into digital values. Some scanners can distinguish between only black and white, useful just for text. More precise models can differentiate shades of gray. Color scanners use red, blue, and green filters to detect the colors in the reflected light.

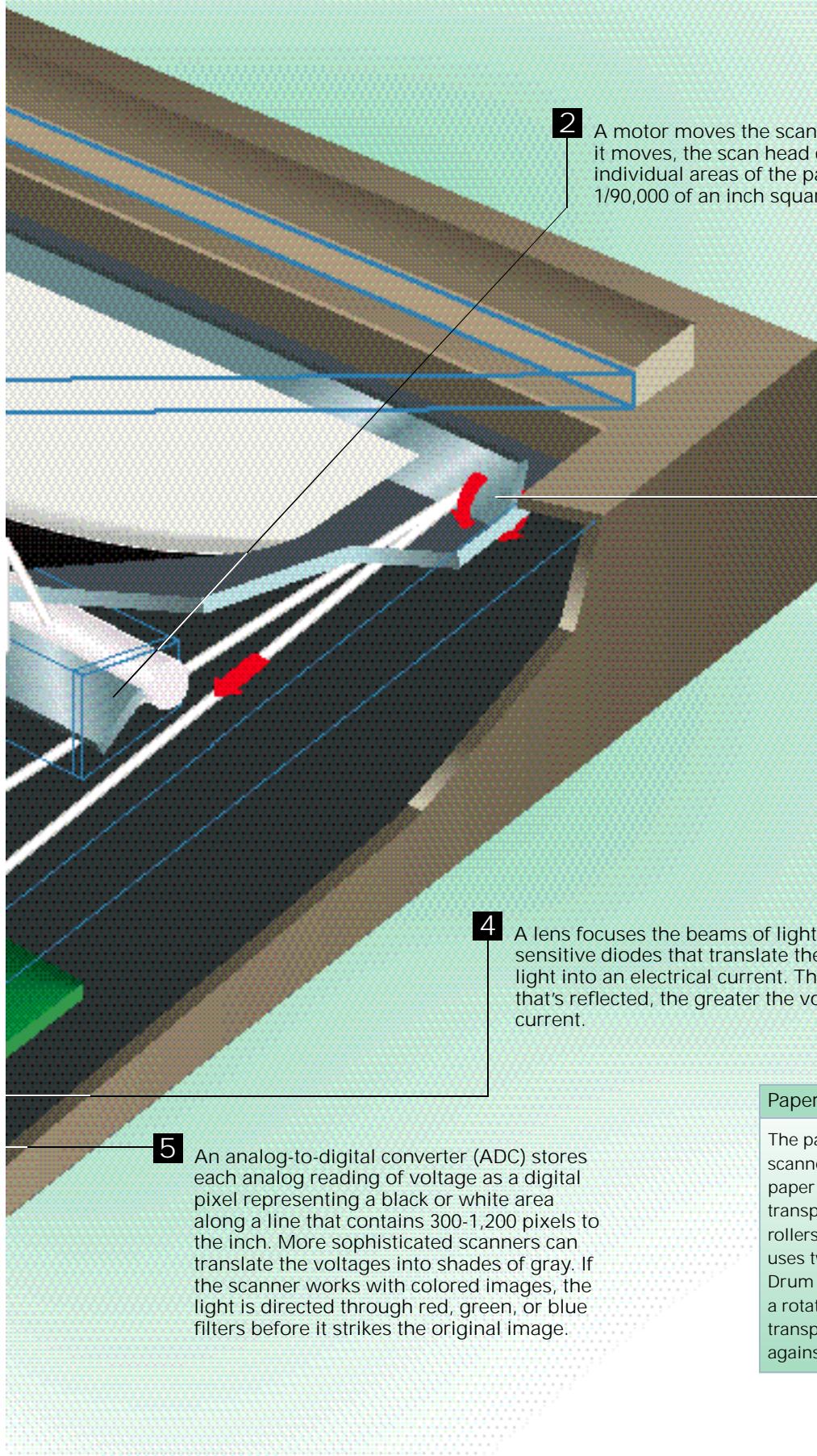
Regardless of a scanner's sensitivity to gray or how the head and paper move, the operations of all scanners are basically simple and similar. We'll look at two that are representative of the technologies involved—a flatbed scanner and a hand-held grayscale scanner. We'll also examine one of the most important reasons for scanning a document—to convert its image into editable text by using optical character recognition software.

# How a Flatbed Scanner Works

- 1 A light source illuminates a piece of paper placed face down against a glass window above the scanning mechanism. Blank or white spaces reflect more light than do inked or colored letters or images.

- 6 The digital information is sent to software in the PC, where the data is stored in a format with which a graphics program or an optical character recognition program can work.





2 A motor moves the scan head beneath the page. As it moves, the scan head captures light bounced off individual areas of the page, each no larger than 1/90,000 of an inch square.

3 The light from the page is reflected through a system of mirrors that must continually pivot to keep the light beams aligned with a lens.

4 A lens focuses the beams of light onto light-sensitive diodes that translate the amount of light into an electrical current. The more light that's reflected, the greater the voltage of the current.

5 An analog-to-digital converter (ADC) stores each analog reading of voltage as a digital pixel representing a black or white area along a line that contains 300-1,200 pixels to the inch. More sophisticated scanners can translate the voltages into shades of gray. If the scanner works with colored images, the light is directed through red, green, or blue filters before it strikes the original image.

#### Paper Movers

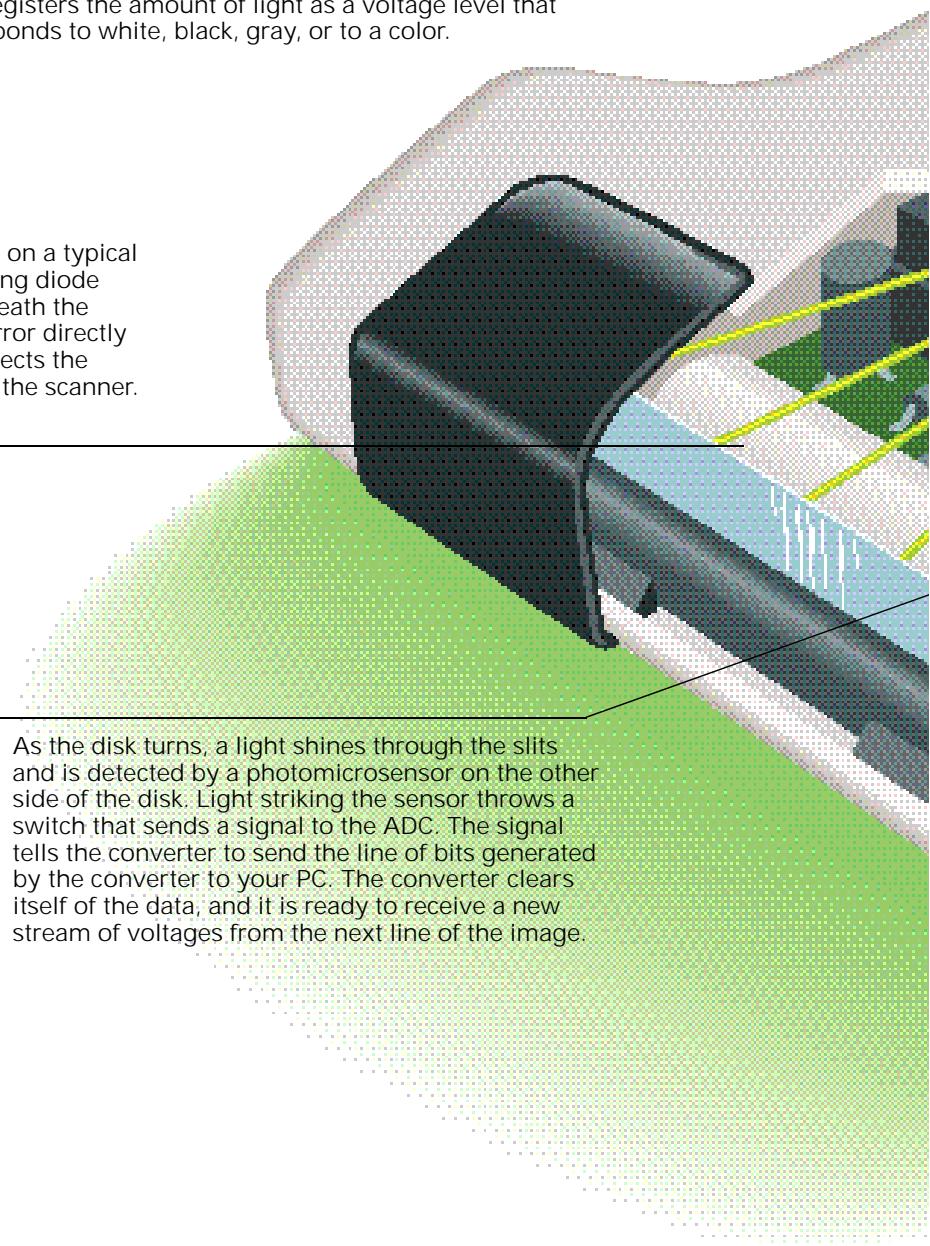
The paper isn't always stationary when it's being scanned. Higher-end scanners feed sheets of paper past a scan head that stays still. A roller transport feeds a document between two rubber rollers, much like a fax machine. A belt transport uses two opposing belts to do the same thing. Drum transports pass the paper through against a rotating drum. A vacuum straight-through transport uses vacuum tubes to hold the paper against a belt as the belt rolls past the scan head.

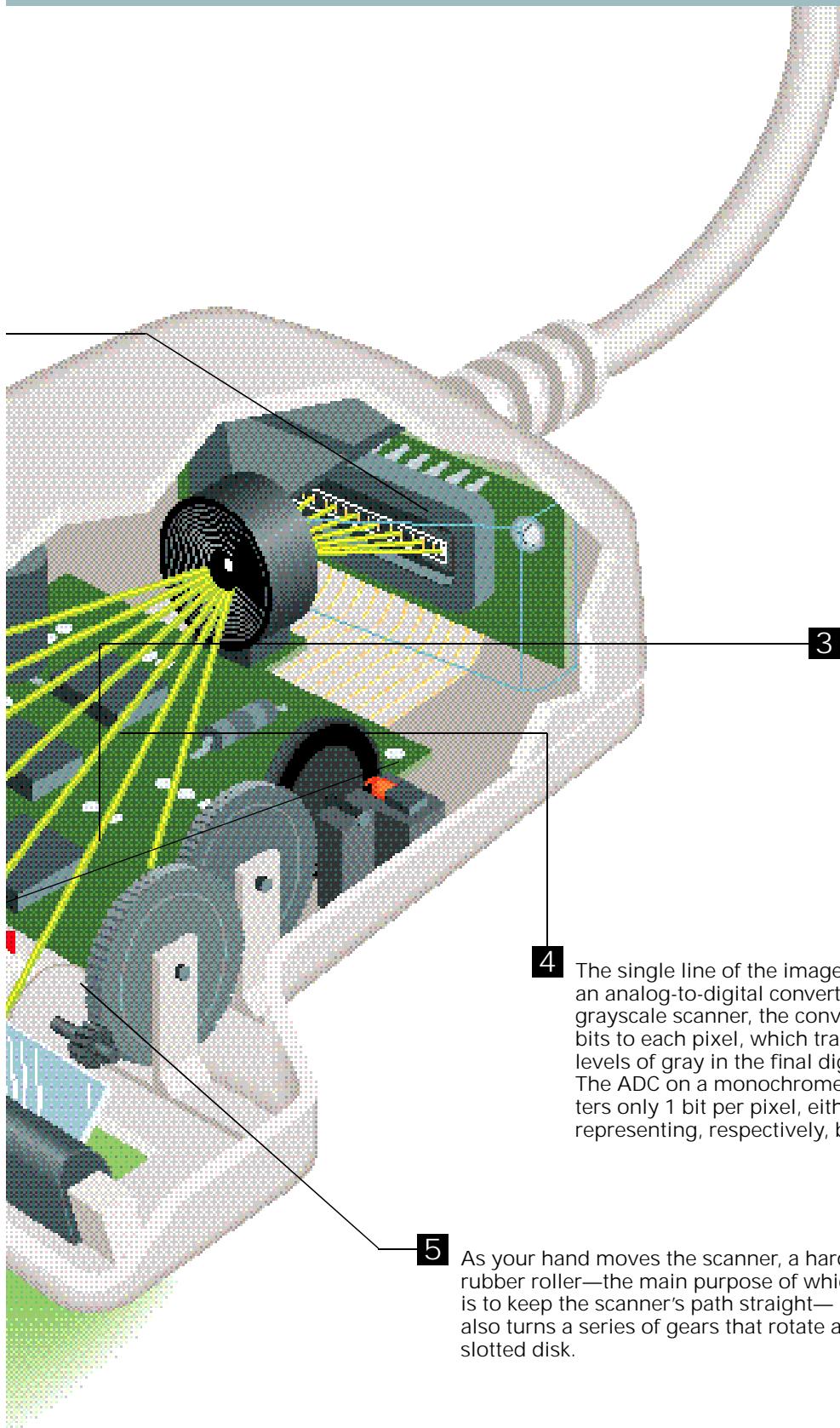
# How a Hand-Held Scanner Works

**2** The lens focuses a single line of the image onto a charge-coupled device (CCD), which is a component designed to detect subtle changes of voltage. The CCD contains a row of light detectors. As the light shines onto these detectors, each registers the amount of light as a voltage level that corresponds to white, black, gray, or to a color.

**1** When you press the scan button on a typical hand-held scanner, a light-emitting diode (LED) illuminates the image beneath the scanner. An inverted, angled mirror directly above the scanner's window reflects the image onto a lens in the back of the scanner.

**6** As the disk turns, a light shines through the slits and is detected by a photomicrosensor on the other side of the disk. Light striking the sensor throws a switch that sends a signal to the ADC. The signal tells the converter to send the line of bits generated by the converter to your PC. The converter clears itself of the data, and it is ready to receive a new stream of voltages from the next line of the image.





**3** The voltages generated by the CCD are sent to a specialized analog chip for gamma correction, a process that enhances the black tones in an image so that the eye, which is more sensitive to dark tones than to light ones, will have an easier time recognizing the image. With some scanners, gamma correction is performed as a software process.

**4** The single line of the image now passes to an analog-to-digital converter (ADC). In a grayscale scanner, the converter assigns 8 bits to each pixel, which translates into 256 levels of gray in the final digitized image. The ADC on a monochrome scanner registers only 1 bit per pixel, either on or off, representing, respectively, black or white.

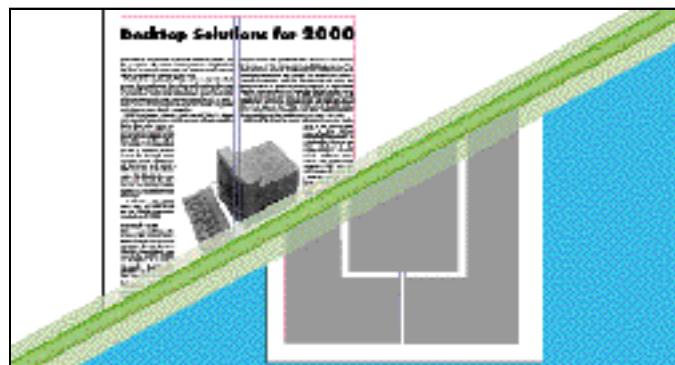
**5** As your hand moves the scanner, a hard rubber roller—the main purpose of which is to keep the scanner's path straight—also turns a series of gears that rotate a slotted disk.

# How Optical Character Recognition Works

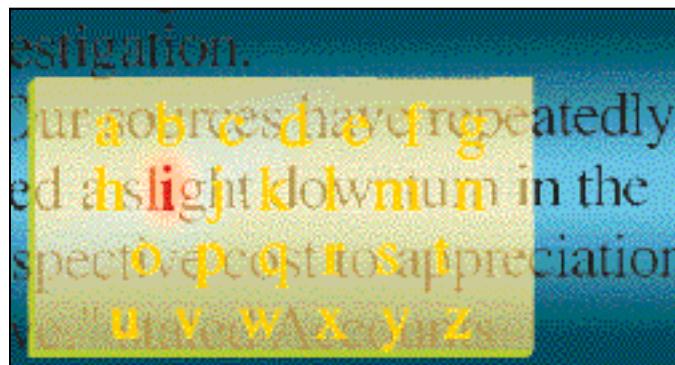
- 1 When a scanner reads in the image of a document, the scanner converts the dark elements—text and graphics—on the page to a bitmap, which is a matrix of square pixels that are either on (black) or off (white). Because the pixels are larger than the details of most text, this process degrades the sharp edges of characters, much as a fax machine blurs the sharpness of characters. This degradation creates most of the problems for optical character recognition (OCR) systems.



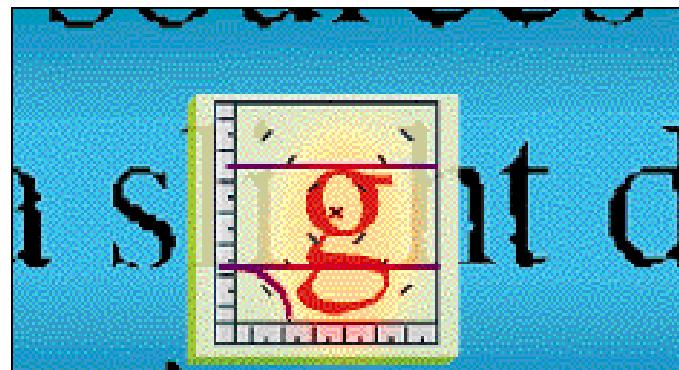
- 2 The OCR software reads in the bitmap created by the scanner and averages out the zones of on and off pixels on the page, in effect mapping the white space on the page. This enables the software to block off paragraphs, columns, headlines, and random graphics. The white space between lines of text within a block defines each line's baseline, an essential detail for recognizing the characters in the text.



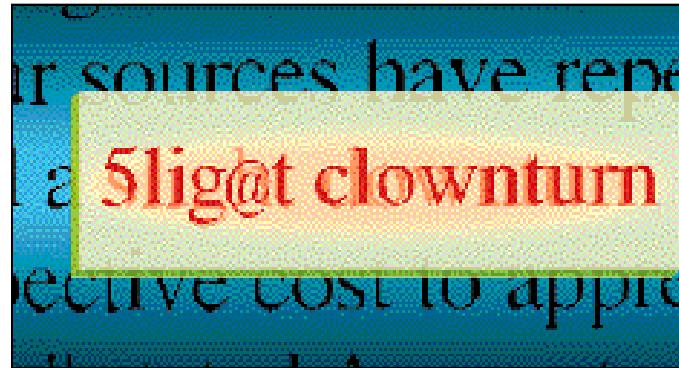
- 3 In its first pass at converting images to text, the software tries to match each character through a pixel-by-pixel comparison to character templates that the program holds in memory. Templates include complete fonts—numbers, punctuation, and extended characters—of such common faces as 12-point Courier and the IBM Selectric typewriter set. Since this technique demands a very close match, the character attributes, such as bold and italic, must be identical to qualify as a match. Poor-quality scans can easily trip up matrix matching.



- 4 The characters that remain unrecognized go through a more intensive and time-consuming process called feature extraction. The software calculates the text's x-height—the height of a font's lowercase x—and analyzes each character's combination of straight lines, curves, and bowls (hollow areas within loops, as in o or b). The OCR programs know, for example, that a character with a curved descender below the baseline and a bowl above it is most likely a lowercase g. As the software builds a working alphabet of each new character it encounters, recognition speed accelerates.



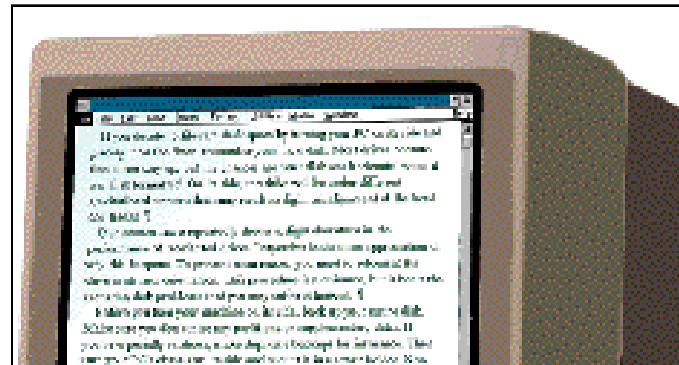
- 5 Because these two processes don't decipher every character, OCR programs take two approaches to the remaining hieroglyphics. Some OCR programs tag unrecognized characters with a distinctive character—such as ~, #, or @—and quit. You must use the search capability of a word processor to find where the distinctive character has been inserted and correct the word manually. Some OCR programs may also display a magnified bitmap onscreen and ask you to press the key of the character needed to substitute for the placeholder character.



- 6 Still other OCR programs invoke a specialized spelling checker to search for obvious errors and locate possible alternatives for words that contain tagged unrecognized characters. For example, to OCR programs, the number 1 and the letter l look very similar, so do 5 and S, or cl and d. A word such as downturn may be rendered as clownturn. A spelling checker recognizes some typical OCR errors and corrects them.



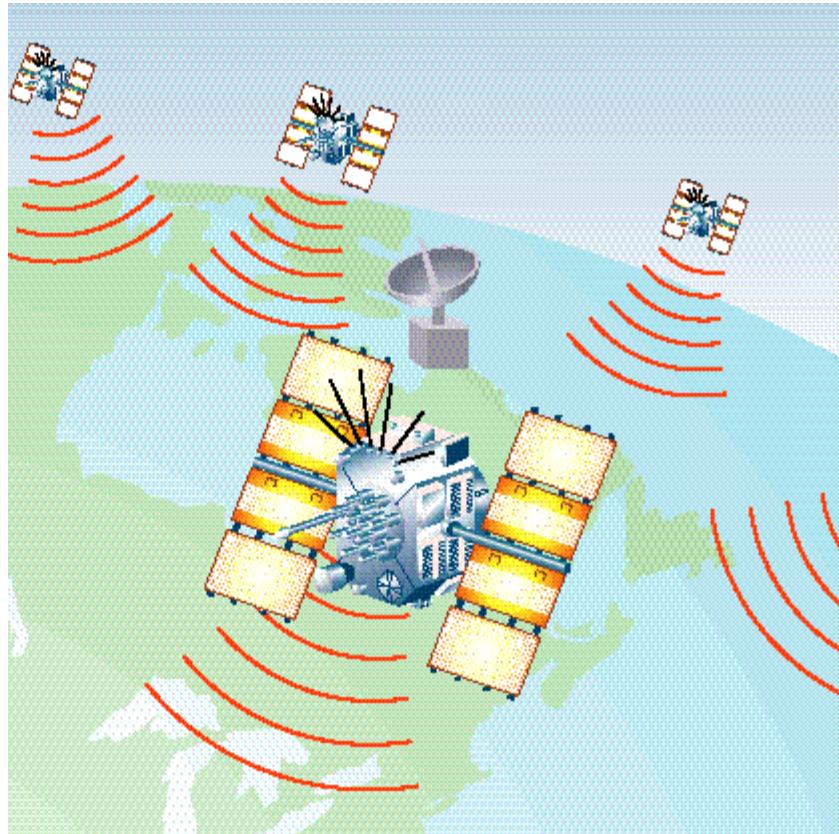
- 7 Most OCR programs give you the option of saving the converted document to an ASCII file or in a file format recognized by popular word processors or spreadsheets.



CHAPTER

27

# How High-Tech Input/Output Works



**THEY'RE** everywhere. Computers are entering areas of our lives we could not have imagined a few years ago. Much of the progress is due to new ways to get information into computers, combined with increased storage capacity and the processing power to handle all that new input.

Digital photography is just taking off. The best digital cameras range in cost from expensive to ridiculous. But the technology is no mystery, and digital cameras will replace conventional film cameras as capabilities rise and prices fall. Eventually, you can expect several facets of everyday life—computers, television, communications, and news—to converge into some ubiquitous appliance. (“Convergence” is the buzzword these days.) When this happens, we’ll think nothing of looking at our family pictures on a computer-converged television rather than in a scrap album.

Voice recognition is likely to wind up as the end-all, sci-fi way to enter information, text, and commands into a computer. (See Chapter 29, where voice recognition is covered as a form of multimedia.) But because some things are better left unsaid, look for a spread of gesture recognition—the ability of a PC to recognize characters and drawings created with a pen-like stylus or even the tip of your finger.

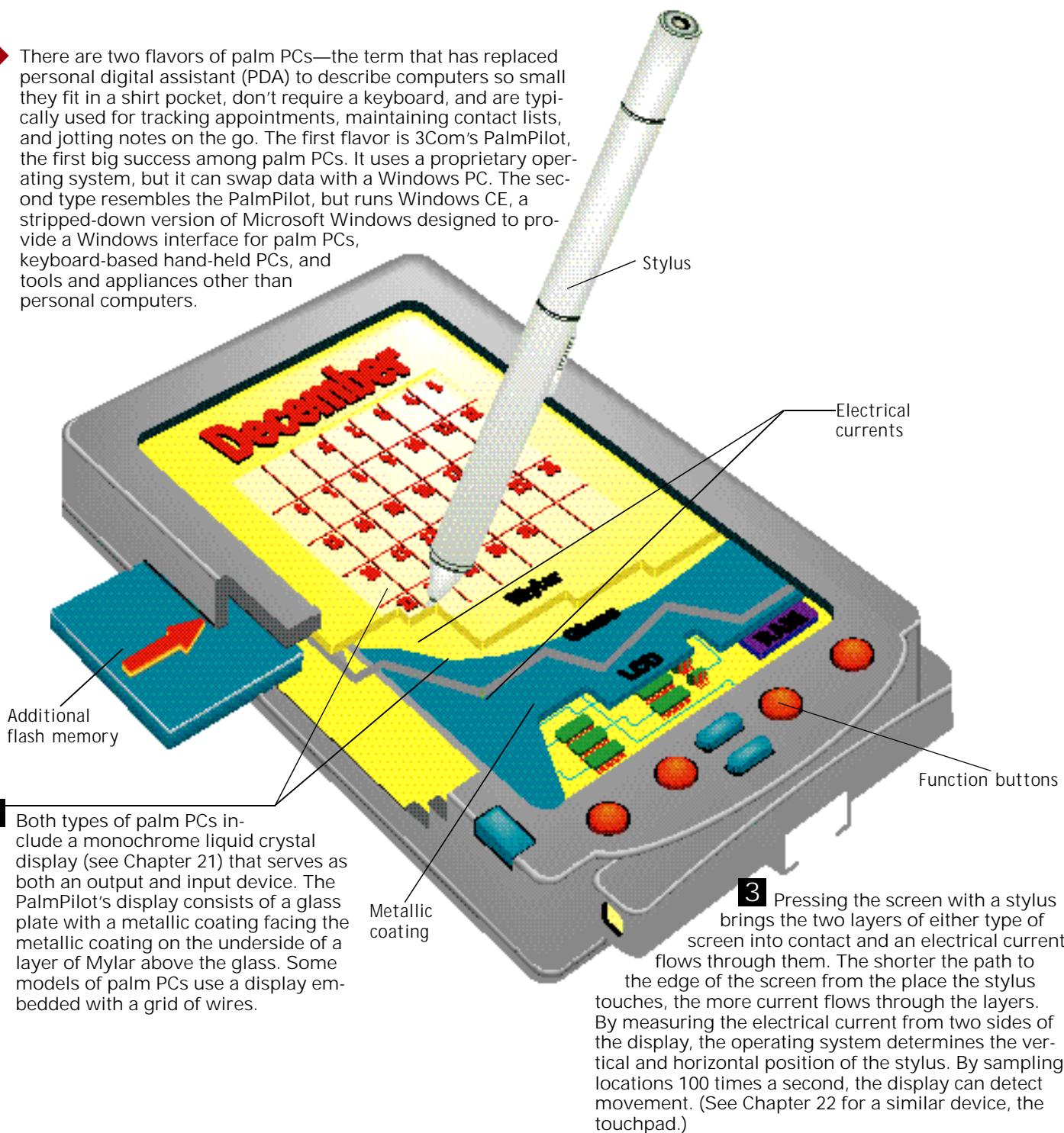
The most amazing new input is the global positioning system (GPS), another technology that would have been science fiction a decade ago. GPS consists of satellites locked in orbit over the earth that send out signals a computer—whether a PC or a microchip in a car—can interpret to find out where it is, and by extension, where you are.

And in a development that would have made the creator of “Dick Tracy” proud, fingerprint recognition is a reality. Two types of fingerprint recognition exist: Systems that match a fingerprint from a known person to a record, a useful tool for controlling entrance to sensitive data or access to machines you don’t want unknown fingers manipulating. The other type of fingerprint recognition is what you’d find at the FBI headquarters to match a fingerprint from an unidentified person compared with millions of records.

This chapter is a hodge-podge of these high-tech methods of getting and using types of information that simply were incomprehensible to all but the most powerful computers a few years ago.

# How Palm PCs Work

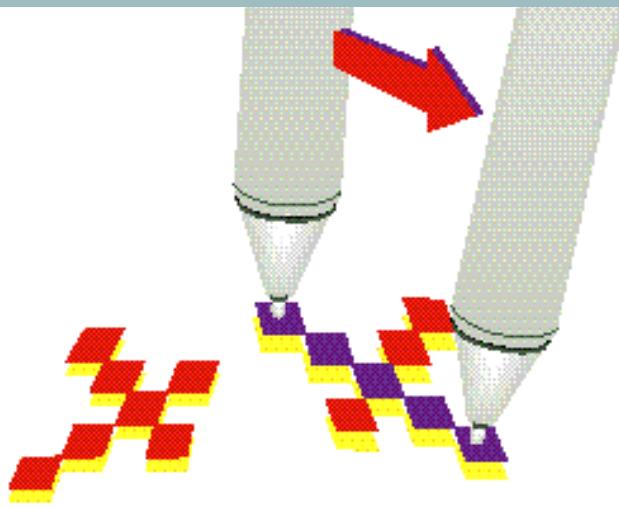
- 1** There are two flavors of palm PCs—the term that has replaced personal digital assistant (PDA) to describe computers so small they fit in a shirt pocket, don't require a keyboard, and are typically used for tracking appointments, maintaining contact lists, and jotting notes on the go. The first flavor is 3Com's PalmPilot, the first big success among palm PCs. It uses a proprietary operating system, but it can swap data with a Windows PC. The second type resembles the PalmPilot, but runs Windows CE, a stripped-down version of Microsoft Windows designed to provide a Windows interface for palm PCs, keyboard-based hand-held PCs, and tools and appliances other than personal computers.



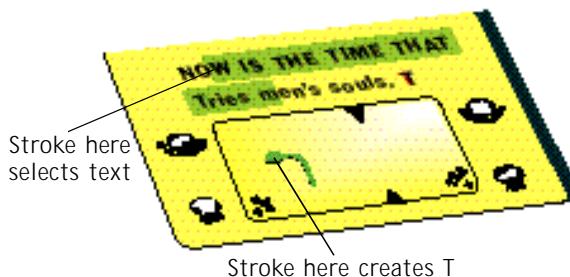
- 2** Both types of palm PCs include a monochrome liquid crystal display (see Chapter 21) that serves as both an output and input device. The PalmPilot's display consists of a glass plate with a metallic coating facing the metallic coating on the underside of a layer of Mylar above the glass. Some models of palm PCs use a display embedded with a grid of wires.

- 3** Pressing the screen with a stylus brings the two layers of either type of screen into contact and an electrical current flows through them. The shorter the path to the edge of the screen from the place the stylus touches, the more current flows through the layers. By measuring the electrical current from two sides of the display, the operating system determines the vertical and horizontal position of the stylus. By sampling locations 100 times a second, the display can detect movement. (See Chapter 22 for a similar device, the touchpad.)

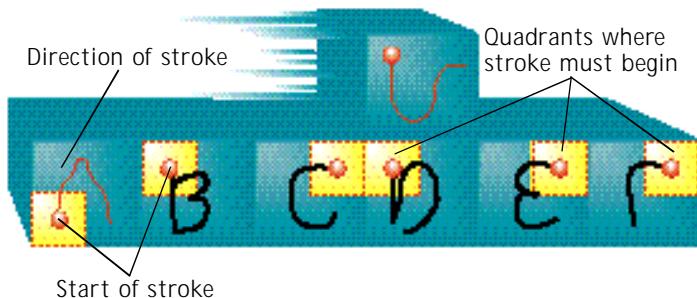
- 4** After reading the position of the stylus, the computer sends a signal to the screen to turn on the pixels at the pen's position, a process called showing ink. As the pen moves, the computer continually calculates its position and instructs other pixels to turn on. The computer distinguishes between pixels that match the stylus's positions (called the input plane) and pixels that are turned on by the application (the output plane). Note that the screen does not contain two separate, physical levels of LCD pixels; rather, the distinction between input and output planes is a logical one. The same pixels are used for input and output planes. The operating system keeps track of on which of the two logical planes the pixels are being used. With this arrangement, a user can draw a rough circle on the input plane, and the application will respond by erasing the input and drawing a perfect circle on the output plane.



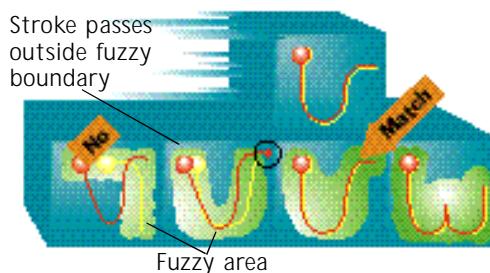
- 5** Before interpreting the stylus stroke, the operating system looks at what application, such as Calendar or Notepad, is active, and where the stroke is taking place. A movement in an area dedicated to handwriting recognition could, for example, represent a T. But the same stroke on another part of the screen could select text to be deleted.



- 6** If the stylus is in an area used for handwriting recognition, the operating system compares the keystroke to a database of shapes that correspond to the alphabet, numbers, and punctuation. PalmPilot uses a character set called Graffiti; Windows CE palm PCs use Jot. Both character sets are designed to be simpler and easier to recognize than normal handwriting, while retaining enough similarity to the actual characters to make the script easy to remember.

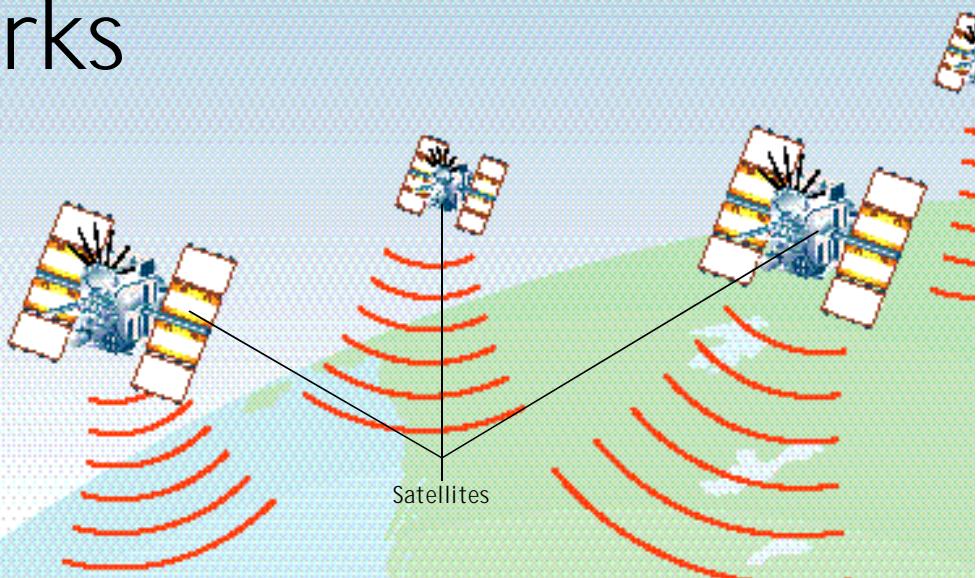


- 7** The character shapes in Graffiti and Jot are based on thousands of samples of handwriting, and the results are intentionally fuzzy. The actual strokes can vary from the ideal and still be recognized, within limits. When the recognition software finds a match in its database, it turns on the pixels to display a normally formed version of that character on the LCD screen's output plane.

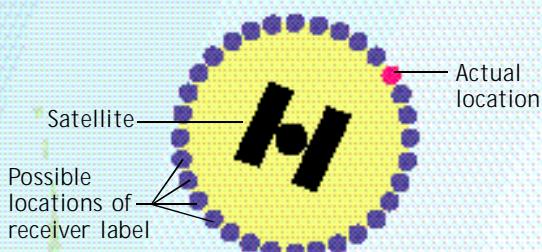


# How a Global Positioning System Works

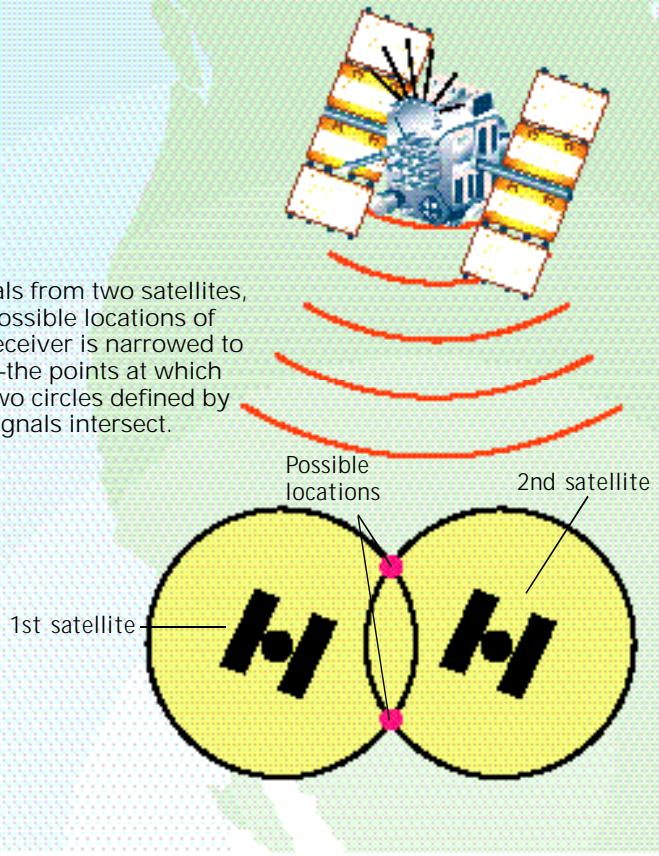
**1** The Global Positioning System (GPS) is a collection of 24 satellites managed by the Defense Department that blanket the earth with telltale radio signals so that anyone equipped with a receiver can determine his exact location. Positioned around the earth in a giant geodesic dome pattern, the satellites are in geosynchronous orbit so that relative to the earth and each other, they maintain the same steady positions.



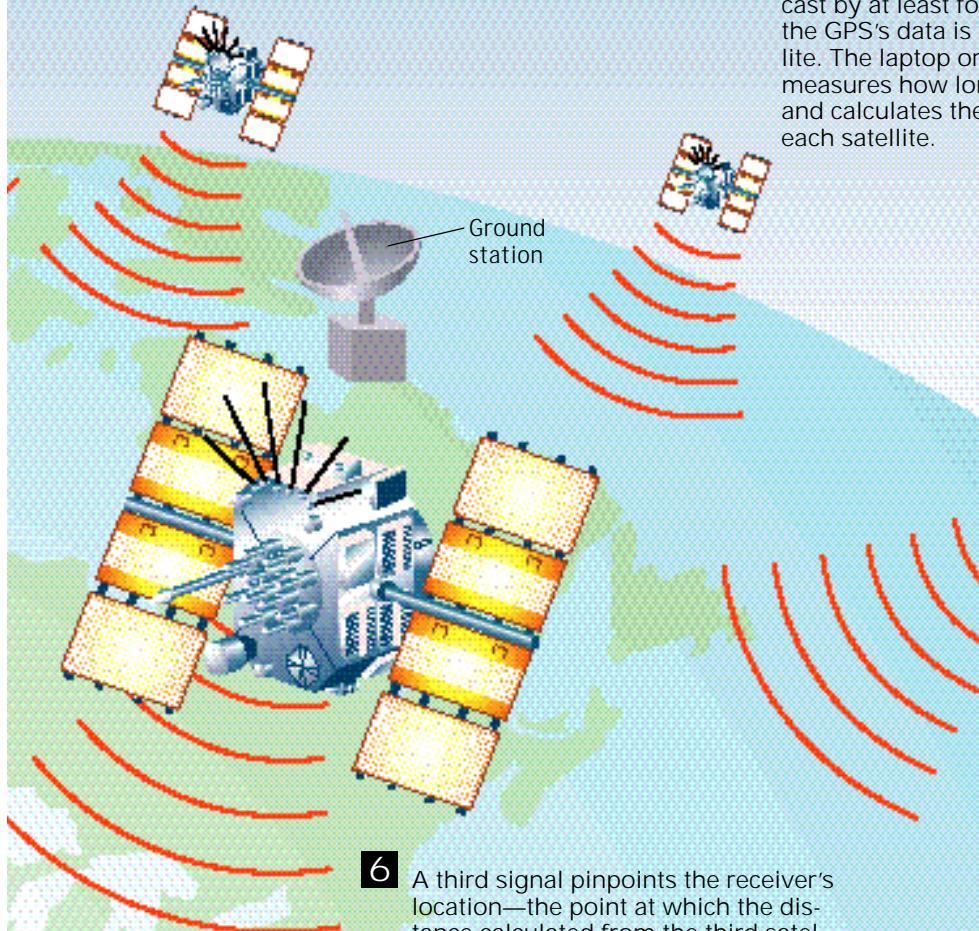
**4** Based on a signal from only one satellite, the receiver could be at any point that matches the distance between that one satellite and the receiver, creating a circle of possible locations centered on the satellite.



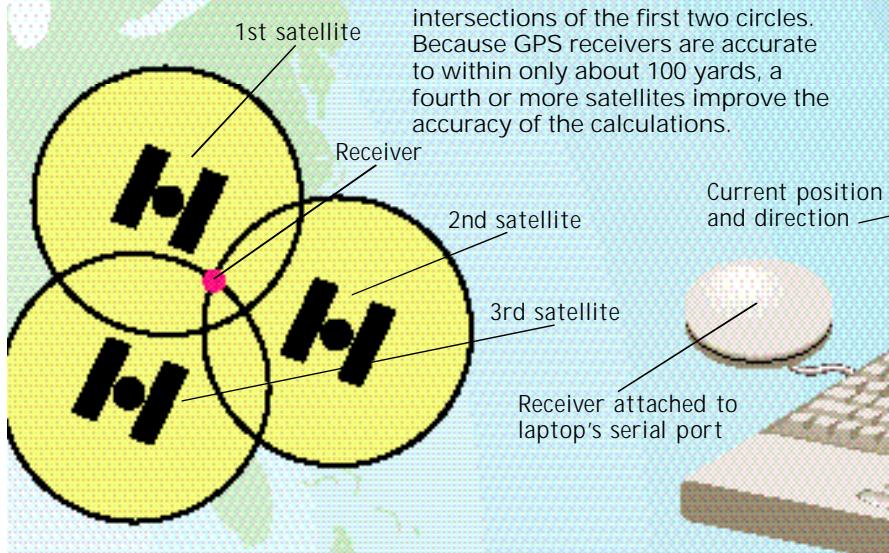
**5** Signals from two satellites, the possible locations of the receiver is narrowed to two—the points at which the two circles defined by the signals intersect.



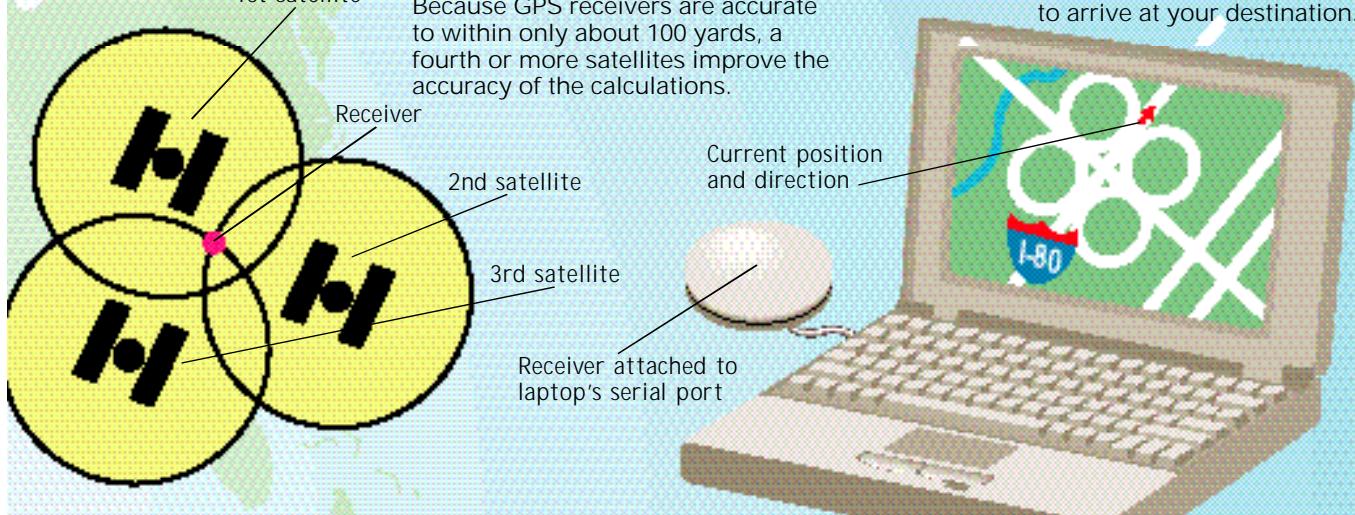
- 2** From any point on earth, between five and eight satellites are visible—or would be if your eyesight were good enough or if you used a telescope. Every thousandth of a second, the satellites broadcast information that identifies each one and the time the signal is sent. A ground station constantly updates and corrects the timing signals.



- 3** On earth, a GPS radio receiver, which can be attached to a laptop computer or built into a stand-alone device, listens to the signals broadcast by at least four satellites. Programmed into the GPS's data is the exact location of each satellite. The laptop or a specialized microcomputer measures how long it took the signals to reach it and calculates the distances from the receiver to each satellite.

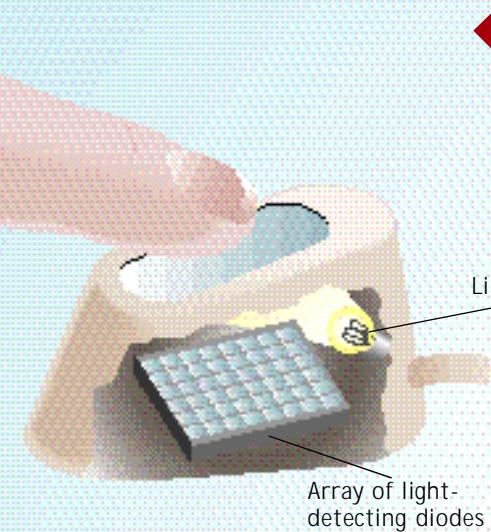


- 6** A third signal pinpoints the receiver's location—the point at which the distance calculated from the third satellite coincides with one of the two intersections of the first two circles. Because GPS receivers are accurate to within only about 100 yards, a fourth or more satellites improve the accuracy of the calculations.



- 7** The longitude and latitude determined by the processor are mapped and turned into a moving point of light on a digital map. As a car or hand-held GPS device moves, its position relative to the satellites changes. Over time, the microprocessor can use this data to calculate speed and direction. Combined with data from mapping software that plots routes, the GPS information can trigger printed and spoken directions for making turns or changing roads, warnings when you leave the path, and estimates of how long it will take to arrive at your destination.

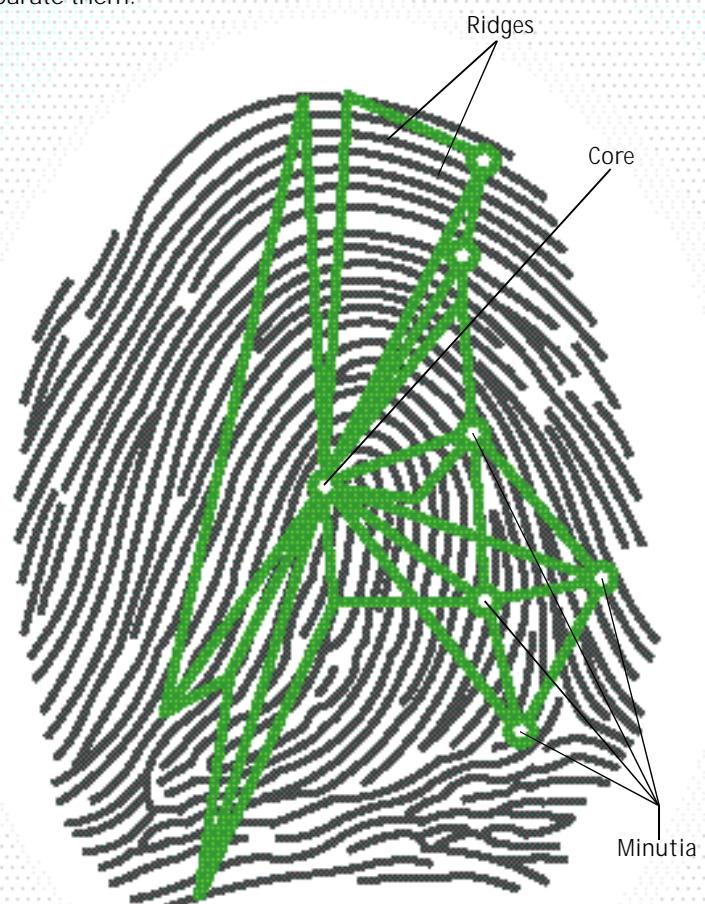
# How Fingerprint Recognition Works

**1**

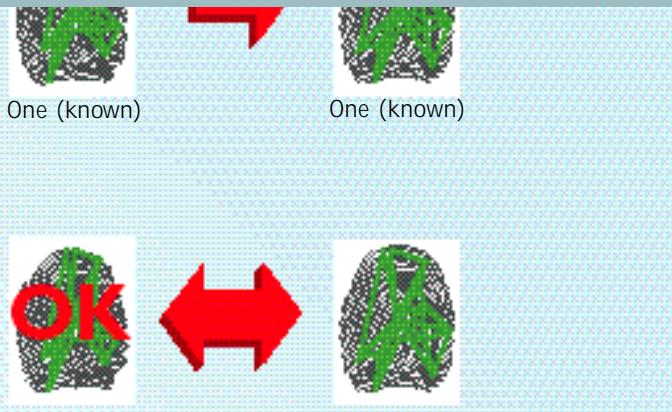
A special-purpose scanner creates a digital image of a fingerprint by shining a light on the fingertip. More light reflects from a fingerprint's ridges—raised areas—than from the valleys that separate them.

**2** Fingerprint Identification (FID) software analyzes the ridges by searching for two types of specific features. One is the core, or center of the print. The other are minutia—points at which ridges end or divide.

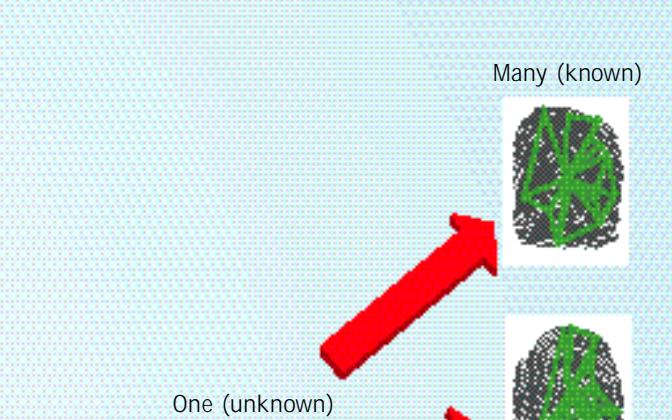
**3** The FID calculates the distances and angles among the minutia. Even if a finger is off-center or rotated during the scan, the relationships among the minutia, shown here in green, do not change vastly and are complex enough to be unique. Ridges and minutia are used to perform one of two possible identifications, one-to-one or one-to-many.



- 4** A one-to-one search occurs in an identity verification system (IVS), which is used for security purposes. The IVS knows in advance who you claim to be, and compares the pattern of minutia of your freshly scanned fingerprint only with a known record of the fingerprint stored in the computer's database.



- 5** Because not all minutia may be captured due to a cut or other aberration, the best the software can do is figure the probability that the print matches some other fingerprint. When a known print matches within an acceptable margin, the software permits access to computer, locked door, or whatever it's guarding.

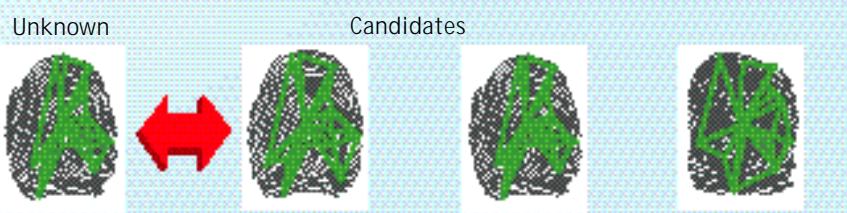


- 6** The best known example of a one-to-many search is performed by the AFIS (Automated Fingerprint Identification System) used by law enforcement. Also called a cold search, it is most commonly used to compare a fingerprint from an unknown person, such as one left at a crime scene, against a database of known persons and fingerprints to determine whether the person is already in the database.



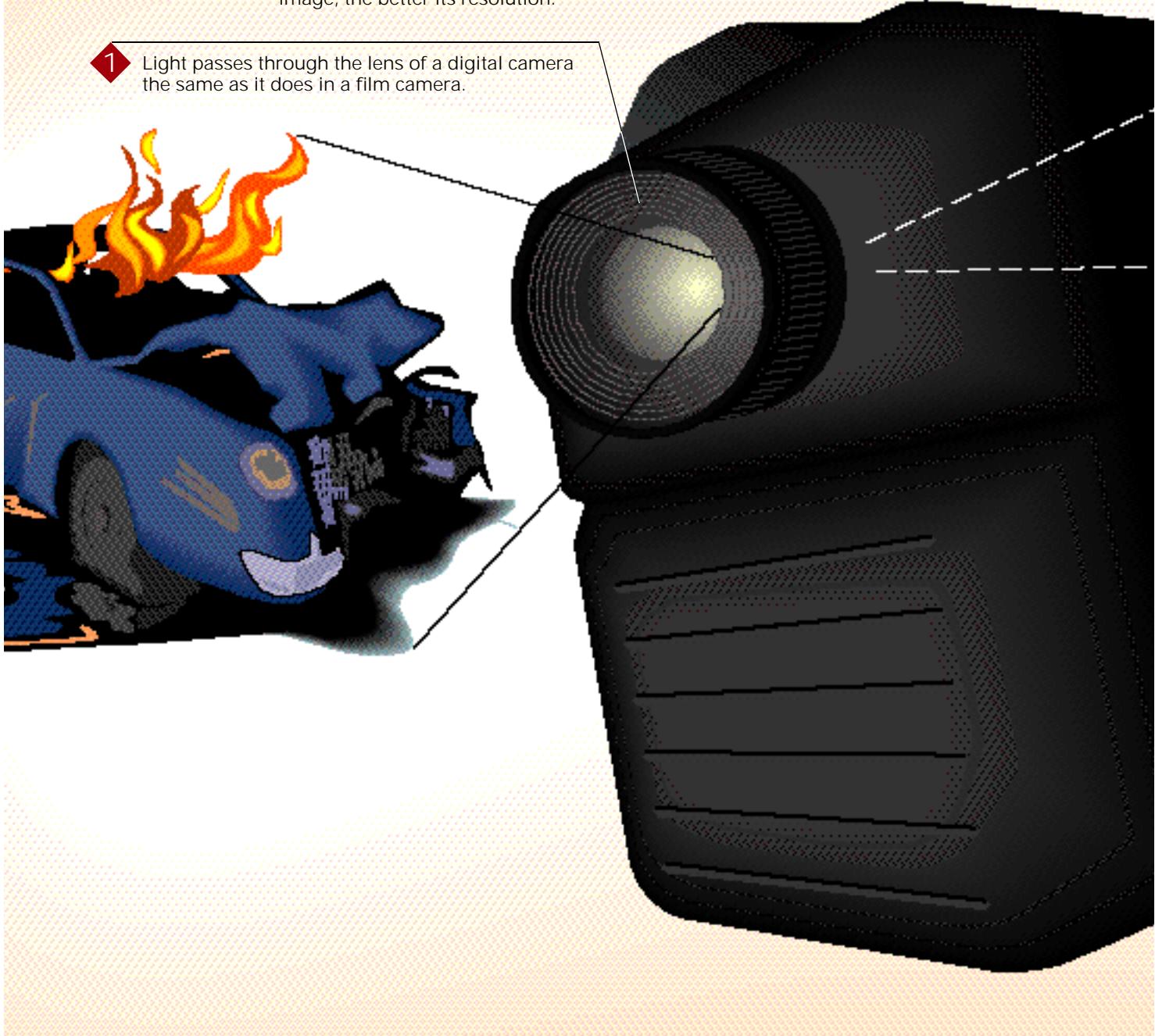
- 7** To narrow the number of prints the unknown must be compared against, the FID first counts the ridges in one direction from the core to the rim of the print. Although counting ridges is quick, it is inherently inaccurate. The number of ridges on the same finger can easily change from print to print, depending on how hard the finger is pressed against the scanner. Also, the range of ridges on all fingers is not that varied—usually yielding a count between 10 and 20. But ridge counting eliminates obvious mismatches.

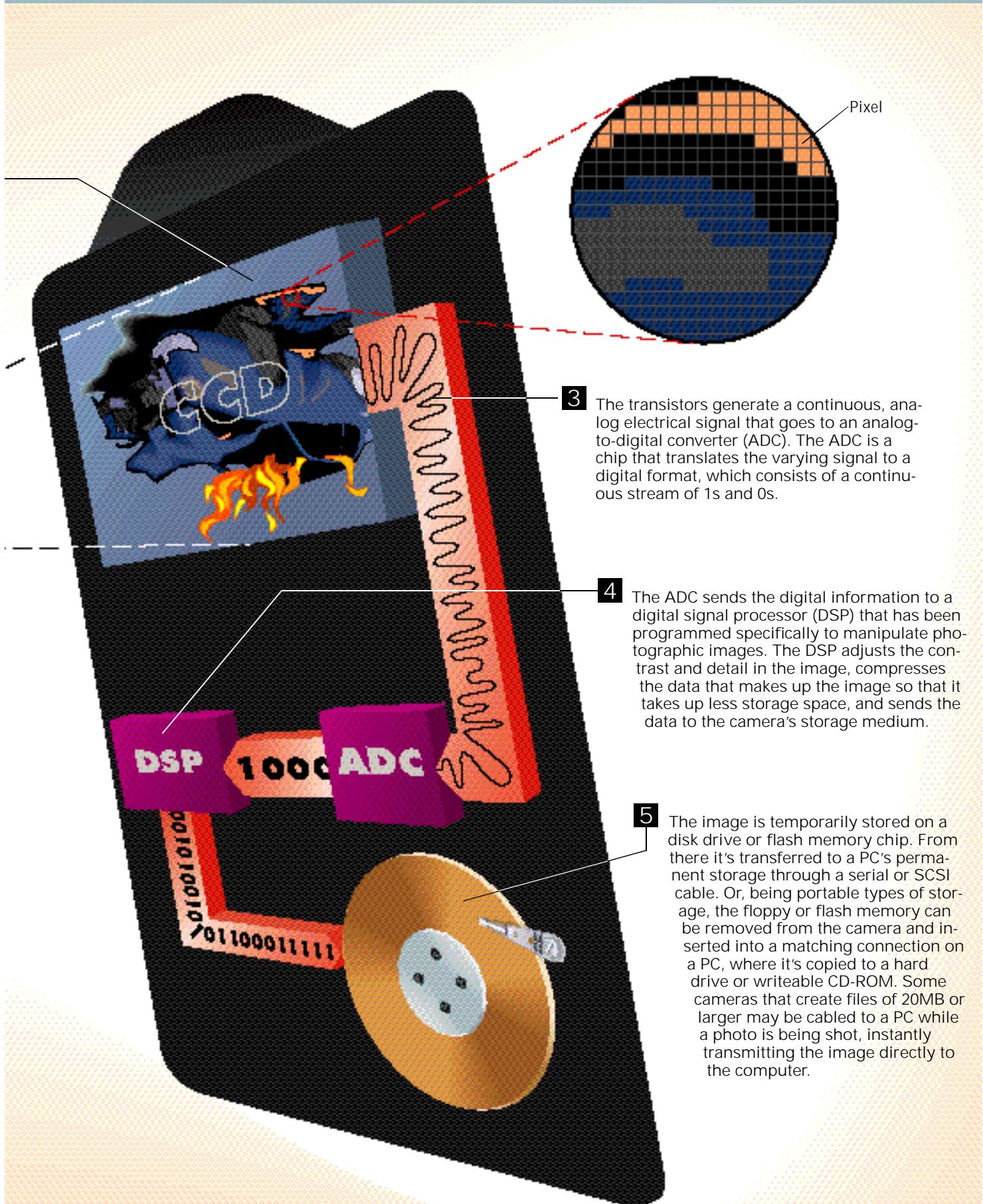
- 8** The FID examines the possible matches generated by ridge counting, looking for further similarities in the pattern of minutia. Because the science is not perfect and the same fingerprint can vary from sample to sample, a one-to-many search often comes up with a list of candidate fingerprints that closely resemble the unknown print. A human makes the final selection.

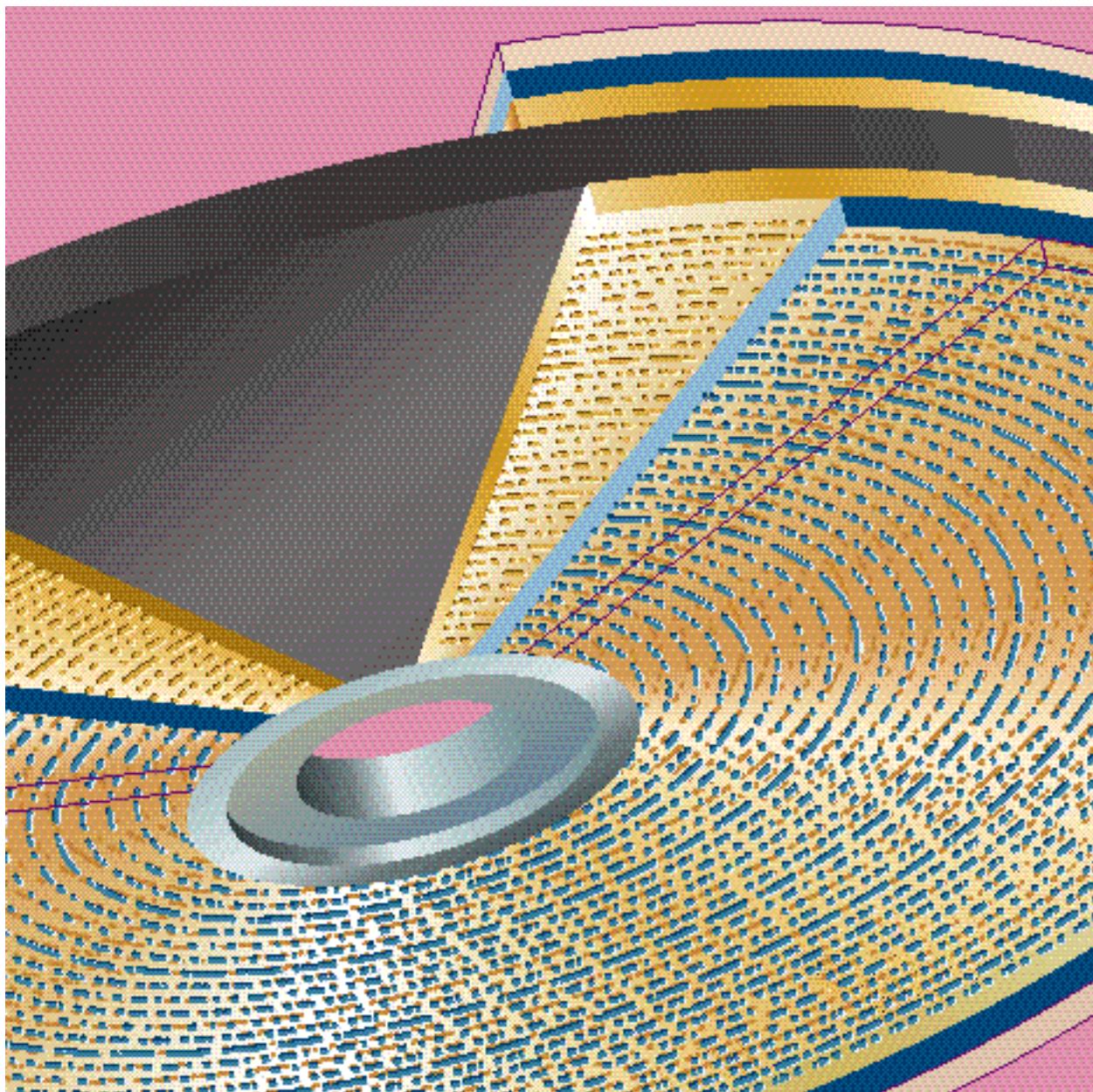


# How a Digital Camera Works

- 2 Instead of being focused on photographic film, the image is focused on a chip called a charge-coupled device (CCD). The face of the CCD is studded with an array of transistors that create electrical currents in proportion to the intensity of the light striking them. The transistors make up the pixels of the image. On a PC's screen or in an input device, such as a scanner or a digital camera, a pixel is the minimum, distinct visual information a component can display or capture. The pixel may be made up of only one transistor for black-and-white photography or several transistors for color. The more pixels in an image, the better its resolution.







P A R T



## MULTIMEDIA

Chapter 28: How CD-ROM and DVD Work  
204

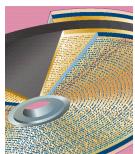
Chapter 29: How Multimedia Sound Works  
216

Chapter 30: How Multimedia Video Works  
224

Chapter 31: How Virtual Reality Works  
228

THE original IBM PC, compared to today's personal computers, was a poor, introverted little thing. It didn't speak, sing, or play the guitar. It didn't even display graphics well or show more than four colors at a time. Not only is today's multimedia revolution changing the way we use PCs, but it is also changing our use of information itself. Where information was formerly defined as columns of numbers or pages of text, we're now communicating with our PCs using our voices, our ears, and our eyes—not simply to read, but also to experience graphics and sound.

What makes a PC a multimedia PC? It's easy to identify multimedia with a CD-ROM drive, which plays computer discs that look like the compact discs played on audio CD players. But a CD-ROM drive in itself does not multimedia make. The first CDs were collections of text—dictionaries, all of Shakespeare's works, adaptations of self-help books—with at best a spare graphic or two. They were not what we think of as multimedia today because they did not have sound or video. In addition to a CD-ROM drive, a multimedia PC must have a sound card, speakers, and the hardware and software needed to display videos and animations. There are, in fact, industry standards for what types of hardware make up an official multimedia PC.



The Multimedia PC standard specifies how fast a CD-ROM must transfer data to the CPU, how much detail is in the sounds played and recorded by the sound board, and the processor power needed to handle sound and video. But today, the MPC standard should be considered only the bare minimum for multimedia. There are faster CD-ROM drives, faster video cards that produce bigger video images and that provide high-resolution, 3-D environments, and sound subsystems that capture and create richer, more realistic sounds. MPEG video provides full-screen, high-resolution video and animation. DVD drives provide the storage space for previously unheard of sound and video quality.

Multimedia has also become an integral part of the Internet. It's hard to land on a Web page that does not have an animated icon or banner. Music and video are becoming more common as Net transmission rates are ramped up by new 56K modems and ISDN. And there are products that incorporate both television and Internet computing so seamlessly that it's hard to say if they are first TVs or computers. The answer is that they're a new breed of home communications, albeit one that's still in its infancy.

But don't get the wrong idea. Multimedia is not simply computerized TV or television. A hard drive, CD-ROM, or DVD is a random access device, the same as memory. They can all access any one piece of data they contain as easily as they can any other data. By contrast, a videotape, movie on film, and tape recording are sequential access devices. You can't get to data at the end of that sequential information—whether it's the airport scene in Casablanca or the last track of a Pink Floyd cassette tape—unless you first go past every other piece of information embedded on the tape or film. You can fast-forward, but there's still a big difference in the time it takes to access the first piece of information and the last. But with a movie on CD-ROM or DVD, such as It's a Wonderful Life or A Hard Day's Night, you can skip directly to any scene you like. Plus a location on the video can be linked to the script, out-takes, or alternative endings. That's the real advantage to multimedia CDs—sound and visuals combined with text and easy access to all of it.

## KEY CONCEPTS

**3-D graphics** Not the same as 3-D movies, in which you have a sense of depth. Rather, computer animation, rendered in real time, in which you can infinitely change the viewpoint.

**AVI** Abbreviation for audio/video interleave, one of the most common file formats that combines video and sound.

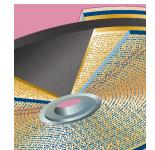
**CD-recordable, CD-R** An optical drive that can record music and computer data that, once written to the disk, cannot be erased or changed.

**DVD** Abbreviation for digital versatile disk, previously digital video disk. A form of optical storage that uses two layers on each side of a disk to store video and other data.

**land** Those areas on the surface of an optical disk that are smooth and reflect a laser beam directly back to a light-sensing diode. See pit.

**laser** An acronym for light amplification by stimulated emission of radiation, a laser is a device that produces a coherent beam of light. That is, the beam contains one or more extremely pure colors and remains parallel for long distances instead of spreading as light normally does.

**MIDI** Acronym for musical instrument digital interface, MIDI is protocol for recording and playing back music on digital synthesizers supported by most sound cards. Rather than representing musical sound directly, it contains information about how music is produced. The resulting sound waves are generated from those already stored in a wave table in the receiving instrument or sound card.



**MPC** Acronym for multimedia personal computer and the official definition for a PC that meets the minimum multimedia standards set by the MPC standards group. MPC3 is the current standard for CD-ROM, sound card, and other components. Most PCs today exceed the MPC standard.

**MPEG** A term derived from the Motion Pictures Expert Group, MPEG is a specific method of decompressing video and sound for real-time playback.

**optical drive** A storage device that uses a laser to read and write data.

**pit** A distorted area on the surface of an optical disk. A pit disperses light from a laser beam aimed at the disk so that the drive's read head does not pick up the beam.

**rewriteable** Applied to both compact discs (CDs) and DVD, the term refers to an optical drive or disk on which data can be changed or erased after it is recorded.

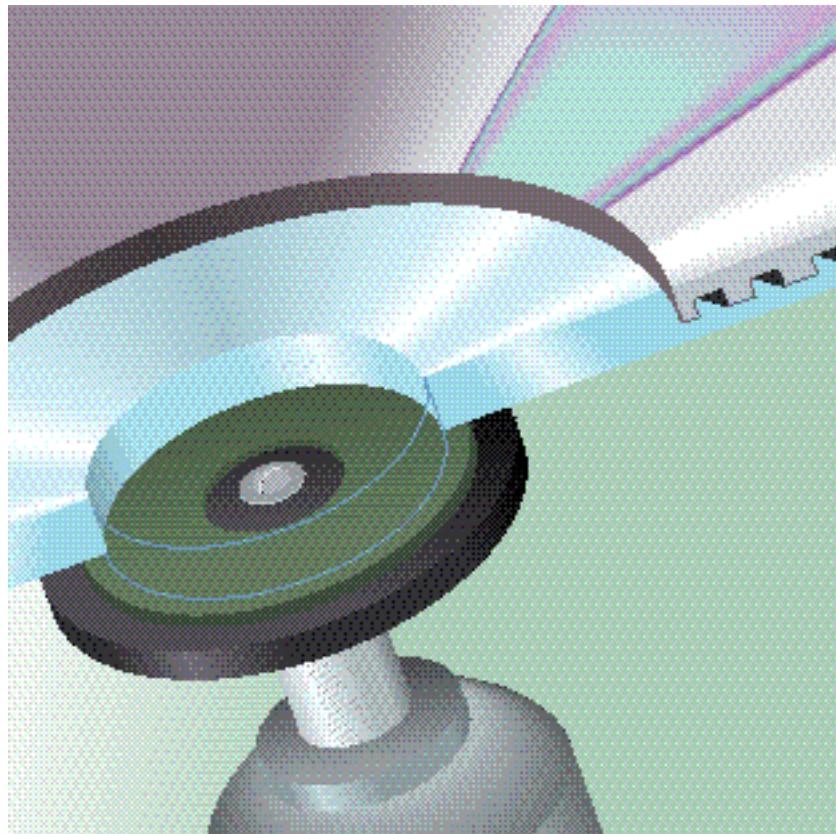
**virtual reality** The simulation of a real or imagined environment that can be experienced visually in the three dimensions of width, height, and depth and may include other sensory experiences including sound, touch, and feedback from "touched" objects or other forces.

**Wave table synthesis** FM synthesis generates sound from digital samples of various musical instruments. FM synthesis works with mathematical descriptions of the sounds rather than the actual recordings. Wave table synthesis is better.

CHAPTER

# 28

# How CD-ROM and DVD Work



A computer's CD-ROM and DVD drives use small, interchangeable, plastic-encased discs from which data is retrieved using a laser beam, much like compact music discs. And like a music CD, computer CD-ROMs and DVDs store vast amounts of information. This is achieved by using light to record data in a form that's more tightly packed than the relatively clumsy magnetic read/write heads that a conventional drive must manage. And like music CD players, computer CD-ROM drives are appearing in jukebox configurations that automatically change among 6 to 100 CDs as you request different information.

Unlike an audio CD player, however, a CD-ROM drive and a DVD drive installed in a PC is nearly devoid of buttons and LCD readouts, except for a button to load and unload a disc and a light to tell you when the drive is reading a disc. The drive is controlled by software in your PC that sends instructions to controller circuitry that's either a part of the computer's motherboard or on a separate board installed in an expansion slot. Together, the software and circuitry manipulate high-tech components that make conventional drives seem crude in comparison.

The CD-ROMs and DVDs that are most common are, again like music compact discs, read-only. Your PC can't write your own data or files to these discs; your PC can retrieve only the information that was stamped on the CDs at the factory. The huge capacity and read-only nature of most CD-ROMs makes the discs the perfect medium for storing reams of data that doesn't need updating, and for distributing large programs. You can easily find CD-ROMs filled with clip art, photographs, encyclopedias, the complete works of Shakespeare, and entire bookshelves of reference material. Finding similar material on DVD is a tougher job.

Although one DVD can hold the equivalent of 13 CDs, with a few exceptions among computer software, most DVDs are hit movies to which extra scenes, subtitles, and dubbing has been added. For once, the storage capacity far exceeds the needs of most software.

DVD has not yet become a standard PC component as CD-ROM drives have been for a few years. But both are perfect for multimedia systems, which use video and sound files that need the voluminous storage the two technologies supply.

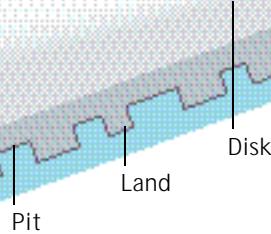
Lately, CD and DVD drives have been getting into the writing business. The price has been falling for CD-R (CD-Recordable) drives that can write data to a special type of compact disc. There is, however, a catch: You can't write to portions of a writeable CD that already have data written to them. You can add to data that was written to the disc in an earlier session, but you can't delete or change what's already there.

Drives for rewriteable CD-ROMs (CD-RW), which overcome the immutability of CD-R, and rewriteable DVDs are quickly dropping in price. One or the other may, at some time in the future, replace the ubiquitous floppy drive. Floppies, CDs, and DVDs are all portable, compact, and cheap. The main difference is storage capacity, and our storage needs long ago outstripped the lowly floppy disk.

# How a CD-ROM Drive Works

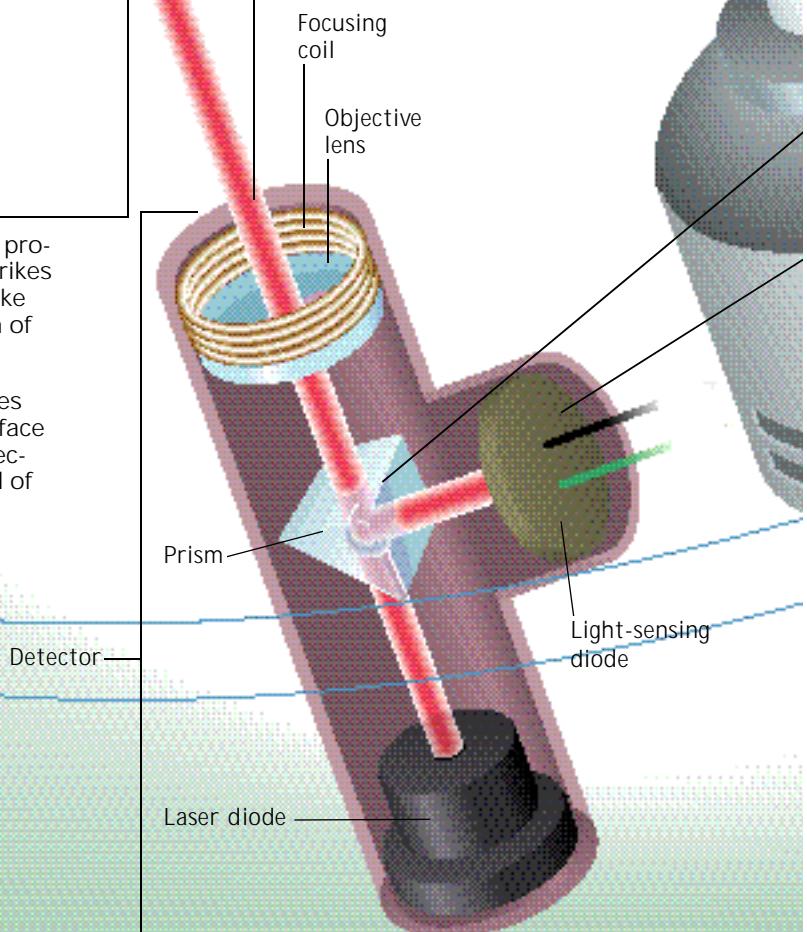
1 A motor constantly varies the rate at which a CD-ROM disc spins so that regardless of where a component, called a detector, is located in relation to the radius of the disc, the portion of the disc immediately above the detector is always moving at the same speed.

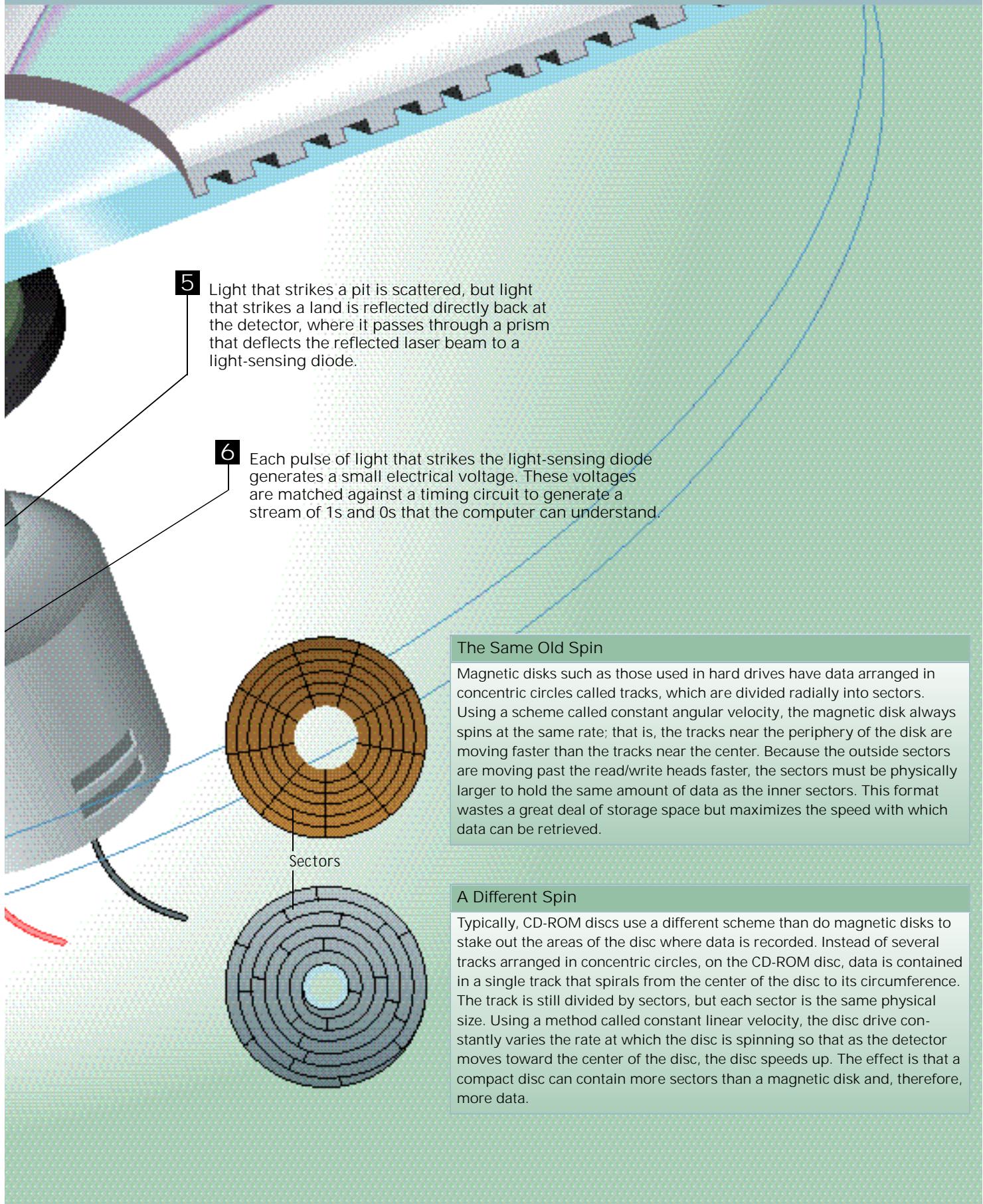
2 The laser projects a concentrated beam of light that is further focused by a lens and a focusing coil.



3 The laser beam penetrates a protective layer of plastic and strikes a reflective layer that looks like aluminum foil on the bottom of the disc.

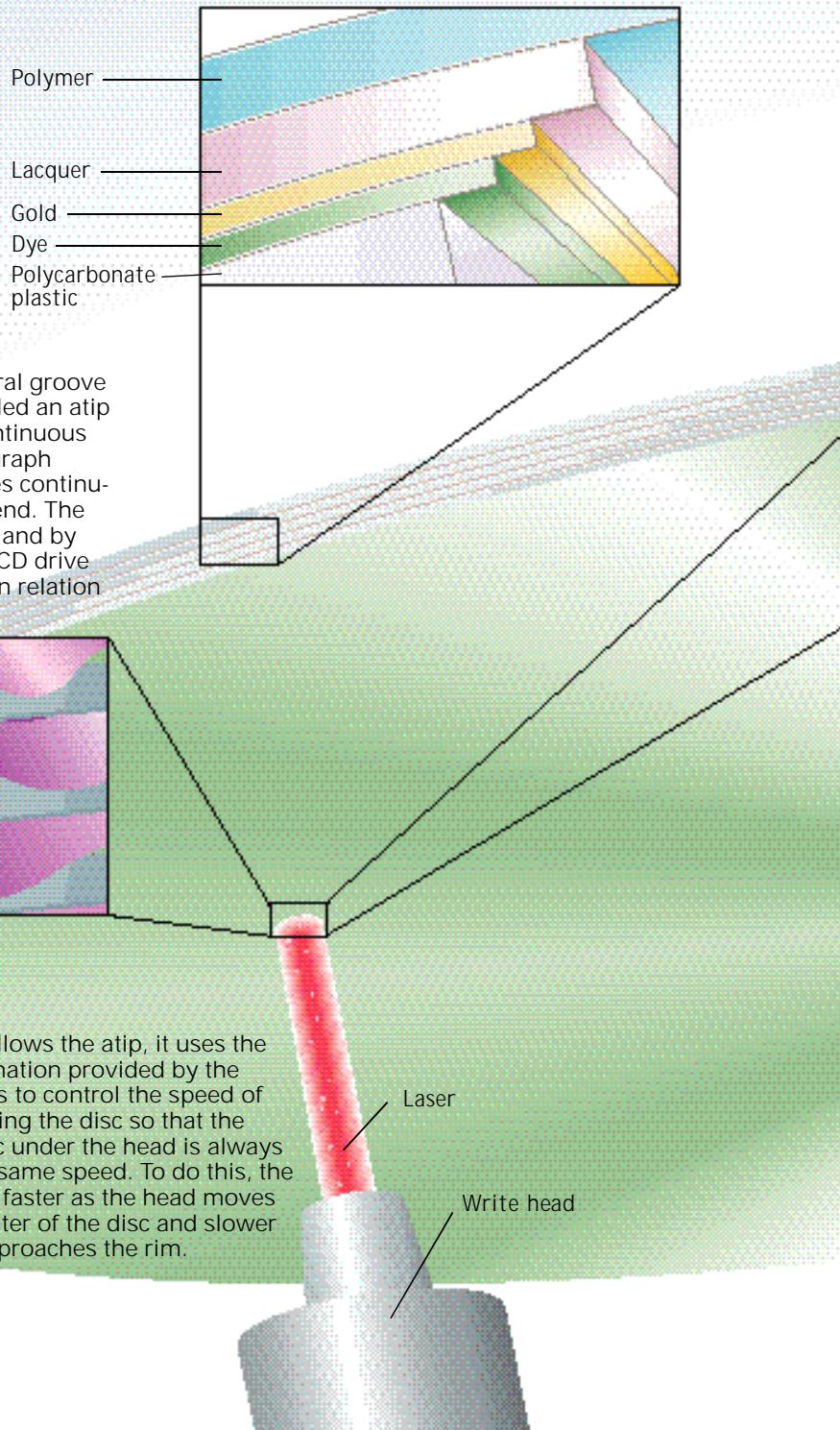
4 The surface of the reflective layer alternates between lands and pits. Lands are flat surface areas; pits are tiny depressions in the reflective layer. These two surfaces are a record of the 1s and 0s used to store data.





# How a Writeable CD-ROM Works

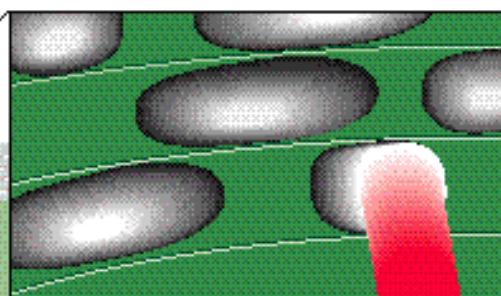
**1** A laser sends a low-energy light beam at a compact disc built on a relatively thick layer of clear polycarbonate plastic. On top of the plastic is a layer of dyed color material that is usually green, a thin layer of gold to reflect the laser beam, a protective layer of lacquer, and often a layer of scratch-resistant polymer material. There may be a paper or silk-screened label on top of all that.



**2** The laser's write head follows a tight spiral groove cut into the plastic layer. The groove, called an atip (absolute timing in pregroove), has a continuous wave pattern similar to that on a phonograph record. The frequency of the waves varies continuously from the start of the groove to its end. The laser beam reflects off the wave pattern, and by reading the frequency of the waves, the CD drive can calculate where the head is located in relation to the surface of the disc.

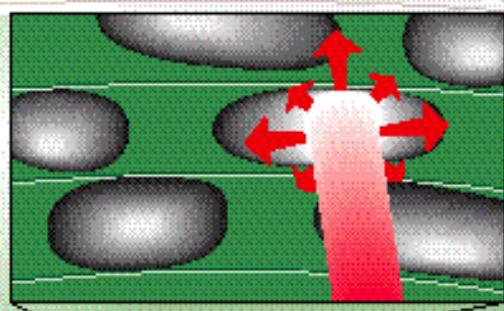
**3** As the head follows the atip, it uses the position information provided by the groove's waves to control the speed of the motor turning the disc so that the area of the disc under the head is always moving at the same speed. To do this, the disc must spin faster as the head moves toward the center of the disc and slower as the head approaches the rim.

- 5** The dye layer is designed to absorb light at that specific frequency. Absorbing the energy from the laser beam creates a mark in one of three ways, depending on the design of the disc. The dye may be bleached; the polycarbonate layer may be distorted; or the dye layer may form a bubble. Regardless of how the mark is created, the result is a distortion, called a stripe, along the spiral track. When the beam is switched off, no mark appears. The lengths of the stripes vary, as do the unmarked spaces among them. The CD drive uses the varying lengths to write the information in a special code that compresses the data and checks for errors. The change in the dye is permanent, making the recordable compact disc a write-once, read-many (WORM) medium.

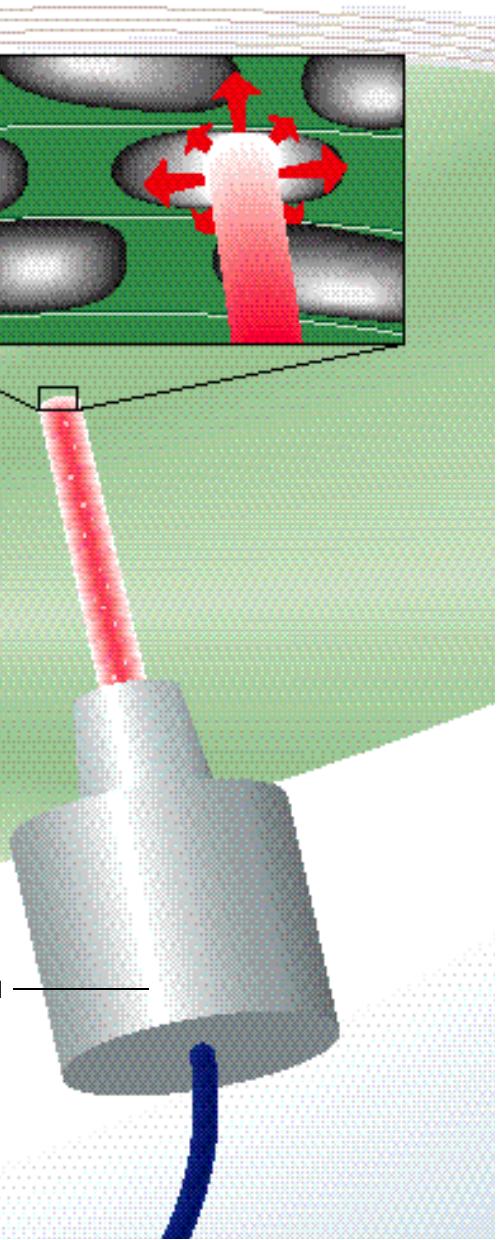
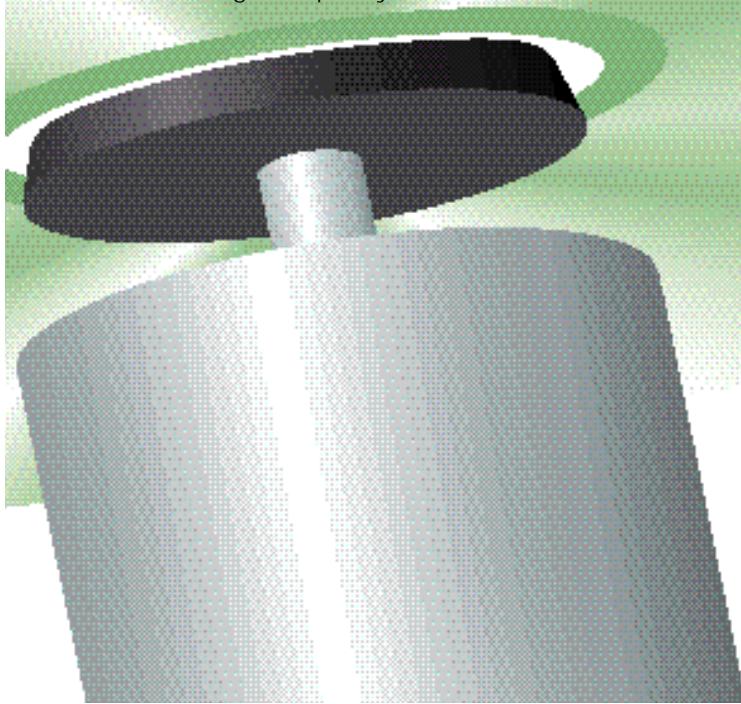


- 4** The software used to make a compact disc recording sends the data to be stored to the CD in a specific format, such as ISO 9096, which automatically corrects errors and creates a table of contents. The table is needed because there is nothing like a hard drive's file allocation table to keep track of a file's location. The CD drive records the information by sending a higher-powered pulse of the laser beam at a light frequency of 780 nanometers.

- 6** The CD-Recordable drive—or an ordinary read-only CD drive—focuses a lower-power laser beam onto the disc to read data. Where a mark has not been made on the surface of the disc, the gold layer reflects the beam straight back to the read head. When the beam hits a stripe, the distortion in the groove scatters the beam so that the light is not returned to the read head. The results are the same as if the beam were aimed at the lands and pits in an ordinary CD-ROM. Each time the beam is reflected to the head, the head generates a pulse of electricity. From the pattern in the pulses of current, the drive decompresses the data, error-checks it, and passes it along to the PC in the digital form of 0s and 1s.



Read head —

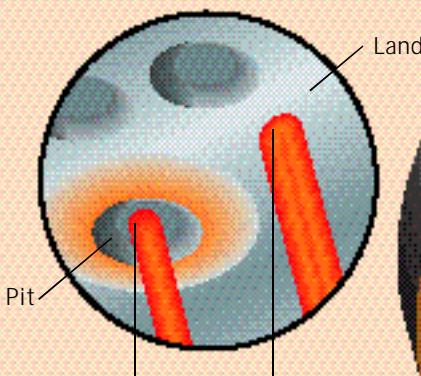


# How a DVD Works

**1** A one-sided digital video disc (DVD) is made up of four main layers. First there's a thick polycarbonate plastic that provides a foundation for the other layers. Next, a much thinner layer of opaque, reflective material is laid on the base. Then comes a thin layer of transparent film, and finally a surface layer of clear, protective plastic. Data, including video, audio, text, or programs, is represented, as with a CD-ROM, by a combination of flat areas (lands) and indentations (pits) on two of the surfaces—the transparent film and the shiny opaque layer. The DVD pits, however, are much smaller, which is part of the reason the DVD holds as much as 8.5MB of data, the capacity of 13 compact discs.

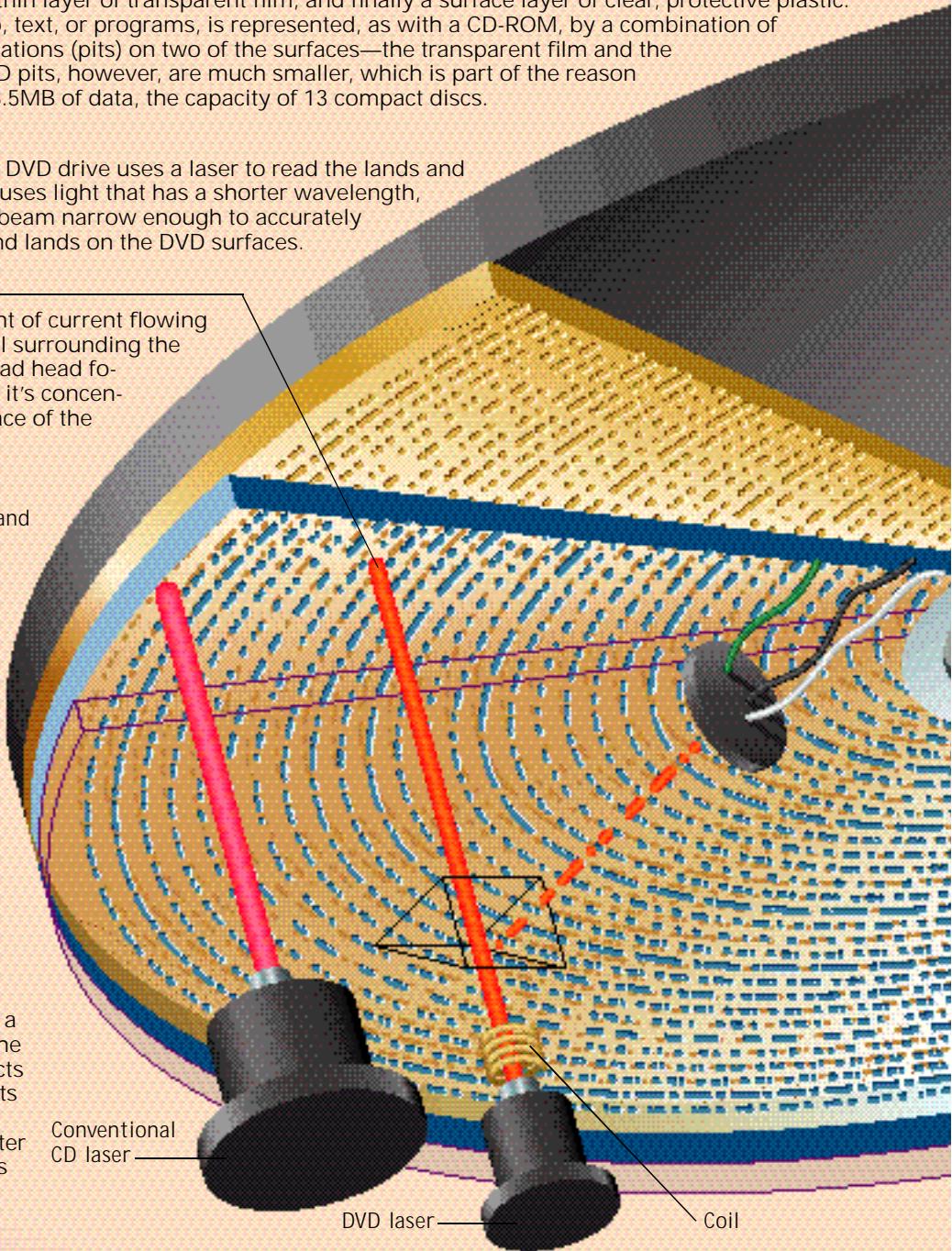
**2** Like a CD-ROM drive, a DVD drive uses a laser to read the lands and pits. But the DVD laser uses light that has a shorter wavelength, which makes the laser beam narrow enough to accurately read the smaller pits and lands on the DVD surfaces.

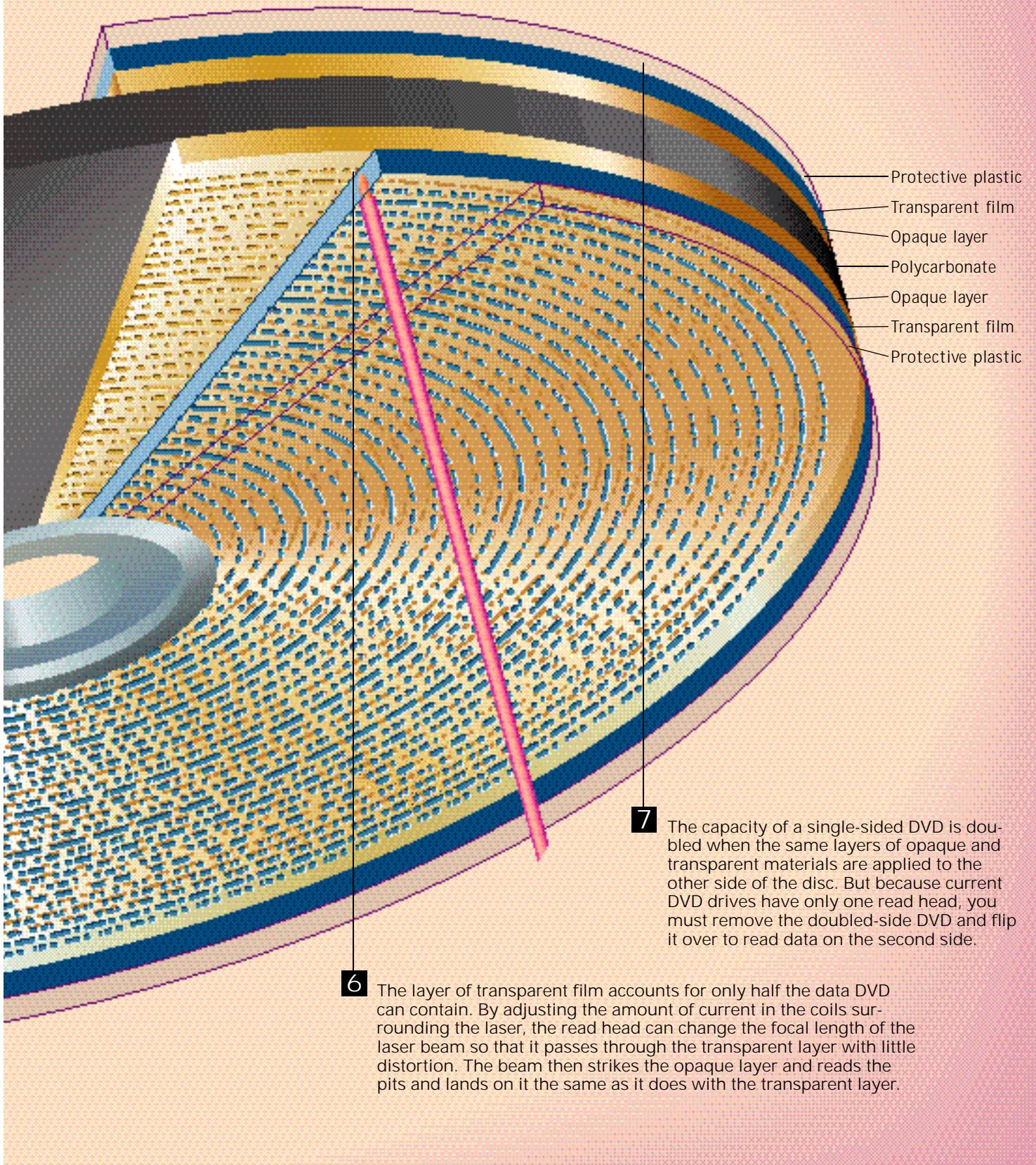
**3** By changing the amount of current flowing through a magnetic coil surrounding the laser beam, the DVD read head focuses the beam so that it's concentrated on only the surface of the transparent film.



**4** Where the laser beam hits a pit, the indentation scatters the light in all directions.

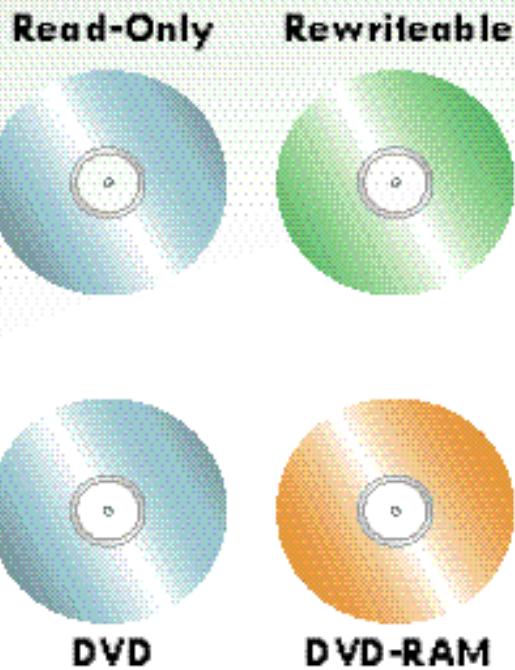
**5** But when the beam of light hits a flat area, it is reflected back to the read head, where a prism deflects the light to a device that converts the bursts of light energy into bursts of electricity. The computer interprets those electrical pulses as code and data.



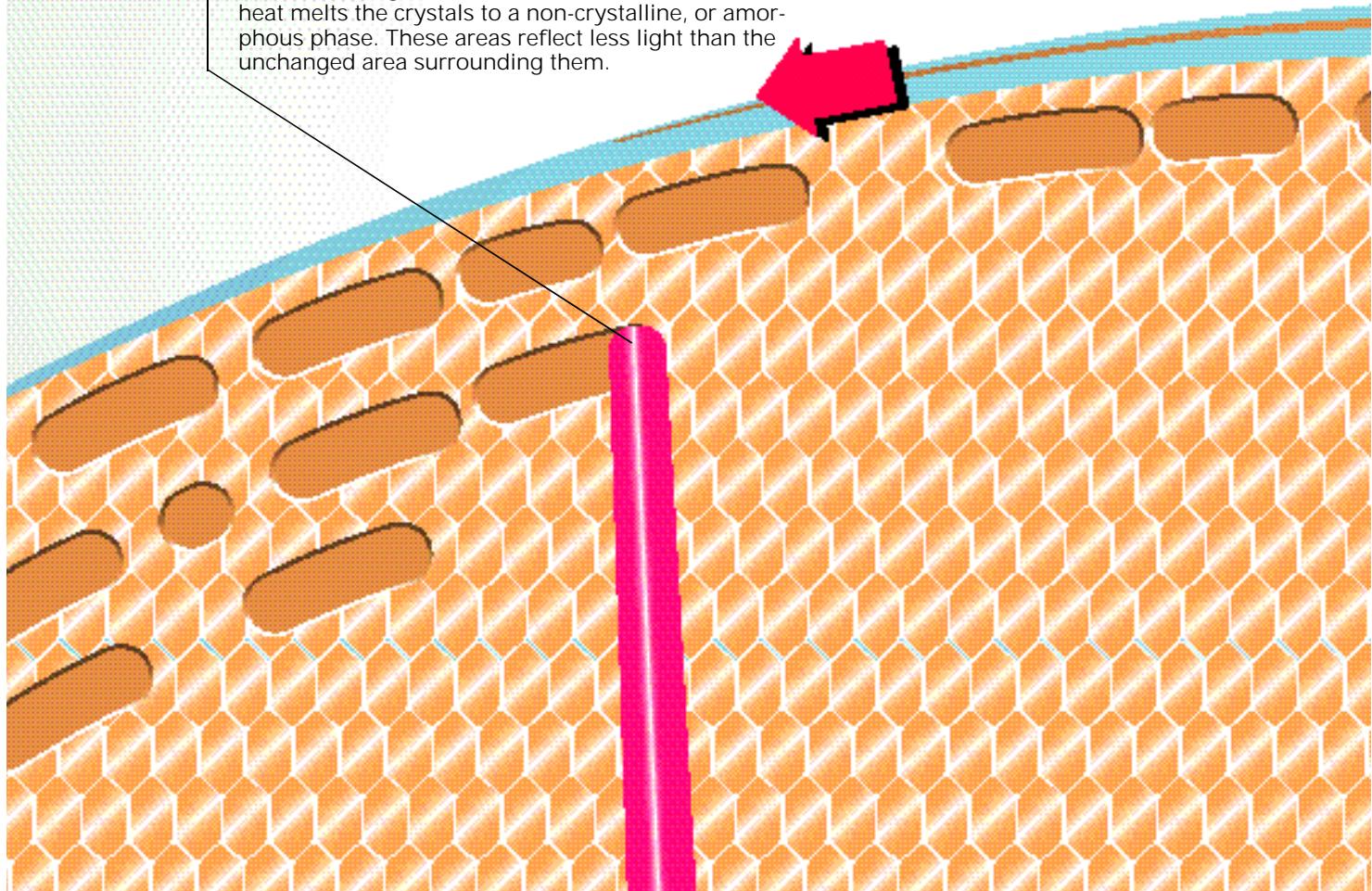


# How CD-RW and DVD-RAM Work

- 1** CD-ROMs and DVD discs—both of which are read-only, optical media—have their rewriteable equivalents. Rewriteable means that not only can you put files on a disc, you can erase and change them, just as you can with a hard drive. CD-ROM has a rewriteable equivalent in CD-Rewriteable (CD-RW). DVD has DVD-RAM. Although CD-RW and DVD-RAM differ in how much data can be written to them, both use a process called phase change technology to write, change, and erase data.
- 2** The process writes data to both types of disc by focusing a high-intensity laser beam on a layer typically made of silver, indium, antimony, and tellurium embedded in the disc's plastic base. In its original state, this layer has a rigid, polycrystalline structure.



- 3** The laser beam selectively heats areas to 900–1,300°F degrees. Where the beam strikes, the heat melts the crystals to a non-crystalline, or amorphous phase. These areas reflect less light than the unchanged area surrounding them.



## Infrared Laser

To read and write data through phase change recording, a CD-Rewriteable laser uses the same light wavelength found in a CD-ROM. The laser beam is made of infrared light, 780 nanometers wide. The wavelength determines how small pits and lands can be and how tightly the spiral recording groove is wound. A CD-RW disc stores up to 650MB of data. It writes data at a rate of 300K a second and reads data at a rate of 450K a second.



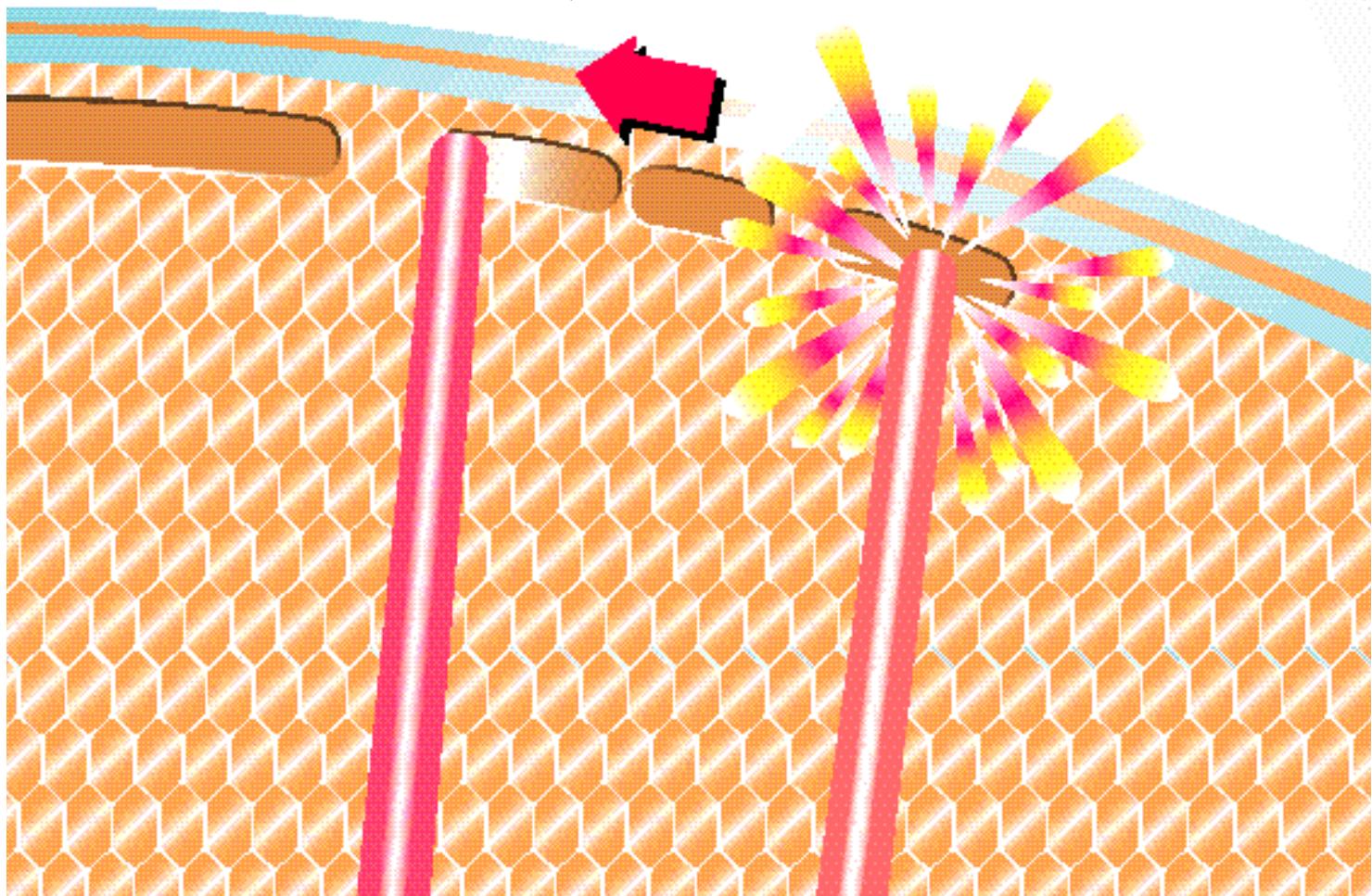
## Red Laser

A DVD-RAM drive uses a red laser to read and write data. Because the light has a wavelength of 635-650 nanometers, shorter than infrared light, the red laser creates pits and lands that are smaller and the spiral recording groove is packed tighter. More precise engineering and optics also contributes smaller pits and lands to the DVD's greater storage capacity of 2.6GB on each side. The drive has a 1.3MB per second write time and a 1.2-2.7MB per second read time.



- 4** Later, when a weaker laser beam, used only to read data from the discs, strikes the non-crystalline area, the beam is scattered and not picked up by the light-sensitive diode in the read head. With their lower reflectance, these areas become pits; representing 1s. Areas that are not heated are the more reflective lands, representing 0s. When the read laser beam strikes the lands it is reflected directly to the diode, creating an electrical current that is sent to the computer. The computer interprets the pattern of electrical pulses, decompresses the data they represent, checks the data for error, and sends the data to the software using the drive. (See "How a CD-ROM Drive Works" for more details.)

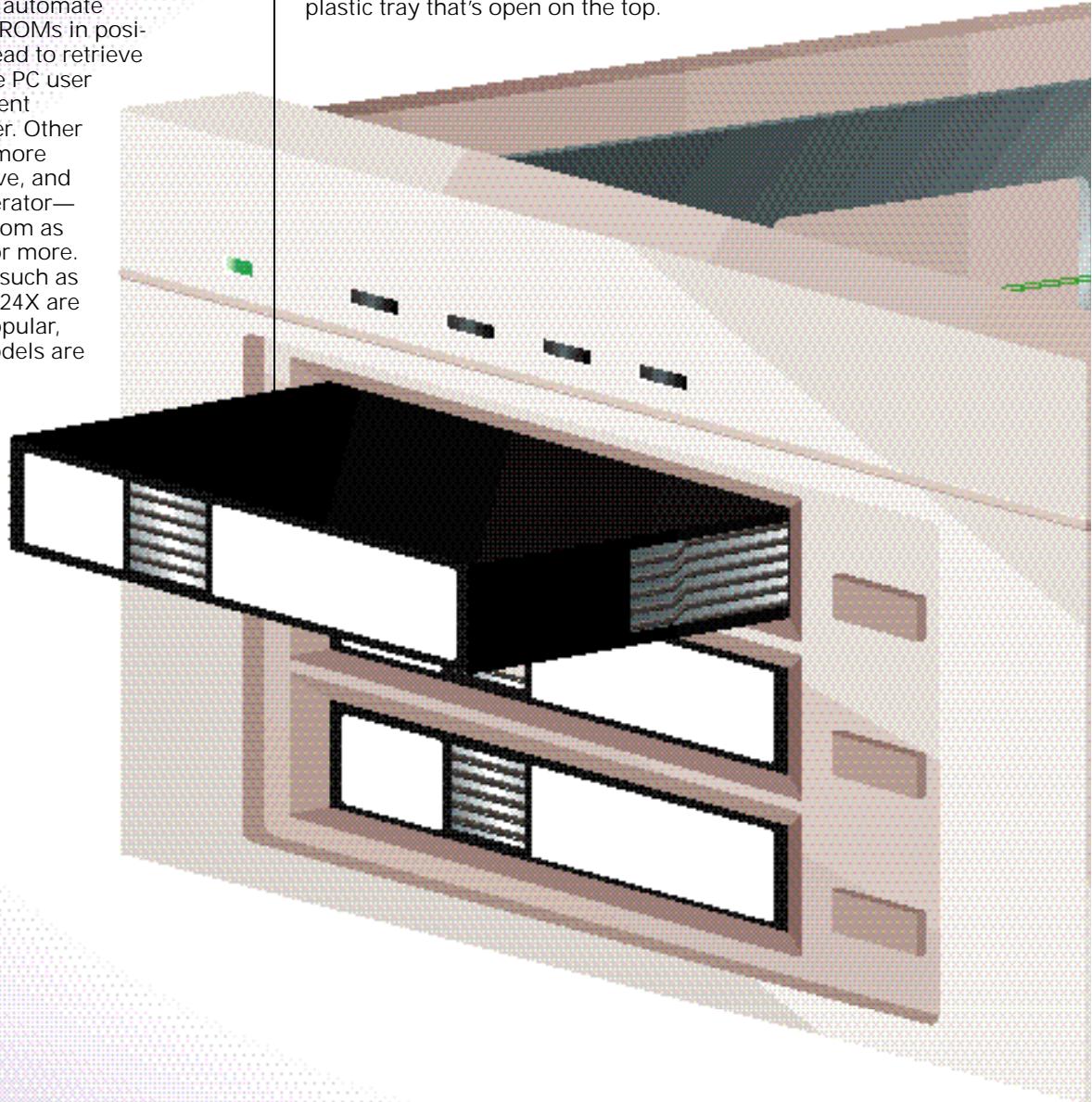
- 5** To erase data or to change a pit back to a land area—a process called the annealing phase—the CD-RW and DVD-RAM drives use a lower-energy beam to heat pitted areas to 400F degrees. This amount of heat is below the melting point, but it still loosens up the phase change media so that it can recrystallize to its original state.



# How a CD Jukebox Works

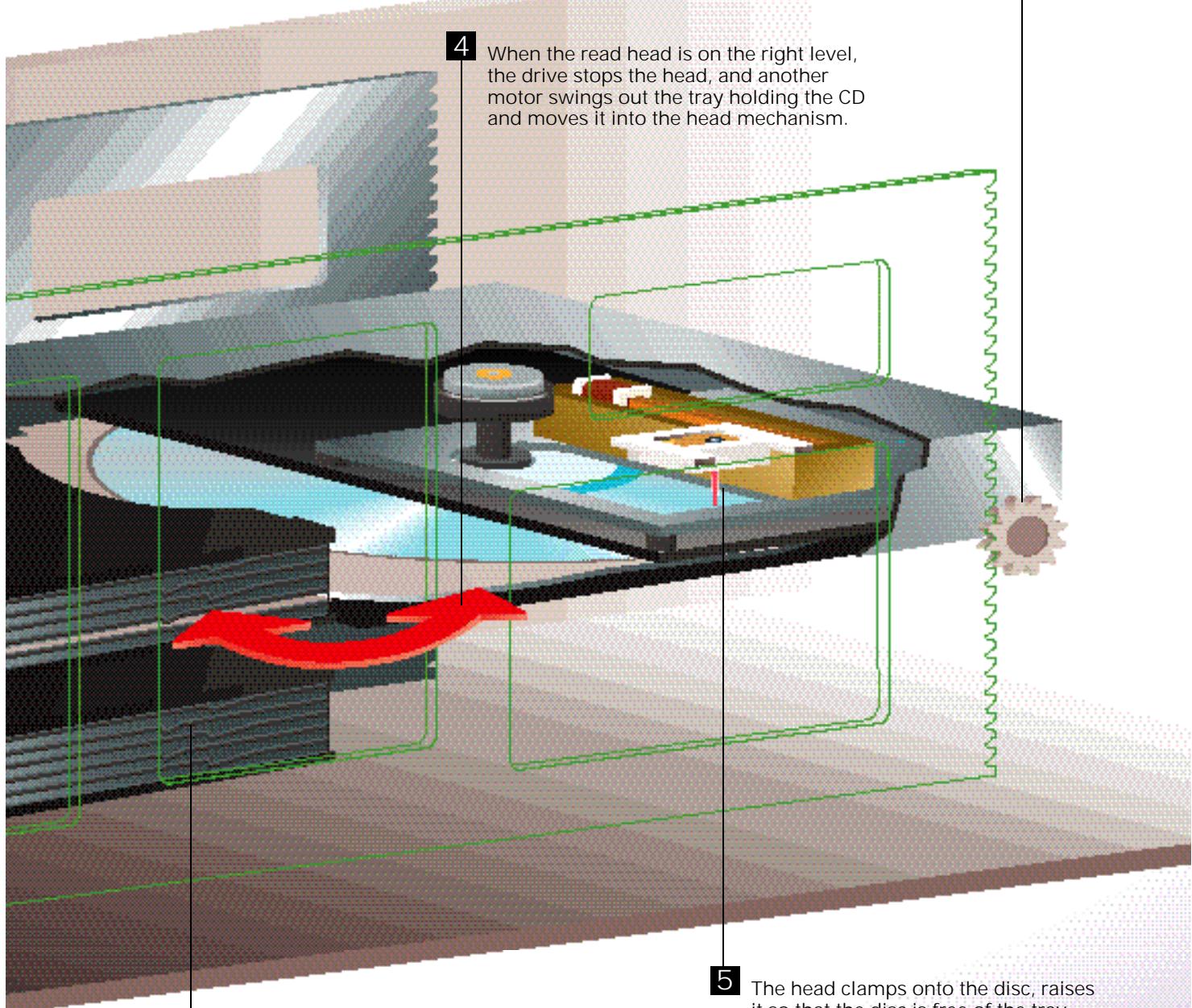
1 A CD drive, such as the Pioneer model shown here, uses a jukebox mechanism to automate putting 6 to 18 CD-ROMs in position for the read head to retrieve their data when the PC user changes to a different CD-ROM drive letter. Other multidisk drives—more elaborate, expensive, and the size of a refrigerator—can retrieve data from as many as 100 CDs or more. The 6-disc models such as the Pioneer DRM-624X are becoming more popular, and the 18-disc models are being phased out.

2 CDs are stored in cassettes that each hold a half-dozen discs. Each disc rests in a thin plastic tray that's open on the top.



- 3 When the PC sends a signal to the drive to load a CD, gears turn to raise or lower the jukebox's read head mechanism until it's at a height matching that of the CD the computer has asked for.

- 4 When the read head is on the right level, the drive stops the head, and another motor swings out the tray holding the CD and moves it into the head mechanism.

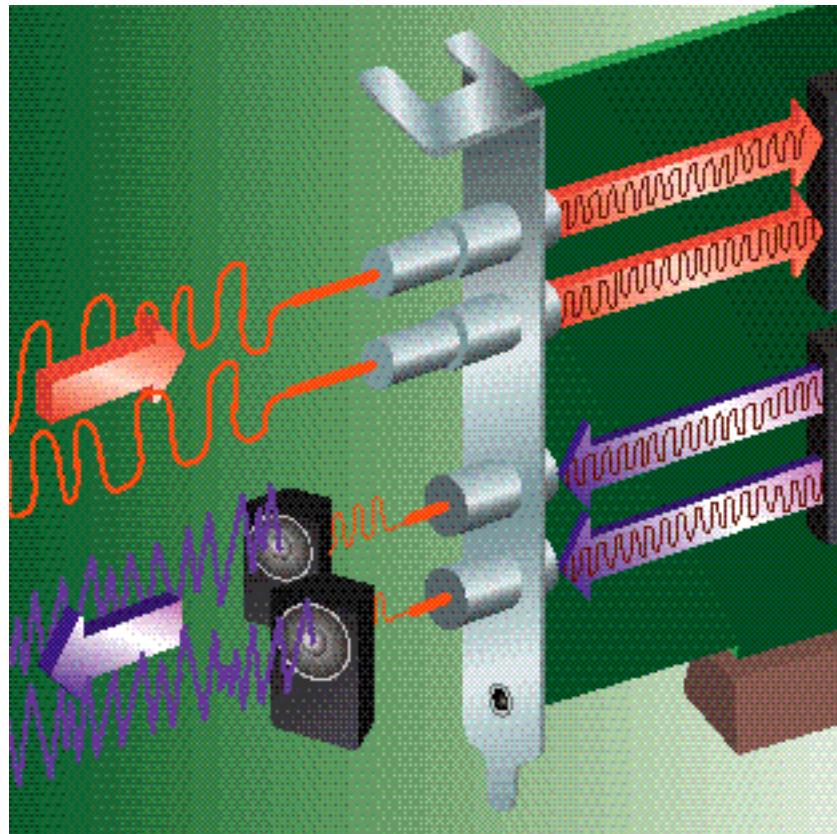


- 6 When the PC requests a different disc, the head mechanism lets go of the disc that's already loaded and places it back into the tray, which returns the disc to the cassette. Then the head mechanism moves to the level of the new CD. It takes about 10 seconds to unload one CD and begin reading from another.

- 5 The head clamps onto the disc, raises it so that the disc is free of the tray, and the head's spindle motor spins the disc. The read head's laser moves along the disc's groove, reading data from the reflections off the disc.

CHAPTER  
**29**

# How Multimedia Sound Works



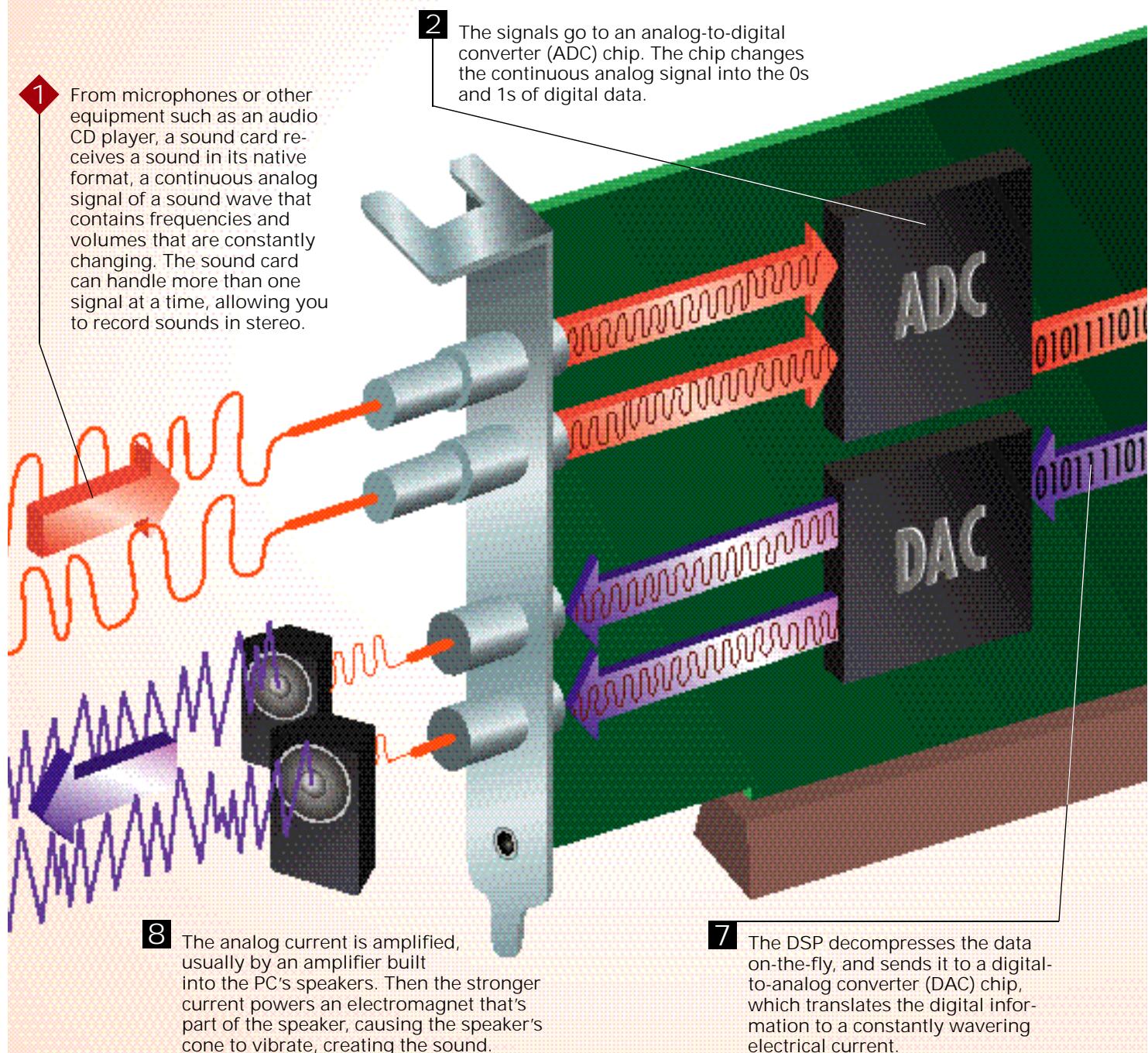
FOR years DOS and Windows personal computers sounded like cartoon roadrunners. They could play loud, high-pitched beeps and low-pitched beeps. But they were still only beeps. There was no hiding that fact.

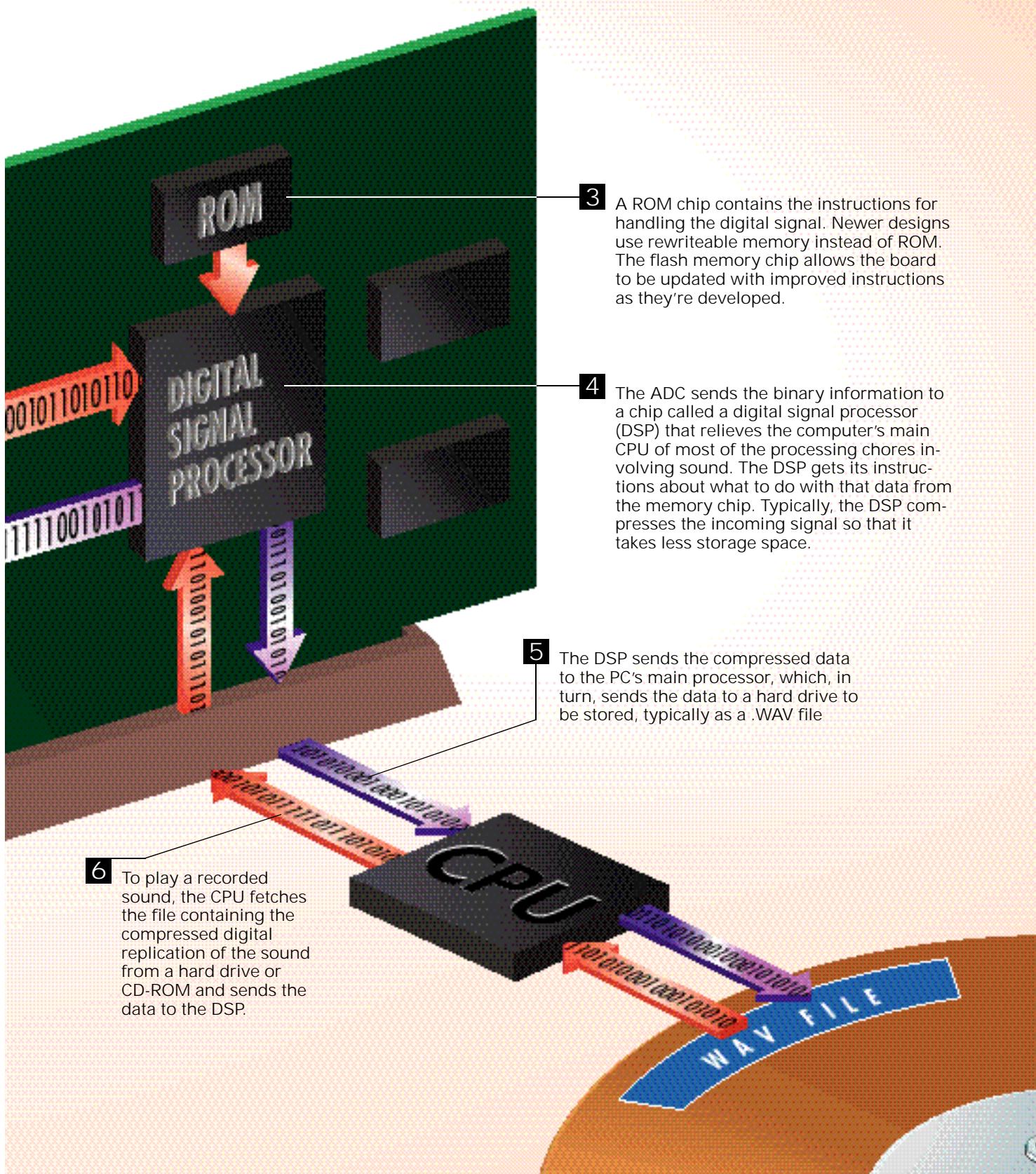
We owe today's multimedia sound abilities to game players. They saw the advantages of hearing realistic explosions, rocket blasts, gun shots, and mood-setting background music long before developers creating business software realized the practical advantages of sound. Now, you can listen to your PC speak instructions as you follow along on the keyboard, dictate a letter by talking into your PC, give your PC spoken commands, attach a voice message to a document, and not have to take your eyes off a hard-copy list while your PC sounds out the numbers as you're typing them into a spreadsheet.

None of the multimedia that enhances business, personal, and family use of a PC could exist without sound capabilities. Multimedia CD-ROMs and DVD bring their subjects to life in ways not possible in books, because you hear the actual sounds of whales, wars, and warblers, of sopranos, space blaster shots, and saxophones. Not that sound capabilities must always enlighten you on a topic. You should have fun with your PC, too. It won't make the day shorter to replace Windows's error chime with Homer Simpson saying, "D'oh!" And you won't be more productive every time a Windows program opens or closes if it makes a sound like those doors in Star Trek. But so what? Taking advantage of the sounds in a multimedia PC personalizes a machine that has a rap for being impersonal. Sound simply adds to the fun of using your computer. And we all spend too much time in front of these things for it not to be fun.

Lately, multimedia sound has taken a reverse spin. Now, instead of us listening to our computers, our computers can listen to us. Although a slow, painstaking version of voice recognition has been possible for years, it's only with the faster processing made possible by the Pentium II processor and MMX technology that natural speed recognition has become possible. We can now dictate, speaking in a normal voice, instead of typing. And although we do so much typing that it seems natural, if you think of it, there's hardly a more unnatural way to communicate than tapping little buttons. Don't throw away the keyboard just yet, but very soon expect to be holding complete conversations with that machine on your desk.

# How Sound Cards Work and Record Sound



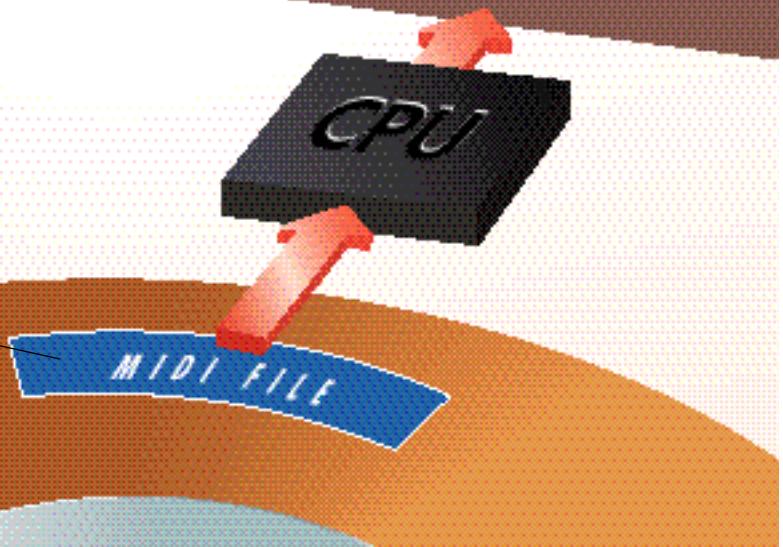
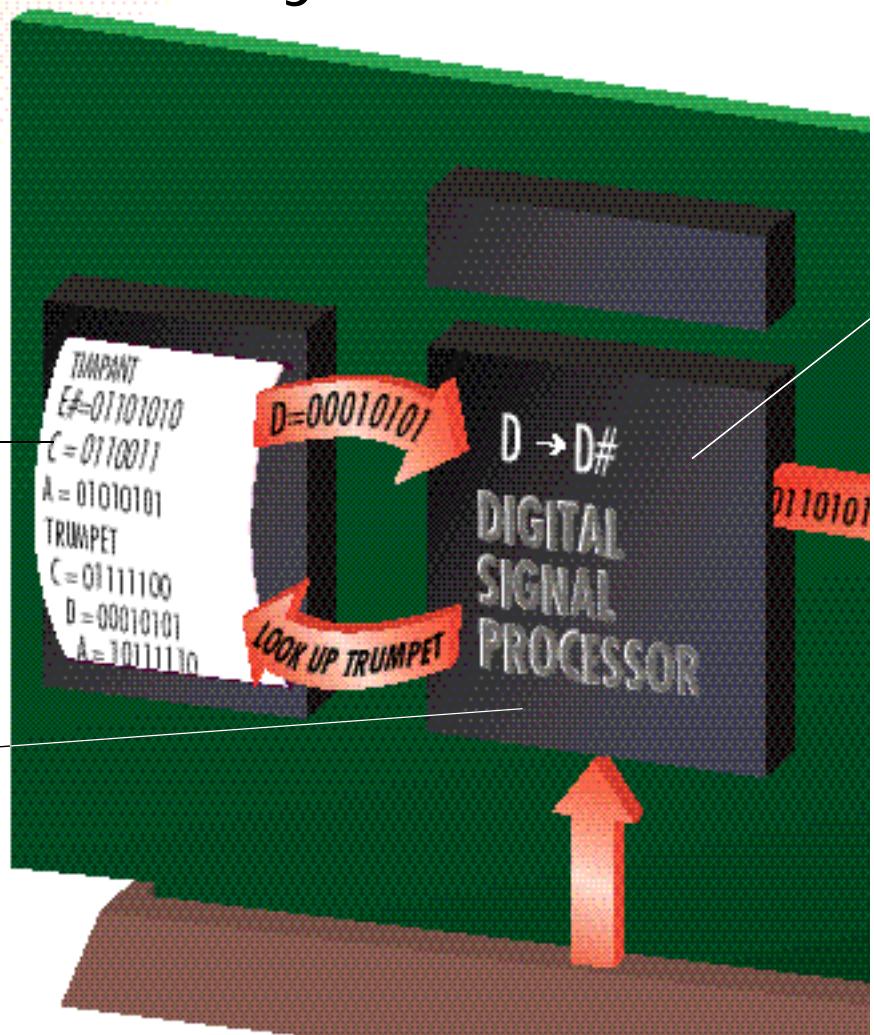


# How MIDI and FM Synthesis Work

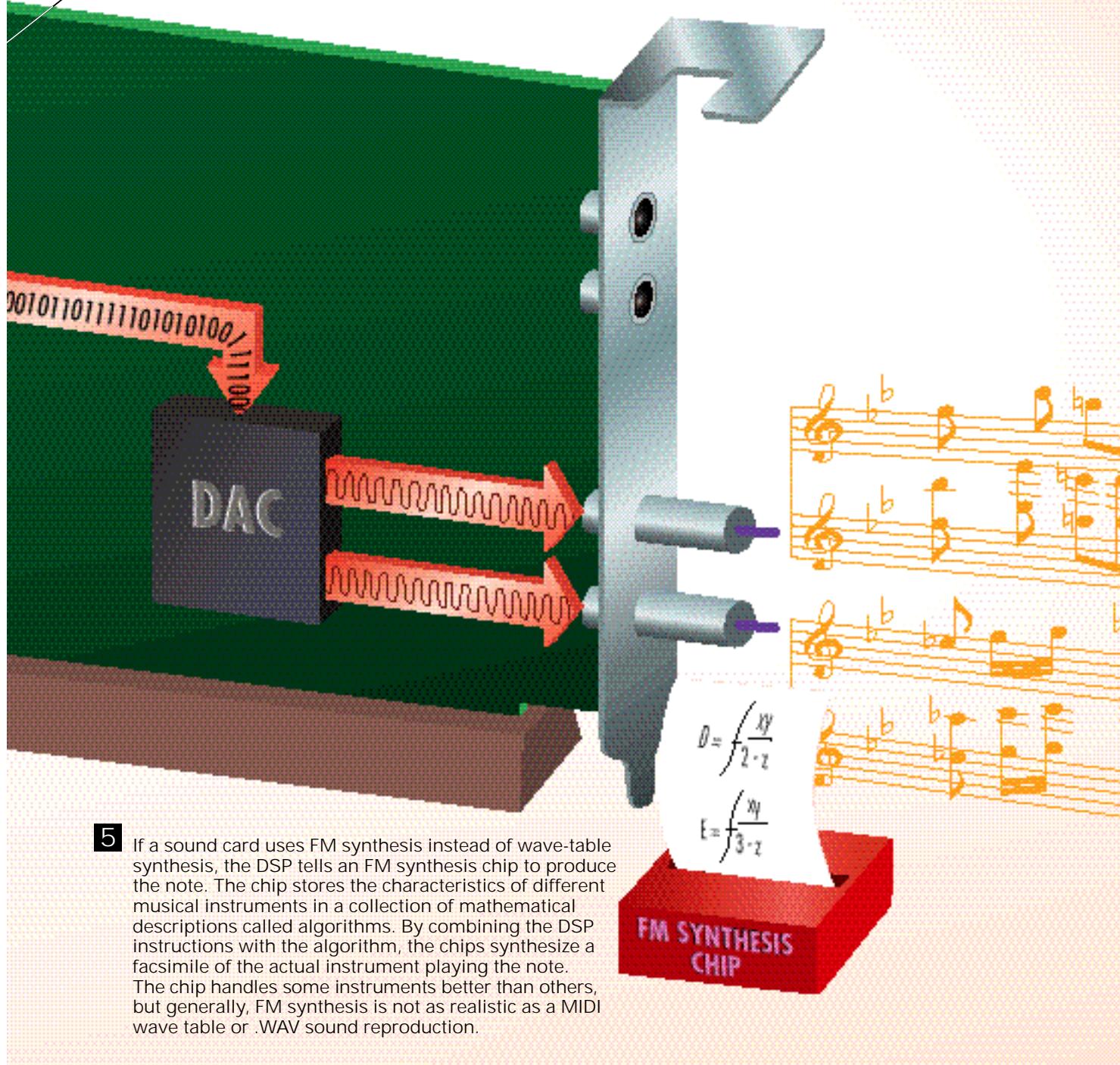
3 If the sound card uses wave-table synthesis to reproduce the sound qualities of musical instruments, samples of the actual sounds made by different musical instruments are stored in a ROM chip.

2 The MIDI instructions tell the digital signal processor (DSP) which instruments to play and how to play them.

1 While some types of sounds are straight-forward recordings, such as those contained in .WAV files, musical instrument digital interface (MIDI) sound was developed to conserve disk space by saving only instructions for how to play music on electronic versions of various musical instruments, not recordings of the actual sounds.



- 4 The DSP looks up the sound in the ROM's table. If the instructions call for, say, a trumpet's D-sharp, but the table has a sample of only a D note for the trumpet, the DSP manipulates the sound sample to raise it to a D-sharp pitch.



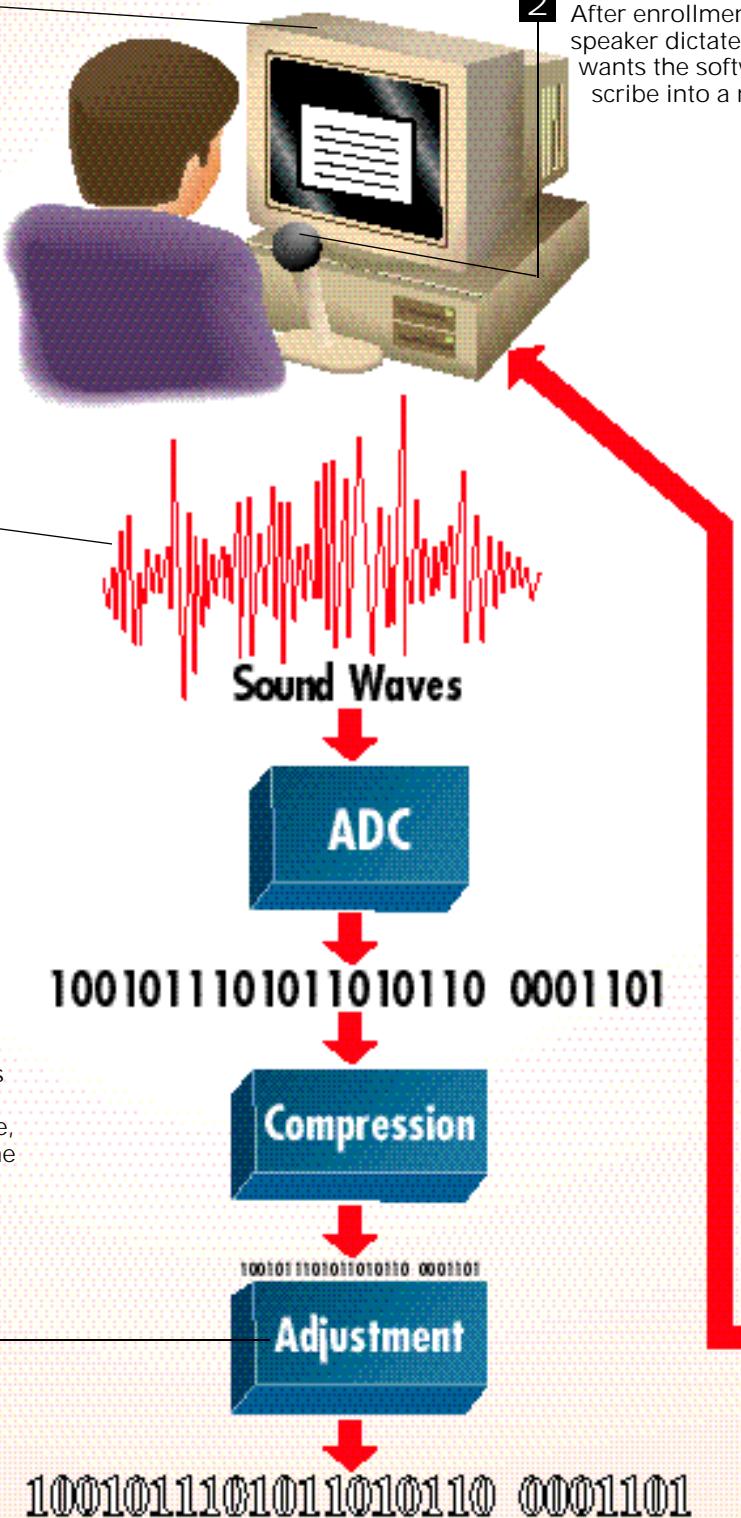
# How Speech Recognition Works

**1** A person who's going to use speech recognition software must first go through an enrollment. This consists of the person dictating text that is already known to the software for 10 minutes to an hour. From this sampling, the software creates a table of vocal references, which are the ways in which the speaker's pronunciation of phonemes varies from models of speech based on a sampling of hundreds to thousands of people. Phonemes are the smallest sound units that combine into words, such as "duh," "aw" and "guh" in "dog." There are 48 phonemes in English.

**3** Once every 10–20 milliseconds, the analog signal generated by the microphone is sampled by an analog-to-digital converter (ADC) that converts the spoken sound to a set of measurements of several factors, including pitch, volume, frequency, length of phonemes, and silences. The measurements are compressed for quicker processing.

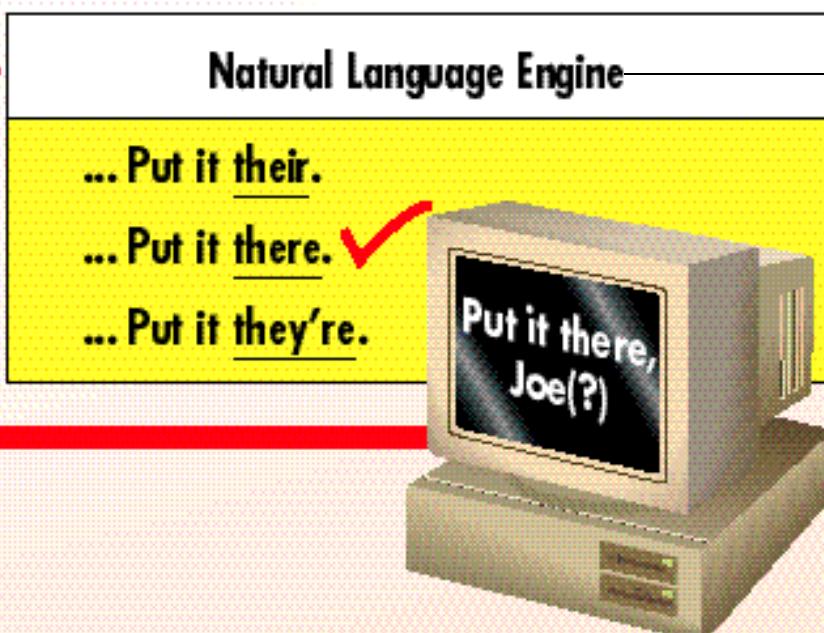
**4** The software's speech engine makes adjustments to the phoneme measurements by factoring in background sounds, the acoustic characteristics of the microphone, and, from the table of vocal references, the speaker's individual idiosyncrasies of accent, regional pronunciations, and voice characteristics.

**2** After enrollment, the speaker dictates the text he wants the software to transcribe into a microphone,



Model Database Material				
	Matches			
	Volume	Pitch	Length	Tremor
i: (ASKmSIX)	✓			
I (as in sit)		✓		
oo (as in ten)	✓		✓	
æ (as in hat)	✓	✓		✓
au (as in home)				
ŋ (as in sing)		✓		

Lexicon	
th e r	Matches
there, their, they're	



5 The acoustic recognizer compares the measurements of the corrected phonemes to a binary-tree database of compressed, known phoneme measurements—models—compiled from sampling the speech of thousands of people. For each measurement, such as pitch, the recognizer finds the database entries that most closely match that specific measurement. To narrow the selection further, the engine compares another measurement—for example, volume—to the volume measurement of only those database models that received a high score on the pitch measurement. The process continues until the engine finds the model phoneme that most closely matches the sample phoneme across the entire range of measurements.

6 To make words out of the phonemes, the speech engine compares groups of successive phonemes to a database of known words. Typically, tens of thousands of words make up the lexicon. Speech recognition for specialized applications, such as medical or legal dictation, uses a database customized with words unique to that subject.

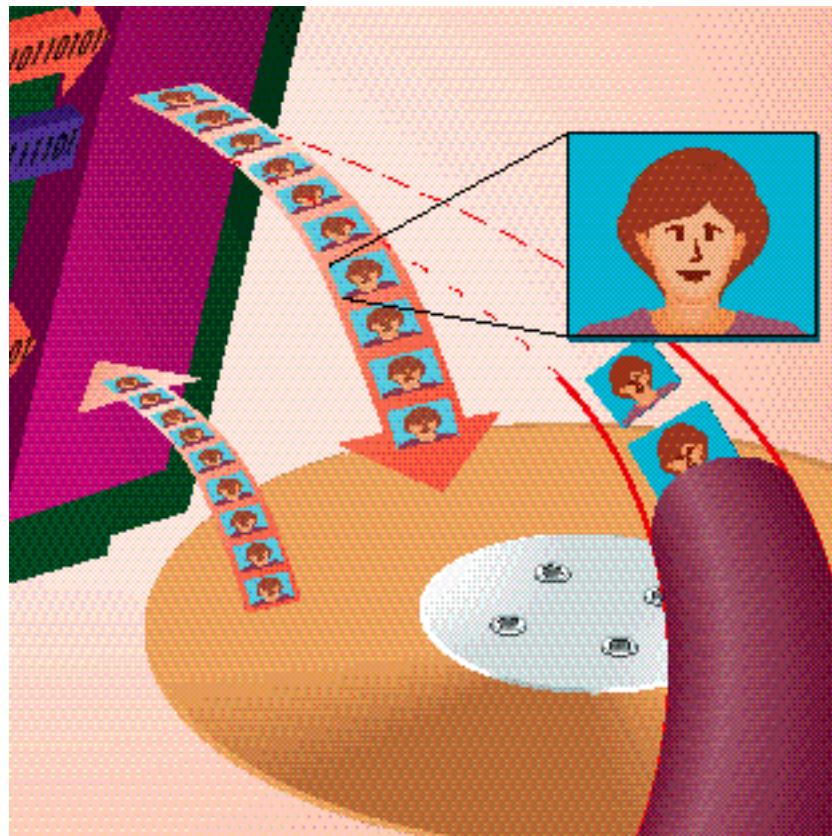
7 In the case of homonyms—words that sound alike but are spelled differently, such as “there” and “their,” the results are turned over to a natural language component, which compares the sounds with grammatical rules, a database of most common phrases, and other words used in the context of the spoken text. The natural language rules predict the most likely word combination to occur at each point in a sentence, based on the relative frequency of all groups of three words in the language. For example, it knows “going to go” is more frequent than “going, too, go” and “going two go.” It displays on screen the word combination that has the highest probability of matching what the speaker said.

8 If the speech engine cannot resolve some ambiguity, it may query the speaker, presenting a list of candidate words and allowing the speaker to choose the correct one. Or, if the engine chooses the wrong word, the speaker corrects the choice, and the change is fed back into the vocal references table of idiosyncrasies to improve the accuracy of future speech sessions.

CHAPTER

**30**

# How Multimedia Video Works



**VIDEO** is nothing new. We grew up with Howdy Doody, Gilligan, and “Cheers.” The camcorder is replacing the 35mm still camera as the memory-catcher of choice. So why does video’s arrival on PCs seem like such a big deal? It’s precisely because video in more traditional forms has become so much a part of our lives. We rely daily on talking, moving pictures to get so much of the information we need to learn, to conduct business, and to lead our personal lives.

All the excitement and technical innovations in multimedia concentrate on video and audio, which has it all backwards. The excitement isn’t really in what video brings to computers. It’s what computers bring to video. Although multimedia audio rivals a good home stereo system, video on PCs has lagged behind. Only recently, with the introduction of MPEG video compression and DVD playback, has video on a PC rivaled that of a cheap TV set.

So why not just use videotape and TVs to convey information? Because videotape doesn’t have random access. The freedom to move to any point in a stream of information is random access. It’s where RAM gets its name, random access memory. Early computers used magnetic tapes to store programs and data, and using them was slow. Random access memory and random access hard drives give computers their speed and versatility.

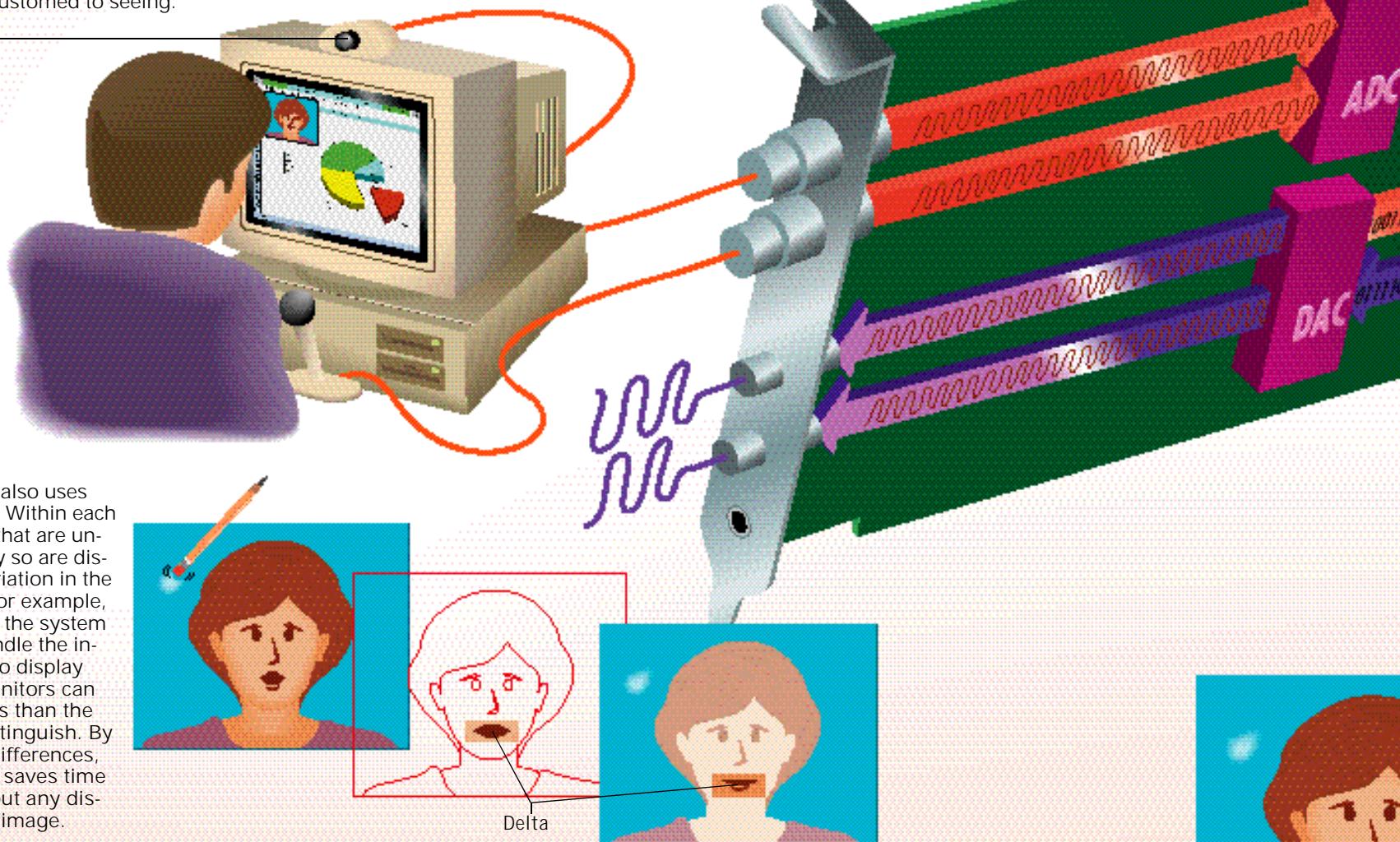
And it’s exactly this issue of access that differentiates multimedia video from a videotape. You have control over what you see and hear. Instead of following a preprogrammed course of videos and animation, you can skip about as you like, accessing those parts of a multimedia program that interest you most. Or with videoconferencing, you can interact live with another person in a different part of the world and work on the same document or graphic simultaneously.

Despite the superficial resemblance between a TV set and a PC monitor, the two produce an image in different ways. At least until high-definition digital TV becomes common, television is an analog device that gets its information from continuously varying broadcasting waves. A computer’s monitor uses analog current to control the image, but the data for what to display comes from digital information—0s and 1s.

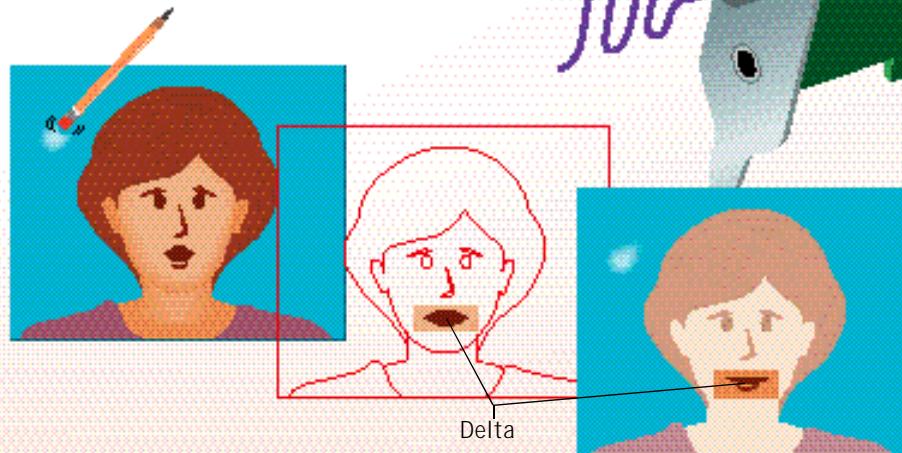
The flood of data can easily exceed what a display can handle. That’s why multimedia video is still often small and jerky. A smaller image means less information—literally, fewer pixels—the PC has to track and update. The jerkiness comes from the slow update of the image—only 5 to 15 times a second, compared to 30 frames for a TV or movie. By further increasing data compression, some of these limitations have been almost eliminated. MPEG compression, for example, lets multimedia video cover the entire screen. Further development of the techniques described here for compressing and transmitting will eventually make computer video as ubiquitous as sitcom reruns.

# How Multimedia Video Works

- 1** A camera and microphone capture the picture and sounds of a video session and send those analog signals to a video-capture adapter board. To cut down on the amount of data that must be processed, the board captures only about half the number of frames per second that movies use, which is one reason the video may look jerky—the frame rate is much slower than your eye is accustomed to seeing.



- 8** Videoconferencing also uses lossy compression. Within each frame, differences that are unnoticeable or nearly so are discarded. A slight variation in the background here, for example, is sacrificed so that the system will not have to handle the information needed to display that difference. Monitors can display more colors than the human eye can distinguish. By discarding minor differences, lossy compression saves time and memory without any discernible change in image.



- 7** Motion Pictures Expert Group (MPEG) compression, which is effective enough to produce full-screen video, only records key frames and then predicts what the missing frames look like by comparing the changes in the key frames.

- 6** Advanced forms of compressing both recorded video and videoconferencing use a sampling process to cut down on the amount of data that must be recorded or transmitted. One method, used by .AVI, makes a record of one complete video frame, and then records only the differences—the delta—in the frames that follow. Each frame is re-created by combining the delta data with the data for the frame that preceded it.



- 2** On the video-capture adapter card, an analog-to-digital converter (ADC) chip converts the wavering analog video and audio signals to a pattern of 0s and 1s, the binary language that describes all computer data.

- 3** A compression/decompression chip or software reduces the amount of data needed to re-create the video signals. As an example, the software compression for Microsoft Video for Windows looks for redundant information. Here, the background is a large expanse of a single color, blue. Rather than save the same information for each pixel that makes up the background, compression saves the color data for that exact shade of blue only once, along with directions for where to use the color when the video is replayed.

- 4** Video for Windows saves more space when it writes the video to disk by interweaving the data for the picture with the audio in a file format called .AVI (for audio/video interleave). To replay the video, the compressed and combined video and audio data is either sent through a compression/decompression chip or processed by software. Either method restores areas that had been eliminated by compression. The combined audio and video elements of the signal are separated and both are sent to a digital-to-analog converter (DAC), which translates that binary data into analog signals that go to the screen and speakers.

- 5** Instead of being recorded, the compressed video and audio signals may be sent over special telephone lines, such as an integrated services digital network (ISDN) line, that transmit the data in a digital form rather than as the analog signal used by ordinary phone lines. A similarly equipped PC at a remote location receives the digital signals, decompresses them, and converts them to the analog signals needed to control the display and audio playback. (See Chapter 35 for more on video and audio over the Internet.)

C H A P T E R

**31**

# How Virtual Reality Works

THE oxymoronic term virtual reality brings to mind the twisted logic of an Escher drawing, but it can deliver the ultimate multimedia experience. Using 3-D animation and sound, virtual reality takes you through exotic worlds, futuristic shoot-outs, and flying aircraft that yaw, pitch, and roll. As with the best multimedia, the key to virtual reality is interactivity. Rides at Disneyland move you through real three-dimensional space, but the ride is always the same, always predictable. Virtual reality gives you control of where you go, what you do, and who you encounter. You have 3-D freedom.

Still, no one could confuse VR worlds with the real one, or even with the holodeck on "Star Trek." Virtual reality uses an enormous amount of processing power to produce moving 3-D environments based on your own movements. But virtual reality systems have become powerful enough—and inexpensive enough—for the mainstream. Virtual reality systems are popping up in public entertainment centers everywhere. Some of these even include capsules that a user climbs into to simulate the motion of an airplane or land vehicle. On the home front, there are also peripherals that can turn your PC into a virtual reality producer.

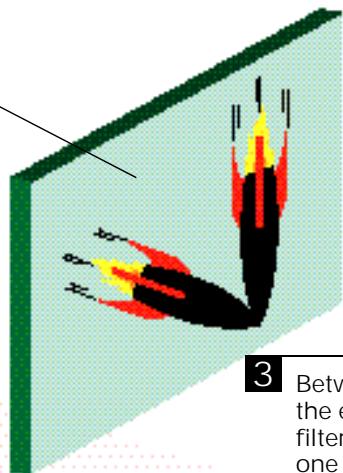
There is no definitive virtual reality system. There is no checklist to delineate exactly what a virtual reality system consists of. Instead, the degrees of virtual reality are usually described in terms of user immersion into the VR world. On one end of the VR spectrum are light partial-immersion systems, which might consist of a joystick or 3-D mouse, a PC monitor, and software. Other partial-immersion systems might make use of special glasses to view the monitor. Total-immersion systems—where the user sees and hears nothing but the virtual reality environment—involve head-mounted displays and a data glove, both of which use sensors to track your position and orientation in the virtual world. The effect is that when you physically turn around, you'll see what's behind you. Virtually speaking, that is.

In this chapter, we'll look at how a typical full-immersion PC system and on-the-fly 3-D animation produce their captivating worlds.

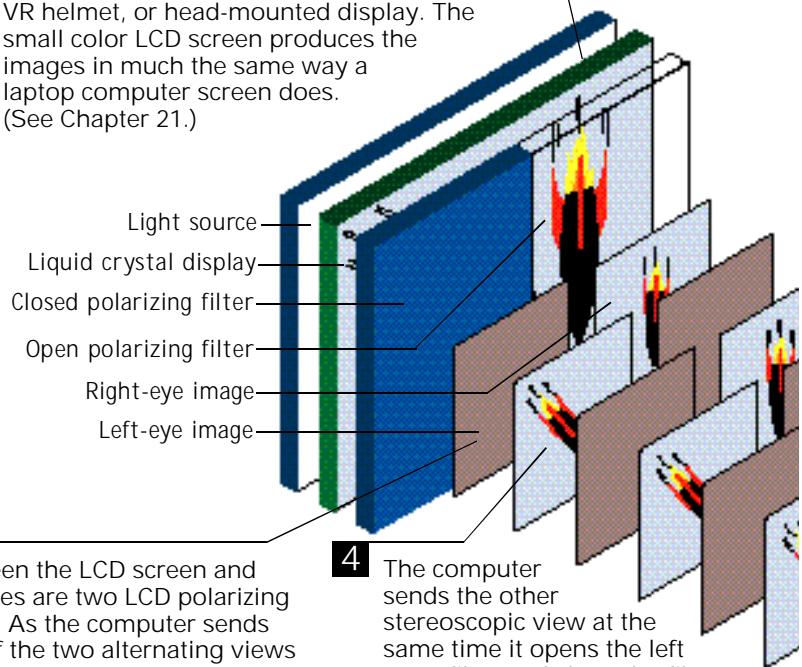
# How Virtual Reality Devices Work

## VR Helmets

- 1** To create the illusion of visual depth, virtual reality (VR) software is constantly calculating the two views of its virtual world to correspond to the way two eyes see the same scene from slightly different angles.



- 2** Instead of sending images to a monitor, the computer transmits the images to liquid crystal display (LCD), mounted in front of the eyes in a VR helmet, or head-mounted display. The small color LCD screen produces the images in much the same way a laptop computer screen does. (See Chapter 21.)



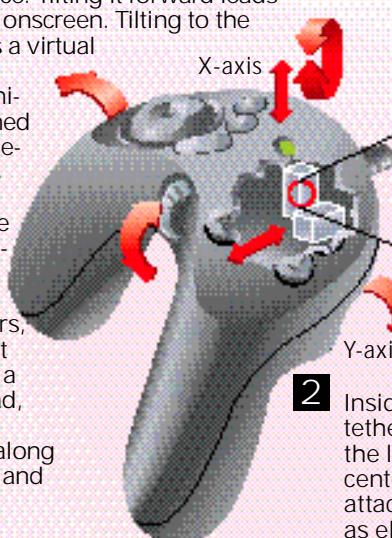
- 3** Between the LCD screen and the eyes are two LCD polarizing filters. As the computer sends one of the two alternating views of the scene, it also turns off the polarizing filter in front of the left eye so that only the right eye sees the image.

- 4** The computer sends the other stereoscopic view at the same time it opens the left eye's filter and shuts the filter over the right eye. The left eye sees the scene from a perspective that is slightly shifted from the view the right eye saw.

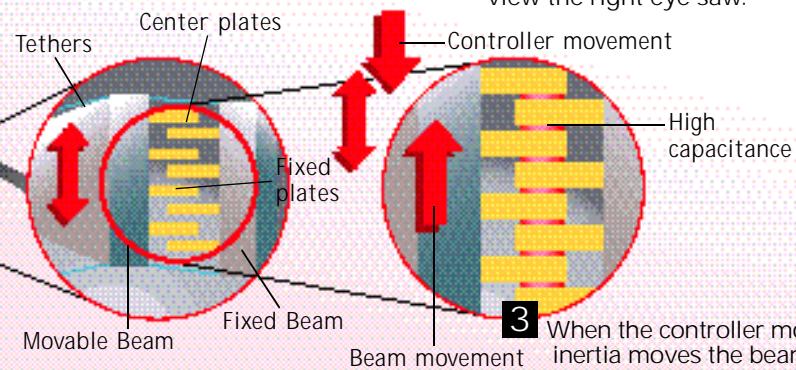
## Accelerometer

- Game controllers such as Microsoft's Sidewinder Freestyle Pro control movement through a virtual reality by sensing how the controller is moving in real space. Tilting it forward leads you down onscreen. Tilting to the left causes a virtual left turn.

Inside, a micromachined accelerometer detects movements. The accelerometer is made of two sensors, each about the size of a match head, that are mounted along the X-axis and Y-axis.



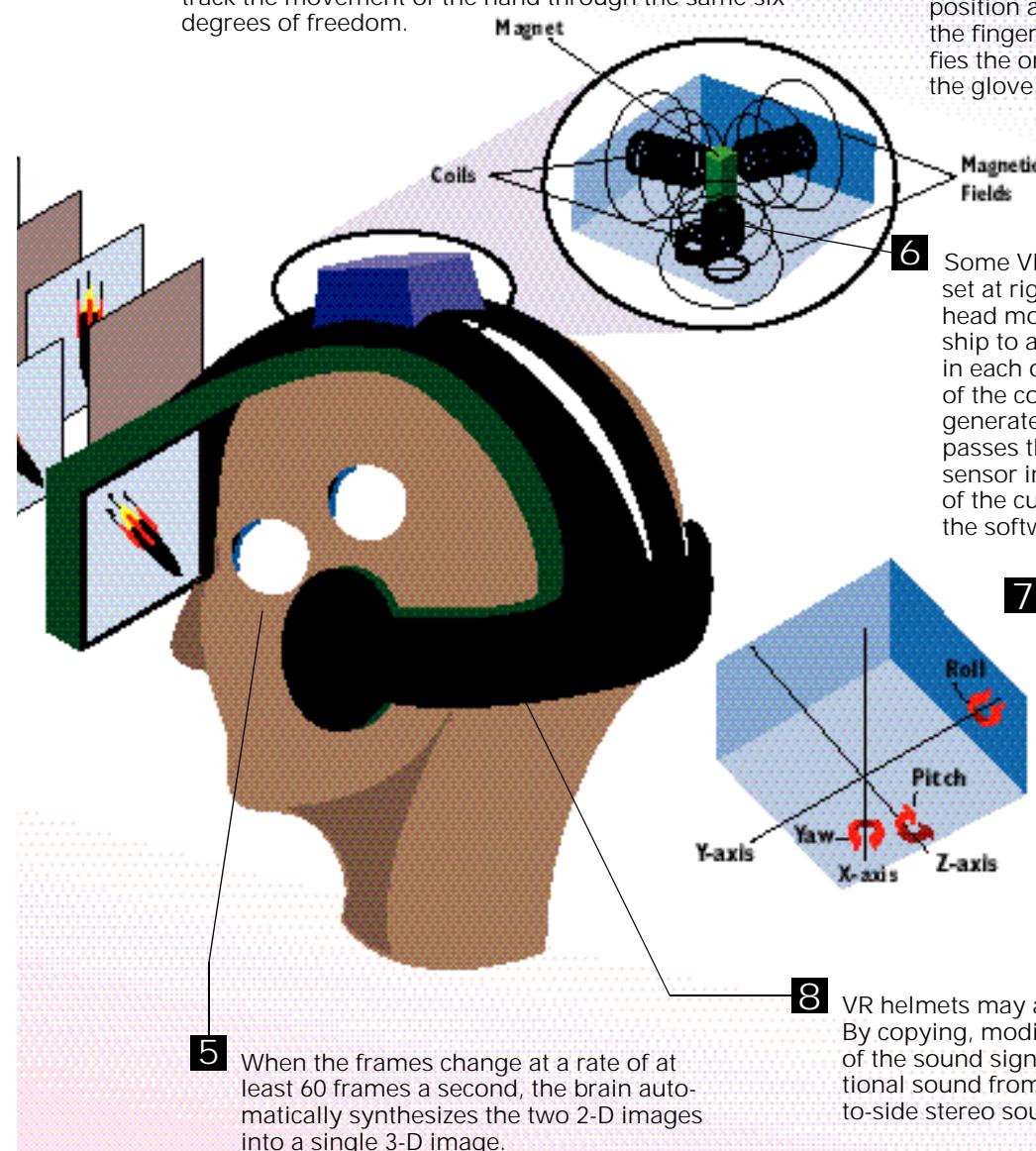
- 2** Inside each sensor is a relatively heavy beam tethered to flexible connections. Studding the length of the beam, several dozen rigid center plates stick out between fixed plates attached to the sensor's shell. The plates act as electrical capacitors with an electrical charge set for a capacitance of 0 volts.



- 3** When the controller moves, inertia moves the beams. The center plates move closer to the fixed plates in the opposite direction of movement. As the plates get closer together, the capacitance rises. Increases in voltage are measured and sent to the software to tell it which way to move onscreen.

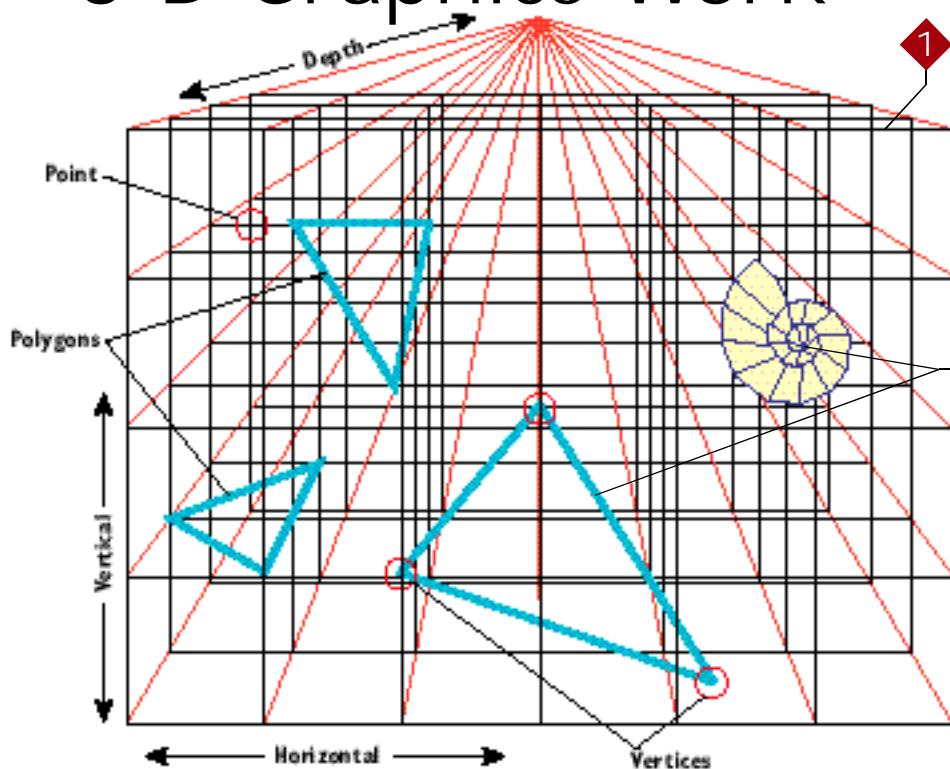
## Data Glove

- 1 At the end of each finger a light emitting diode (LED) shines light through optical fibers woven into the glove's material.
- 2 The fibers carry the light up the fingers to photo transistors at the base of each finger. As a finger bends, it compresses the optical fiber so that less light passes through it. The photo transistors constantly measure the varying light intensities, and send that information to the VR software.
- 3 Like helmets, data gloves also have a 6DOF sensor to track the movement of the hand through the same six degrees of freedom.



- 4 The software uses the data from the glove's 6DOF sensor and fiber optics to calculate the position and orientation of the hand and how the fingers are bent. Then the software modifies the onscreen hand to match the pose of the glove.
- 5 Some VR helmets contain three coils of wire set at right angles to each other. When the head moves, it moves the coils in relationship to a magnet, generating electric currents in each of the coils. The different orientation of the coils to the magnet causes each to generate a different amount of electricity as it passes through the same magnetic field. A sensor inside the helmet reads the strength of the currents and sends the information to the software.
- 6 The VR program interprets the signals to calculate the head's orientation in the six degrees of freedom (6DOF). The first three degrees describe the position of the sensor (called a 6DOF sensor) in space—along its X, Y, and Z coordinates. The second three degrees of freedom describe the rotation around each of the three axes—yaw, pitch, and roll. Some helmets sense movement by listening to the echoes of ultrasonic waves.
- 7 VR helmets may also provide earphones for sound. By copying, modifying, and delaying selected parts of the sound signals, the earphones simulate directional sound from the front and rear as well as side-to-side stereo sound.

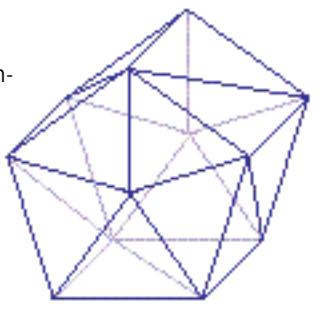
# How Accelerated 3-D Graphics Work



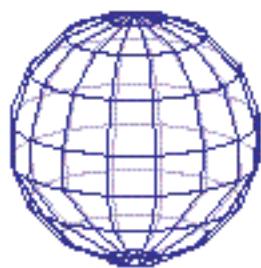
**1** Imagine space being divided into an endless number of cubes stretching off to infinity in three directions. A single point in space is defined by providing three values along the three axes of the cubes for the point's vertical, horizontal, and depth positions.

**2** In three-dimensional space, three points are the minimum needed to define a two-dimensional plane, or polygon. Virtual reality worlds—a room or an entire game level—are constructed of such polygons, usually triangles, because they have the fewest angles, or vertices, making them the easiest and quickest polygon to calculate. Even a square, rectangle, or rounded area is created as a combination of triangles.

- 3** Three-dimensional objects are created by connecting two-dimensional polygons. Even curved surfaces are made up of flat planes. The smaller the polygons, the more curved an object appears to be. (A process called spline animation calculates curved lines, but it is not used as commonly as flat polygons.) The software's geometry engine calculates the height, width, and depth information for each corner of every polygon in a 3-D environment, a process called tessellation, or triangulation. The engine also figures out the current "camera angle," or vantage point, from which any part of a level is seen. It rotates, resizes, or repositions the triangles as the viewpoint changes. Any lines outside the viewpoint are eliminated, or clipped. The engine also calculates the position of any light sources in relation to the polygons. Tessellation makes intense use of floating-point math. An MMX component in some Pentium-class processors includes special instructions for making these calculations, and some 3-D accelerator adapter cards include coprocessors to ease the calculation burden on the main CPU. A changing scene must be redrawn at least 15 to 20 times a second for the eye to see smooth movement.

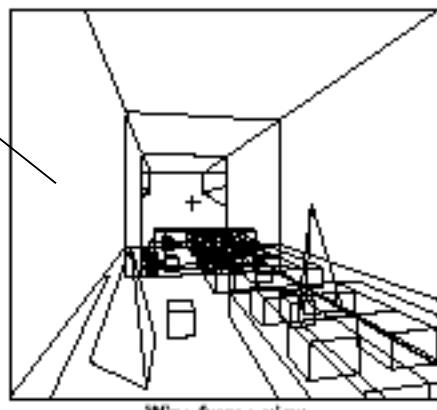


— = **Foreground**  
— = **Background**

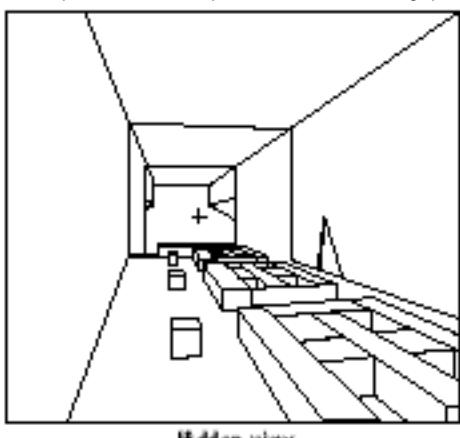


Polygon 3-D object

- 4** The results of the geometry engine's calculations are passed to the rendering engine, most likely located on a 3-D accelerator card. It has the job of rasterization—figuring out a color value, pixel by pixel, for the entire 2-D representation of the 3-D scene. It first creates a wire-frame view in which all the vertices that define the polygons are connected by lines, or wires. The result is like looking at a world made of glass with lines visible only where the panes of glass intersect.

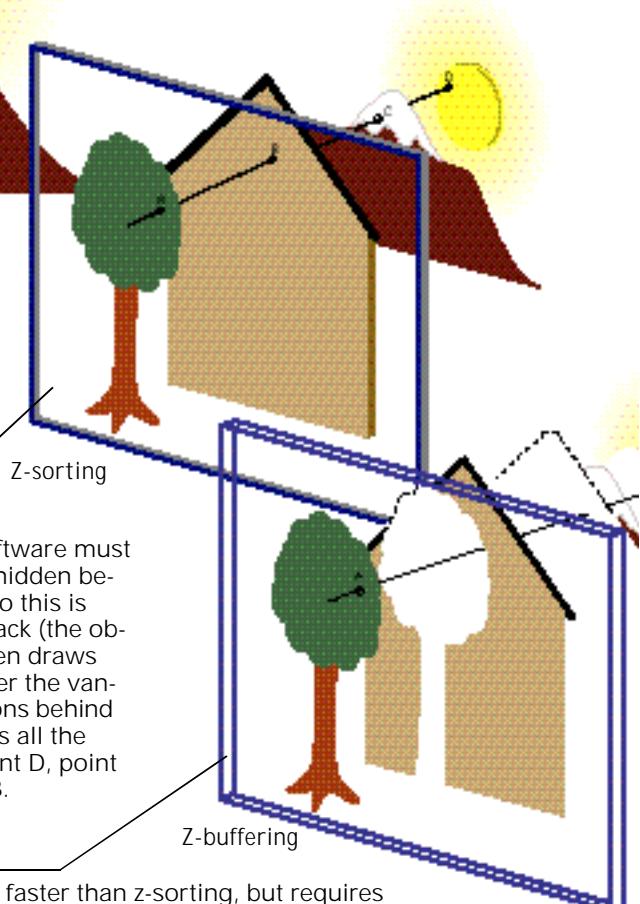


- 5** To eliminate the effect of looking through glass, the 3-D software must determine, from the camera's viewpoint, what objects are hidden behind other objects. The easy, memory-conserving way to do this is z-sorting. The rendering engine sorts each polygon from back (the objects closest to the distant vanishing point) to front, and then draws each polygon completely in that order so that objects nearer the vantage point are drawn last and cover all or part of the polygons behind them. In the illustration here of z-sorting, the engine renders all the points A, B, C, and D on the line AD. But point C covers point D, point B is painted over point C, and finally point A covers point B.



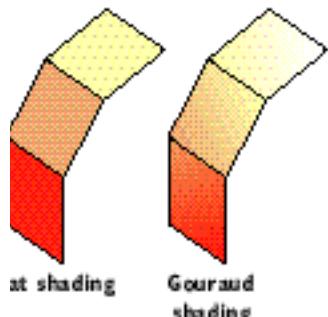
- 6** Z-buffering is faster than z-sorting, but requires more memory on the video card to record a depth value for each pixel that makes up the surface of all the polygons. Those pixels that are nearer the vantage point are given smaller values. Before a new pixel is drawn, its depth value is compared to that of the pixels along the same AB line that passes through all the layers of the image. A pixel is drawn only if its value is lower than that of all the other pixels along line AB. In the illustration here of z-buffering, pixel A is the only one the engine bothers to draw because pixels in the house, mountain, and sun along line AB would be covered by pixel A. With either z-sorting or z-buffering, the result is called a hidden view, because it hides surface that should not be seen.

[Continued on next page.]



# How Accelerated 3-D Graphics Work

(Continued)

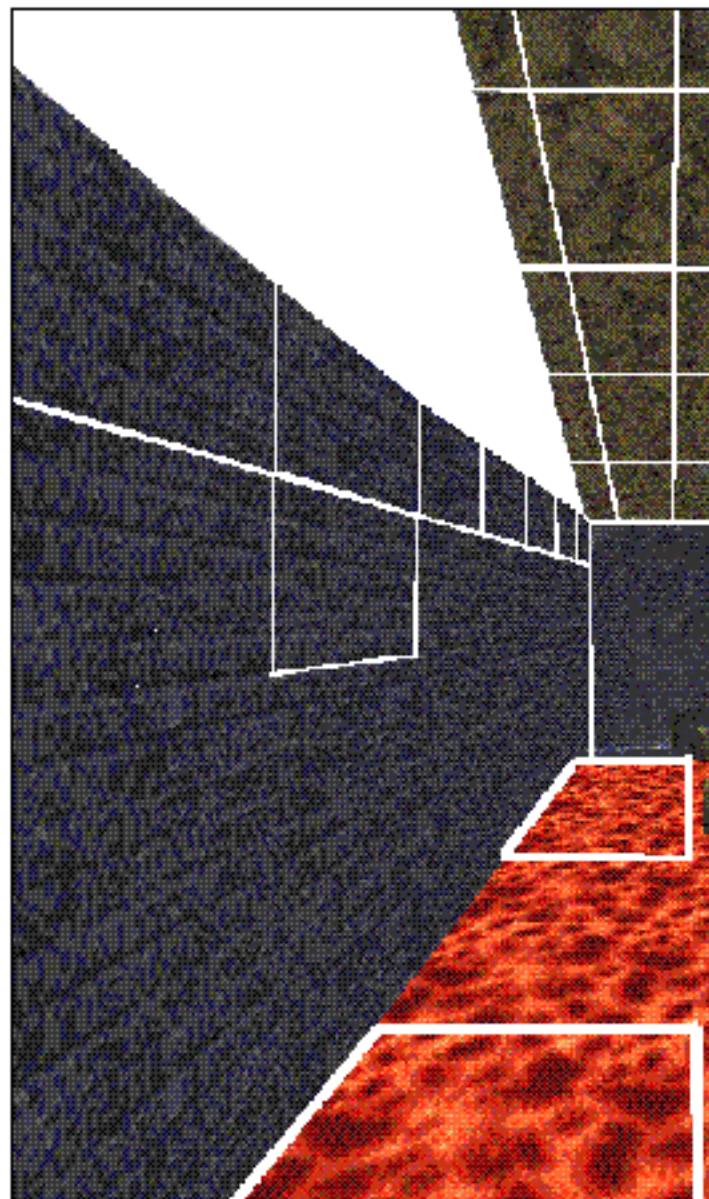


- 7** Using information from the geometry engine about the location of light sources, the rendering engine applies shading to surfaces of the polygons. The rudimentary flat shading applies a single amount of light to an entire surface. Lighting changes only between one surface and an adjacent surface. A more sophisticated and realistic method is Gouraud shading, which takes the color values at each vertex of a polygon and interpolates a graduated shading extending along the surface of the polygon from each vertex to each of the other vertices.

Texture maps



- 8** In the real world, few surfaces are smooth. Computerized 3-D worlds use texture maps to simulate realistic surfaces. Texture maps are bitmaps—unchanging graphics—that cover surfaces like wallpaper. In this scene from a Quake level—and seen on the previous pages in wireframe and hidden views—texture maps are tiled to cover an entire surface. In simple 3-D software, a distortion called pixelation occurs when the viewpoint moves close to a texture-mapped object: The details of the bitmap are enlarged and the surface looks as if it has large squares of color painted on it.



- 9** MIP mapping corrects for pixelation. The 3-D application uses variations of the same texture map—MIP stands for multim in parvum or “many in few”—at different resolutions, or sizes. One texture map is used if an object is close up, but a bitmap saved at a different resolution is applied when the same object is distant.

- 10** Perspective correction makes tiles of texture maps at the far end of a wall narrower than tiles near the viewer and changes the shape of texture maps from rectangles to more of a wedge shape.

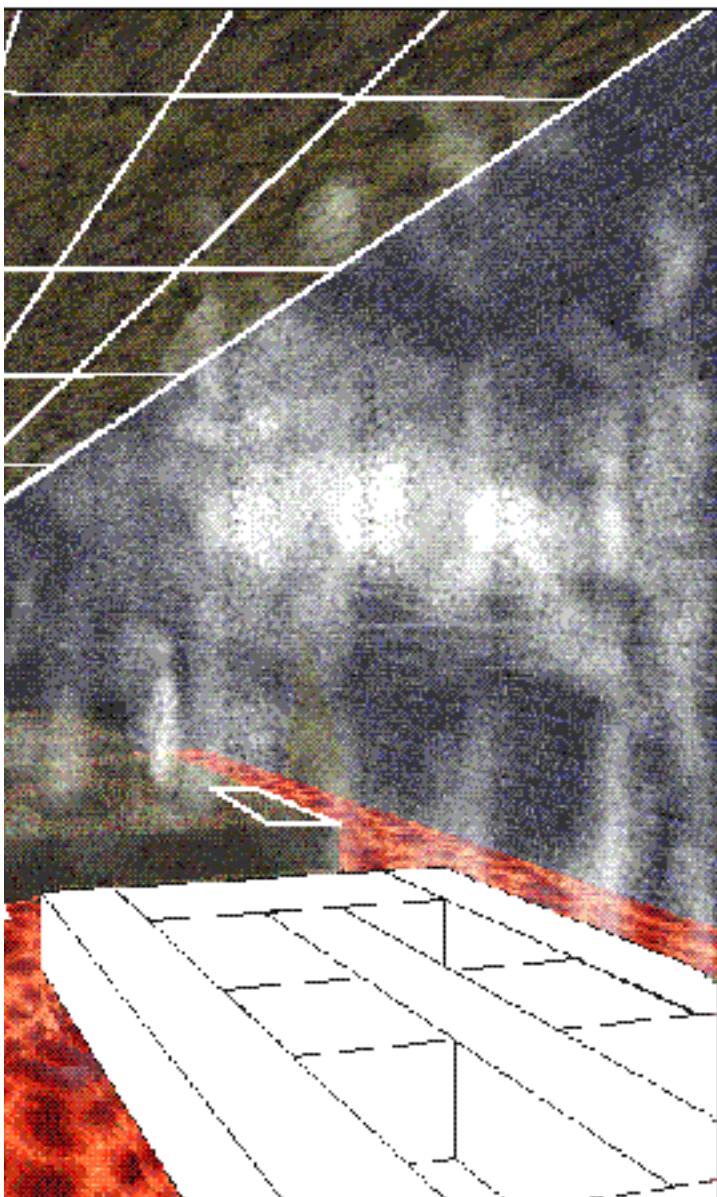


Distance fogged

Not fogged

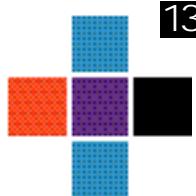
11

Fogging—not to be confused with the effect of wisps of fog that comes from alpha blending—and depth cueing are two sides of the same effect. Fogging, shown in the next screen shot from British Open Golf, blends distant areas of a scene with white to create a hazy horizon. Depth cueing adds black to colors by lowering the color value of distant objects so that, for example, the end of a long hall is shrouded in darkness. The effect is not merely atmospheric. It relieves the rendering engine of the amount of detail it has to draw.



12

To create effects such as the semi-transparency that occurs through smoke or under water, the rendering engine uses alpha blending between texture maps that represent the surface of an object and another texture map representing such transient conditions as fog, clouds, blurriness, or a spreading circle of light. The rendering engine compares the color of each texel in a texture map with a texel in the same location on a second bitmap, takes a percentage of each color and produces an alpha value somewhere between the two colors. A less memory-intensive way to accomplish a similar effect is stippling to blend two texture maps. Instead of performing calculations on each pair of texels, stippling simply draws the background texture and then overlays it with only every other texel of the transparency texture.



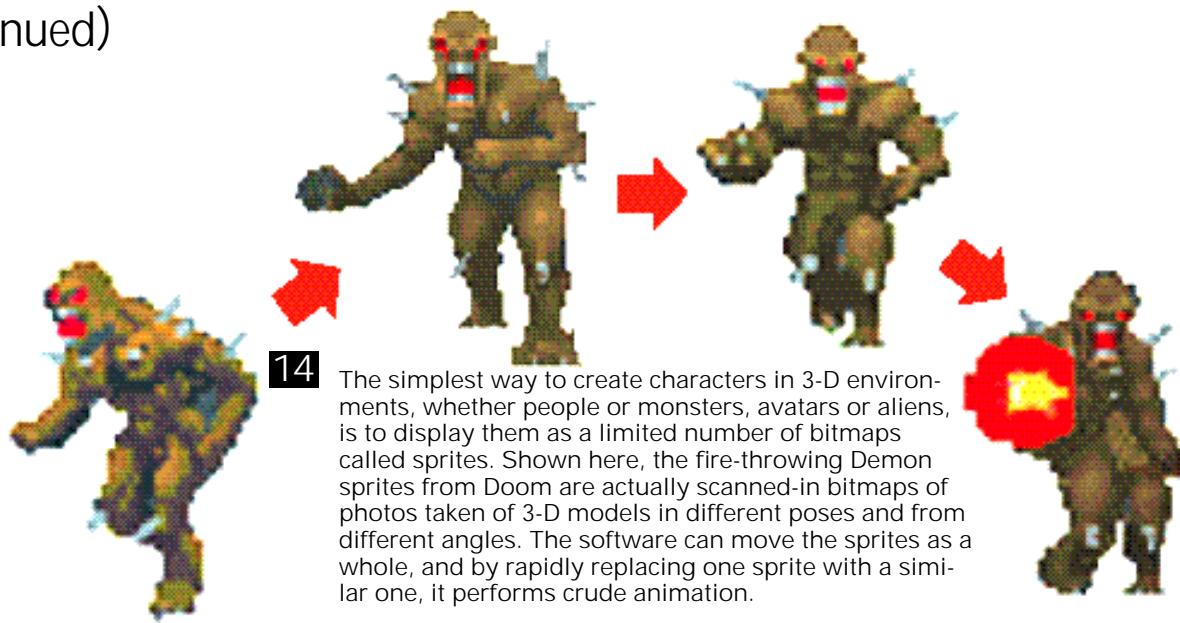
13

Bilinear filtering smoothes the edges of textures by measuring the color values of four surrounding texture-map pixels, or texels, and then making the color value of the center texel an average of the four values. Trilinear filtering smoothes the transition from one MIP map to a different size of the same texture.

[Continued on next page.]

# How Accelerated 3-D Graphics Work

(Continued)

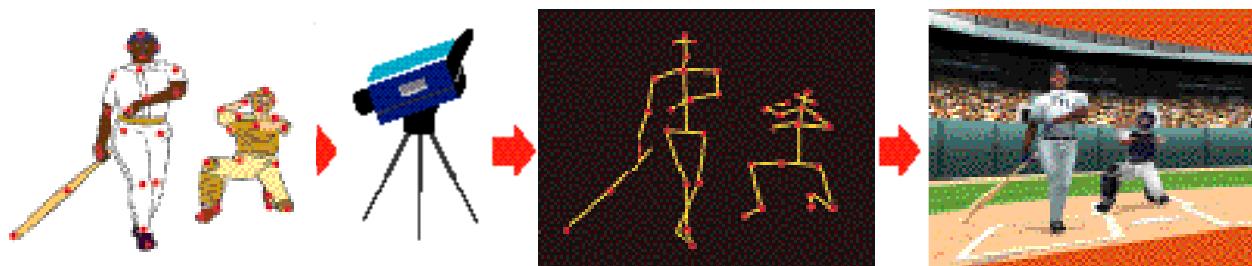


14

The simplest way to create characters in 3-D environments, whether people or monsters, avatars or aliens, is to display them as a limited number of bitmaps called sprites. Shown here, the fire-throwing Demon sprites from Doom are actually scanned-in bitmaps of photos taken of 3-D models in different poses and from different angles. The software can move the sprites as a whole, and by rapidly replacing one sprite with a similar one, it performs crude animation.

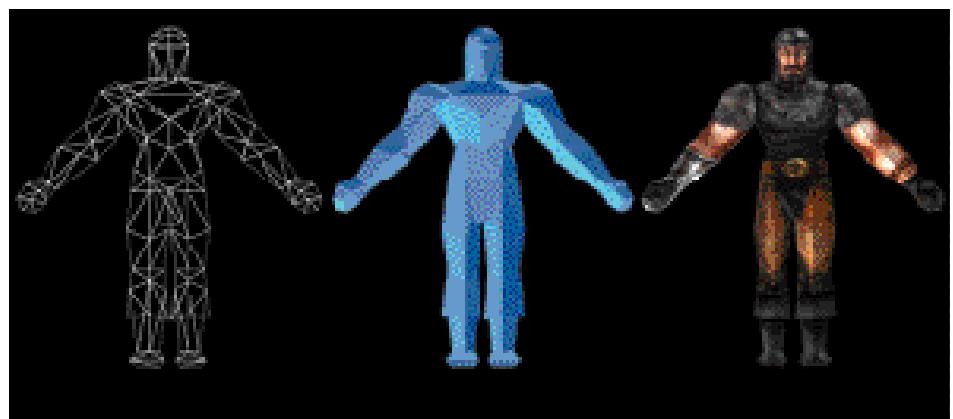
15

Some games, such as Microsoft Baseball 3D shown here, use motion capture to automate the animation process. A person—often a real sports figure—is videotaped wearing lights at joints such as the shoulders, elbows, knees, and ankles. The movement of the lights is fed to a computer that uses the information to reposition the joints of the polygons making up a character. The animation is limited to the specific motions that have been recorded and animated.



6

Full polygon animation creates a character the same as any other object in the 3-D world. It is constructed of polygons that obey rules governing how they move and interact with other objects. In a detailed construction, such as this character from Die by the Sword, individual muscles are articulated so that, for example, an arm simply doesn't change position; you see the arm's muscles flex. The software creates this type of animation on-the-fly rather than displaying hand-drawn or computer-drawn animation. This gives characters the freedom to interact with objects and other characters without the limitations of pre-rendered animation frames. The only limits are those of the physical laws governing such things as inertia, mass, and gravity that are programmed into the environment.



## How the 3-D Video Card Works

- 17** The most recently used texture maps are cached in the video card's own RAM on the theory—usually right—that if one frame of animation requires a brick wall texture map, it's likely the next frame will need the same map. By caching the maps, the video card can retrieve them faster than if it had to go to a computer's main RAM. The exception is a video card working from an accelerated graphics port (AGP). In this case, the texture maps may be held in a computer's main memory, but the AGP's special path between memory and the video cards allows the texture maps to be retrieved more quickly from main RAM. (See Chapter 18.)
- 
- 18** A 3-D graphics coprocessor contains instructions optimized specifically for making the extensive calculations required by on-the-fly 3-D rendering. It relieves the PC's main processor of some of its workload and can do the job faster than the main processor. But if a game or other software has not been tweaked to take advantage of the coprocessor's unique abilities, the job is still done by the main processor.
- 19** On some 3-D cards, a 2-D graphics coprocessor—sometimes called a graphics accelerator—contains instructions optimized for the creation of dialog boxes, icons, bitmaps, and other familiar features of the Windows landscape. Some video cards have a 2-D coprocessor and no 3-D coprocessor.
- 20** After the color value of a pixel has been determined by being pushed through all the effects available on a 3-D accelerator card, the values are written to one of two frame buffers, which are areas of memory located in high-speed chips on the video board called VRAM. This is memory designed specifically for the needs of video cards. It is dual-ported. The PC can write to the memory (to change what will be displayed), while the video adapter continuously reads the memory (to refresh the monitor's display). The card reads these values to pass through a digital-to-analog converter to the monitor, where the values change the color and intensity of the actual physical pixels glowing on the inside of your screen.
- 21** As those values are being read, the 3-D accelerator card is assembling the next frame in a second frame buffer, a technique called double buffering. When the second frame is completed, its values are fed to the monitor and the card recycles the first buffer with values for the third frame. One buffer is always being read while the other is being constructed.



P A R T



# HOW THE INTERNET WORKS

Chapter 32: How Local Area Networks Work  
242

Chapter 33: How a LAN Connects to the Internet  
250

Chapter 34: How E-Mail Works  
256

Chapter 35: How Internet Video and Audio Work  
260

Chapter 36: How the World Wide Web Works  
266

IT would be a lot easier to explain how the Internet works if you could hold it in your hand. Hardware—real, tangible, with a weight and size—is always easier to understand because you can see it and you can point with confidence and say this gizmo leads to that gadget, every time. The Internet is not just a single thing. It is an abstract system. To understand the significance of this term, consider a less abstract system—your body.

The molecules that make up your body are not the same all your life. New molecules are constantly being taken in as food, water, and air and are recombined into different molecules of muscle, blood, and bone. But no matter which molecules make up your hair and eyes and fingers at any moment, the structure of your body remains the same. Your heart doesn't refuse to pump because new molecules of blood are created. If you remove some parts of your body, the system continues to function—sometimes, as in the case of brain damage, transferring the job of the missing parts to healthy parts.

As a system, the Internet is similar to a living organism. It grows, taking in new “molecules” in the form of PCs and networks that attach themselves to the Internet. Parts of the Internet communicate with other parts that then respond with some action, not unlike the muscle activity set off by nerve impulses. You can think of the Internet as a network of networks. Amoeba-like, smaller networks can break off the Internet and live independent lives. Unlike amoebas, those smaller networks can later rejoin the main body of the Internet.



The Internet is ephemeral. Some pieces—the supercomputers that form the backbone of the Internet—are always there. The local area networks (LANs) found at countless businesses qualify as individual organs in the Body Internet. But nothing is really fixed in place—hard-wired. Each time you use your PC to connect with, say, a PC in Pittsburgh that maintains information on “Star Trek,” you don’t have to use the same phone lines, switching devices, and intermediate networks to reach it. The Internet’s route to Pittsburgh may one time run through Chicago; the next time it may run through Copenhagen. Without realizing it, you may bounce back and forth among several networks from one end of the country to the other, across an ocean and back again, until you reach your destination in cyberspace.

Explaining how the Internet works is difficult; it’s a lot easier to explain what you can get from the Internet: information of all kinds. Being a system without physical limitations, it’s theoretically possible for the Internet to include all information on all computers everywhere, which in this age means essentially everything the human race knows, or thinks it knows. But because the Internet is such an ad-hoc system, exploring it can be a challenge. And you don’t always find exactly what you want. There are plenty of software tools that make surfing the Internet easier, but the Internet itself has no overall design to help those using it. You’re pretty much on your own when you jump in with whatever software you can find.

Despite the amorphous nature of the individual elements that make up the Internet, it is possible to describe the structure of the Internet—the system that always remains the same even as the elements that make it up are changing from moment to moment.

## KEY CONCEPTS

**backbone** The main computer and LANs that form the basis of the Internet.

**bridge** A device that connects two local area networks, even if they are different types of LANs.

**browser** A PC program that displays pages from the Internet.

**client** A computer or software that depends on another computer—a server—for data, other programs, or the processing of data. Part of a client/server network.

**domain** A group of computers on a network that are administered as a unit, usually by the same company or organization.

**e-mail** Electronic mail sent within a network or over the Internet.

**gateway** Hardware and software that link two networks that work differently, such as a Novell and a Windows NT network.

**GIF** File extension for graphics interchange format, a compressed, bitmapped graphics format often used on the Web for animated graphics.

**HTML (hypertext markup language)** The coding used to control the look of documents on the World Wide Web.

**http** Part of a URL that identifies the location as one that uses HTML.

**hub** A device where various computers on a network or networks on the Internet connect.

**Internet** A worldwide network with more than 100 million users that are linked for the exchange of data, news, conversation, and commerce. The Internet is decentralized—that is, no one person, organization, or country controls the Net or what is available through it.



**IP, Internet provider** A computer system that provides access to the Internet, such as AOL and Concentric. Most phone companies are IPs. IP also sometimes stands for Internet protocol, a format for contents and addresses of packets of information sent over the Net. See also TCP/IP.

**IP address** An identifier for a computer or device on a TCP/IP network. Networks using the TCP/IP protocol route messages based on the IP address of the destination. The format of an IP address is a 32-bit numeric address written as four numbers separated by periods. Each number ranges from 0 to 255.

**link** Text or graphics on a Web page that lead you to other pages if you click them.

**local area network (LAN)** A more-or-less self-contained network (that may connect to the Internet), usually in a single office or building.

**search engine** A program that searches documents located on the Internet for key words or phrases entered by a person browsing the Net. It returns a list of sites, sometimes rated, related to the topic searched for. HotBot, Yahoo!, and Excite are examples of sites that provide search engines.

**server** Part of a network that supplies files and services to clients. A file server is dedicated to storing files, and a print server provides printing for many PCs. A mail server handles mail within a network and with the Internet.

**TCP/IP (transmission control protocol/Internet protocol)** A collection of methods used to connect servers on the Internet and to exchange data. TCP/IP is a universal standard for connecting to the Net.

**URL (universal resource locator)** An address of documents and other resources on the Web.

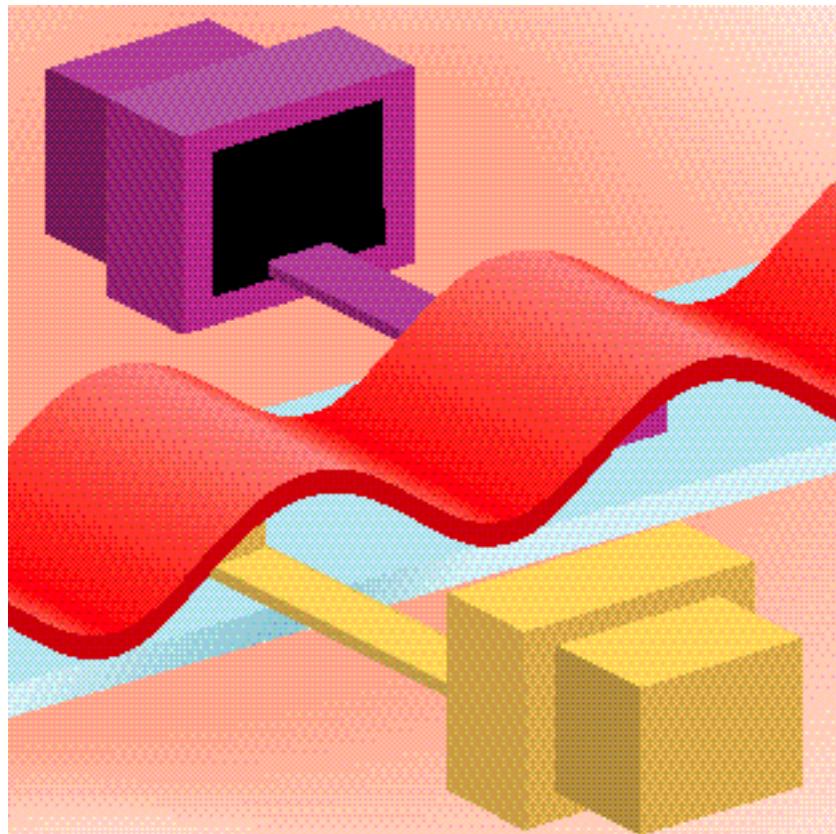
**wide area network (WAN)** A single network that extends beyond the boundaries of one office or building.

**World Wide Web (WWW)** A loose confederation of Internet servers that support HTML formatting.

CHAPTER

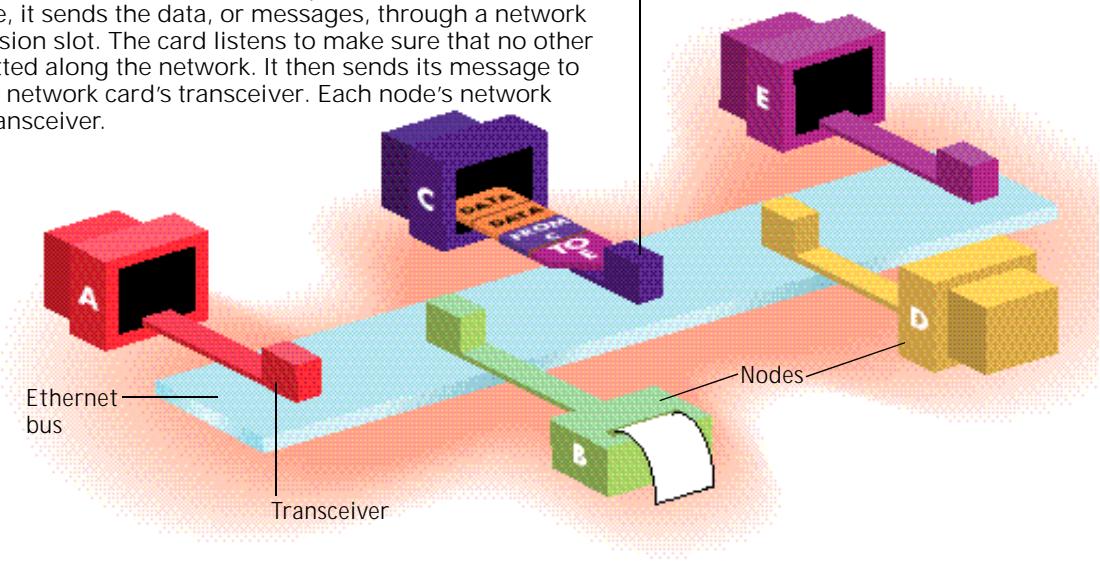
# 32

# How Local Area Networks Work

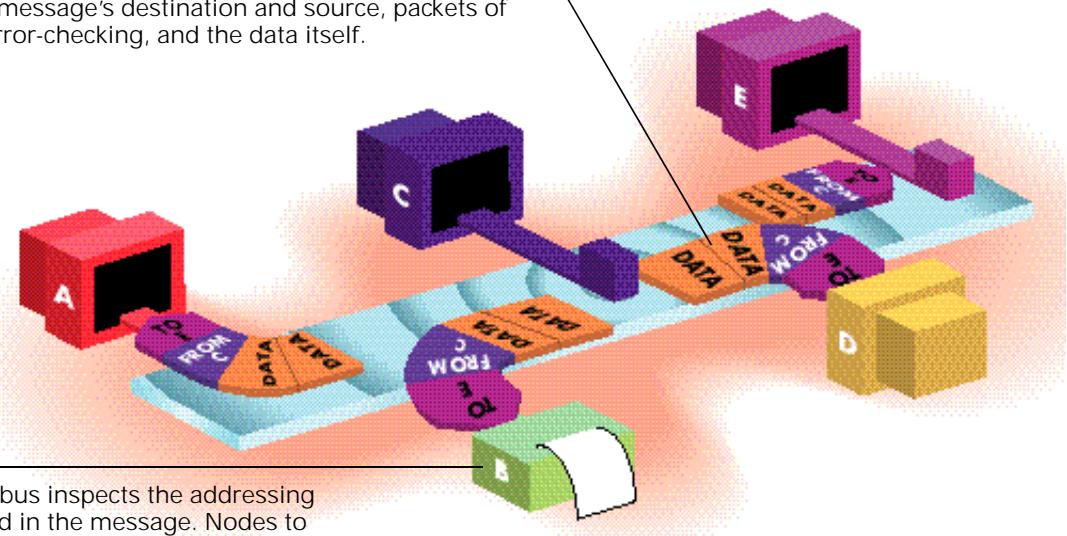


# How an Ethernet Network Works

- 1** All nodes, clients and servers, on an Ethernet network—also called a bus network—are attached to the LAN as branches off a common line. Each node has a unique address. When a node—a PC, file server, or print server—needs to send data to another node, it sends the data, or messages, through a network card installed in an expansion slot. The card listens to make sure that no other signals are being transmitted along the network. It then sends its message to another node through the network card's transceiver. Each node's network connection has its own transceiver.



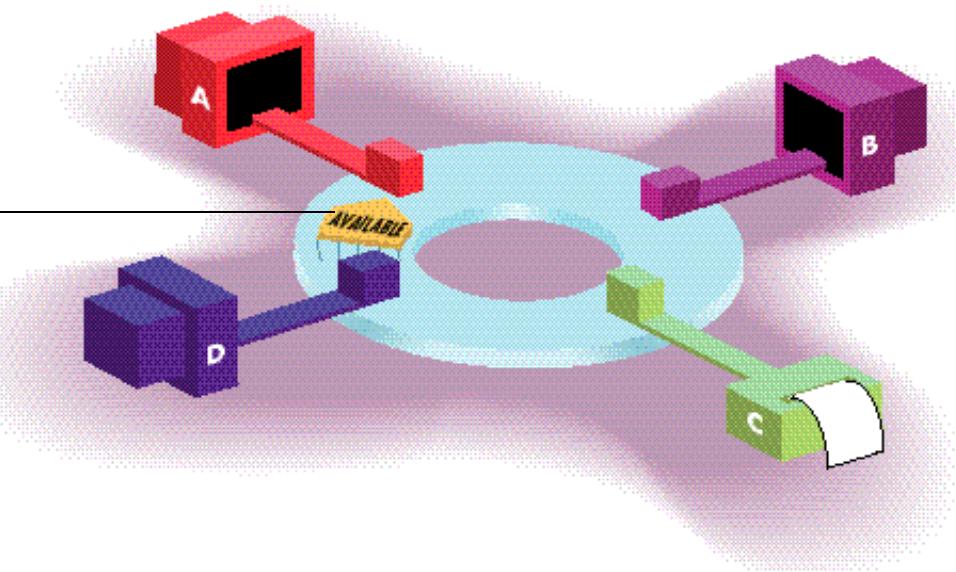
- 2** The transceiver broadcasts the message in both directions so that it will reach all other nodes on the network. The message includes the addresses of the message's destination and source, packets of data to be used for error-checking, and the data itself.



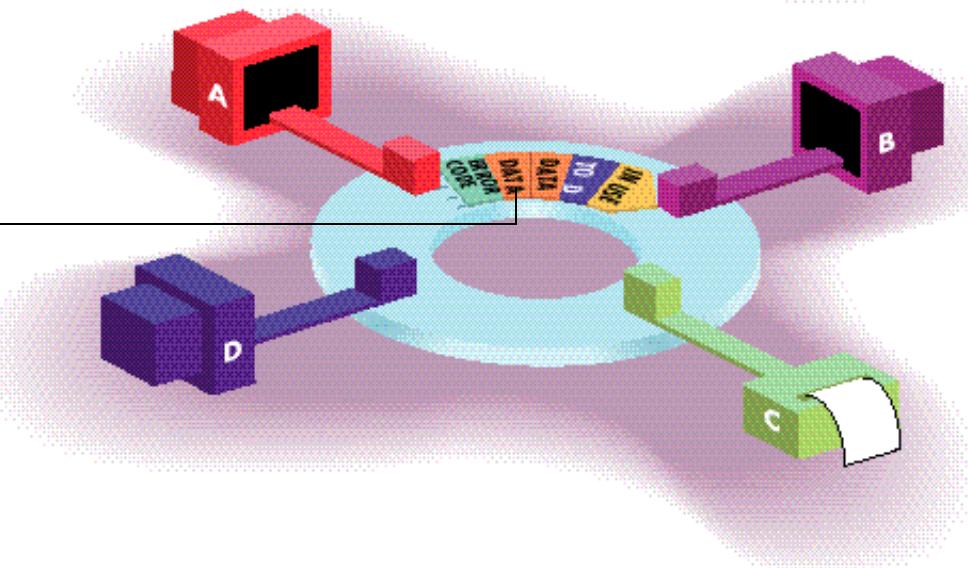
- 3** Each node along the bus inspects the addressing information contained in the message. Nodes to which the message is not addressed ignore it.

# How a Token-Ring Network Works

- 1 All nodes on a token-ring network are connected to the same circuit, which takes the form of a continuous loop. A token—which consists of a short all-clear message—circulates continuously around a loop and is read through a token-ring adapter card in each node as the token passes by.

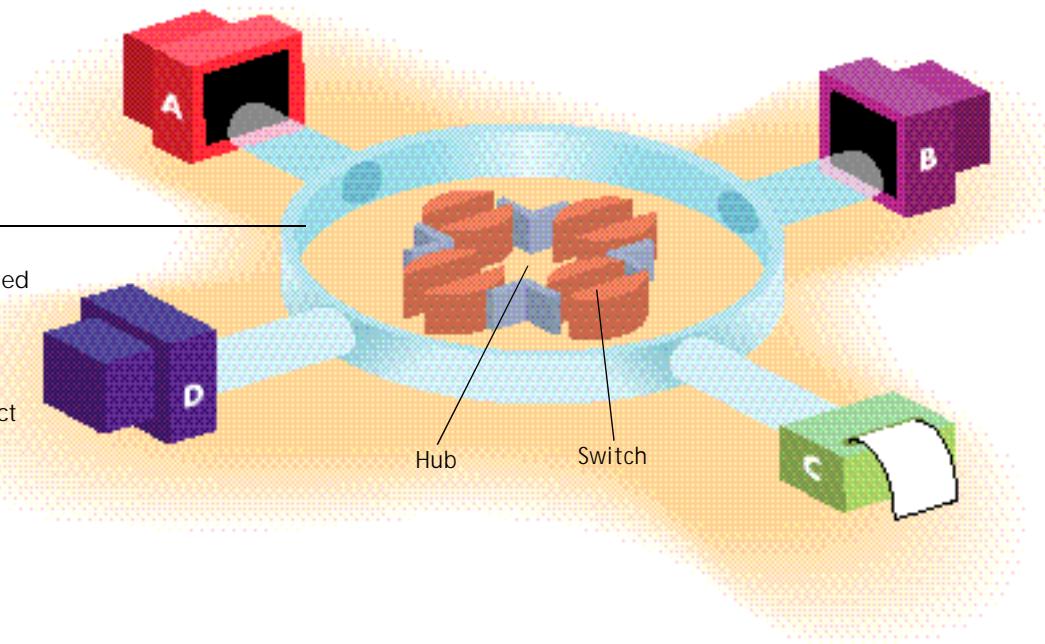


- 2 A node wanting to send a message grabs the token as it passes by, changes the binary code in the token to say that it is in use, and attaches a message, along with the address of the node for which the message is intended and error-checking code. Only one message at a time can be circulated on the network.

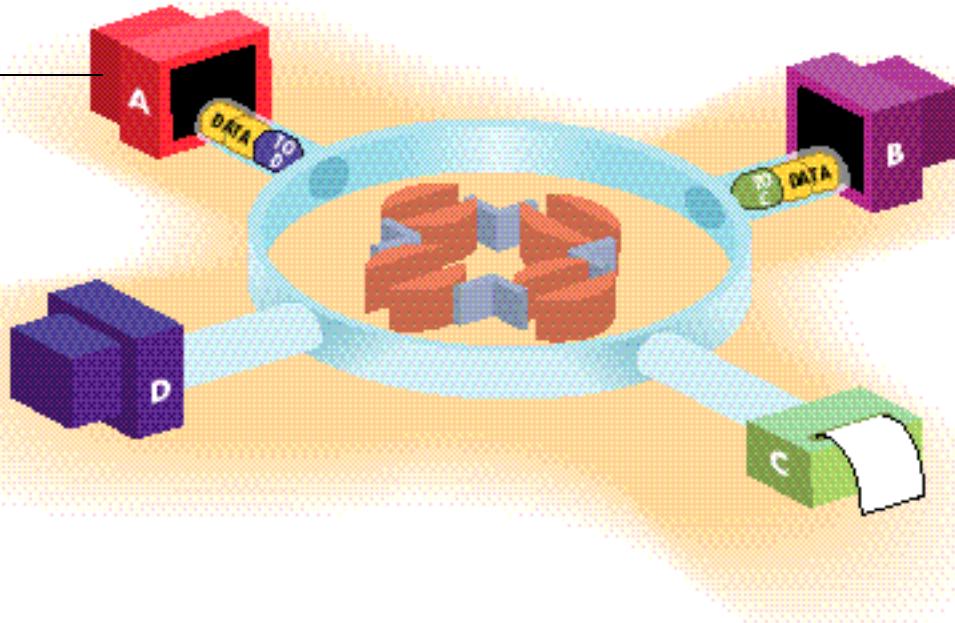


# How a Star Network Works

- 1 Nodes in a star network configuration are attached to separate lines, all of which lead to the same hub, or central station. The central station contains switches to connect any of the lines to any other line.



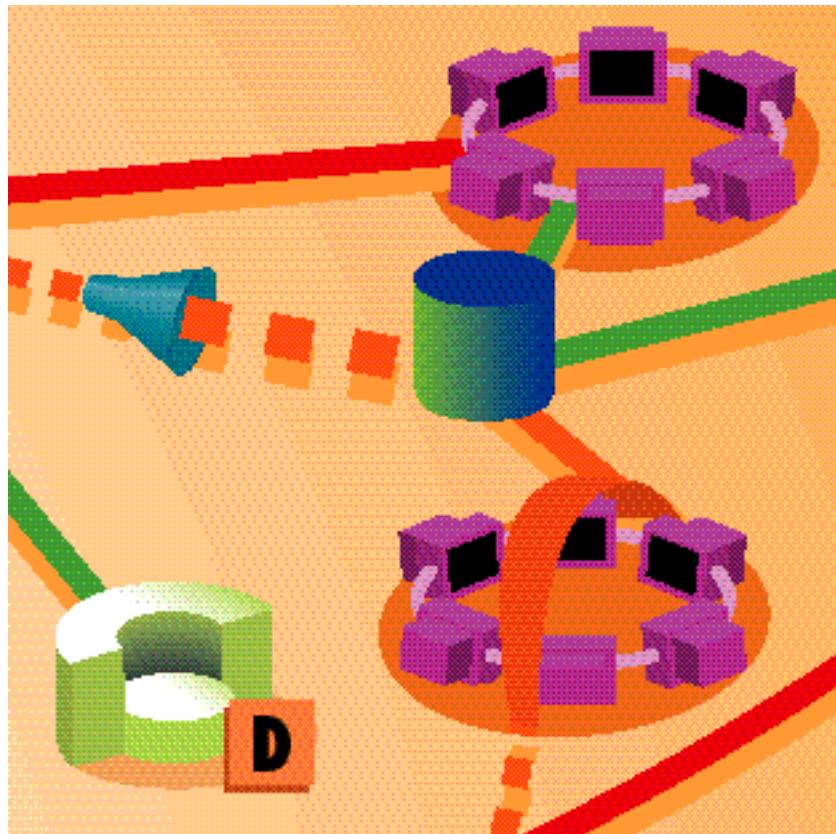
- 2 A node sends to the central station a message that includes the address of the node for which the message is intended and the data and error-checking code. More than one node can originate a message at the same time.



CHAPTER

33

# How a LAN Connects to the Internet



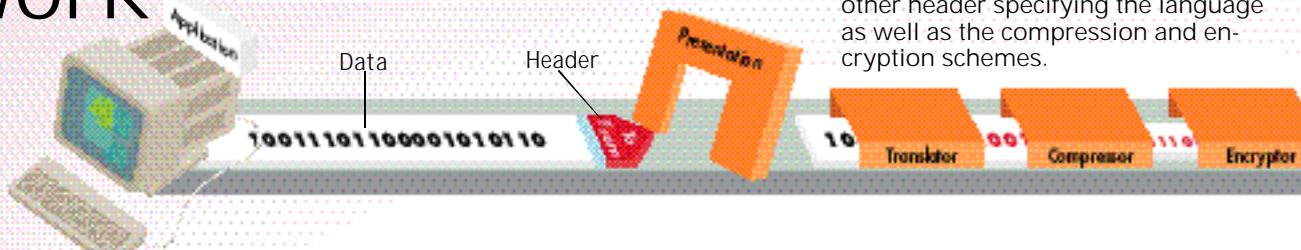
THE Internet is like a singing dog. One is amazed, not that the dog sings well, but that it sings at all. In the case of the Internet, it's amazing, not that it ties together the entire world into one giant conglomeration of civilization's accumulated data, but that your mom got the e-mail you sent her yesterday.

Even the simplest communication over the Net is broken up into scores of discrete packages of data. Those packages are sent over thousands of miles of telephone lines, satellite signals, and microwaves—and all the packages in a single communication may not be sent over the same path. Along the way, they pass through systems running operating systems as diverse as UNIX, NetWare, Windows, and AppleTalk, through PCs, mainframes, and minicomputers—hardware and software that were never designed, really, to work peacefully with one another. Messages are coded and decoded, compressed and decompressed, translated and retranslated, bungled and corrected, lost and repeated, shredded and stitched back together again, all so many times that it's a miracle anything even resembling coherency gets from one place to the other, much less that it gets there filled with gorgeous fonts, musical fanfares, eye-popping photos, and dazzling animation.

Want to make it even more amazing? Consider this: No one's in charge! There's no one who owns the Internet. Oh, people and companies own bits and pieces of it, and there are some organizations that by common consent rule over things such as the domain names that appear in addresses such as [www.zdnet.com](http://www.zdnet.com), [www.whi.tehouse.gov](http://www.whi.tehouse.gov), [www.ucla.edu](http://www.ucla.edu), and [ron\\_whi\\_te@zd.com](mailto:ron_whi_te@zd.com). But their rule is entirely at the whim of the governed. There's no technological reason why someone couldn't start a rival Net service—and in fact that's exactly what services such as America Online, CompuServe, Microsoft Network, and Prodigy are, except that they're more than just themselves, because they are also portals to the whole of the Internet. There is, however, one good reason why someone would not start another Internet—it'd be stupid. The thing that makes the Internet the Internet is its universal receptiveness. It's open to anyone with access to a computer and a phone line. It includes more information, and disinformation, than you will learn in your lifetime.

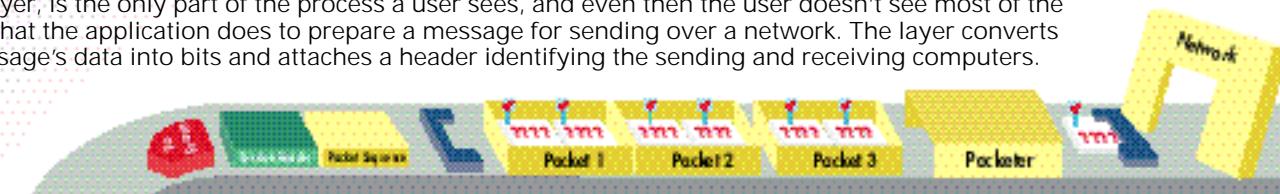
The Internet is built on cooperation. The illustrations in this chapter show both how complex Internet communications are and how the technical challenges they present are overcome. The illustrations don't show an important element—the people behind these languages, routers, protocols, and . . . contraptions that all cooperate to make sure you can send a dancing, singing e-mail to Mom on her birthday.

# How Network Communications Work



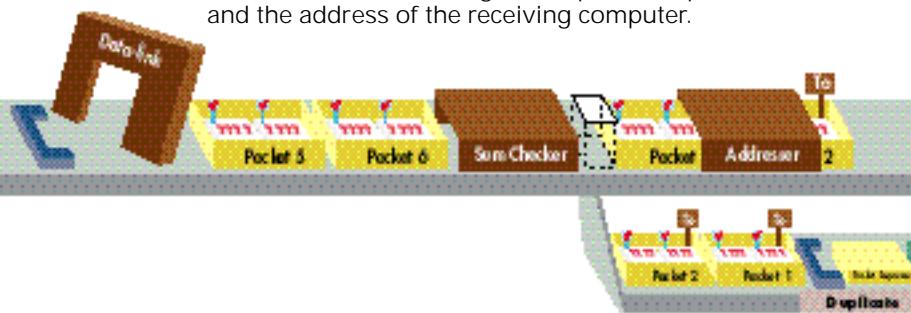
**2** The presentation layer translates the message into a language that the receiving computer can understand (often ASCII, a way of encoding text as bits). This layer also compresses and perhaps encrypts the data. It adds another header specifying the language as well as the compression and encryption schemes.

- 1** For a message, file, or any other data to travel through a network, it must pass through several layers, all designed to make sure the data gets through intact and accurate. The first layer, the application layer, is the only part of the process a user sees, and even then the user doesn't see most of the work that the application does to prepare a message for sending over a network. The layer converts a message's data into bits and attaches a header identifying the sending and receiving computers.

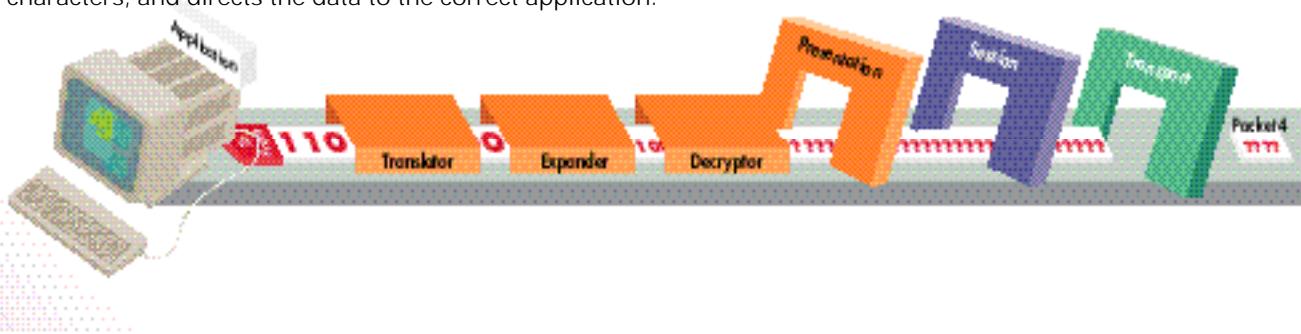


- 6** The data-link layer supervises the transmission. It confirms the checksum, then addresses and duplicates the packets. This layer keeps a copy of each packet until it receives confirmation from the next point along the route that the packet has arrived undamaged.

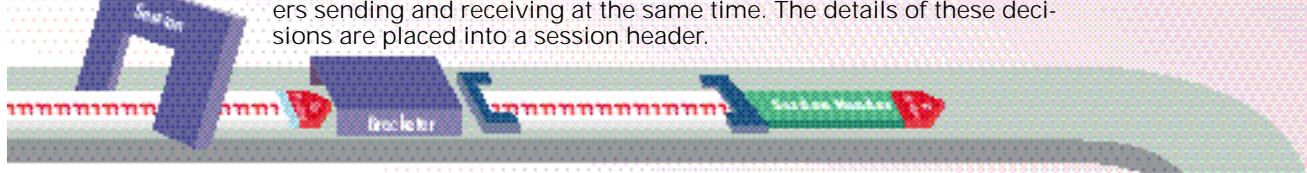
**5** The network layer selects a route for the message. It forms segments into packets, counts them, and adds a header containing the sequence of packets and the address of the receiving computer.



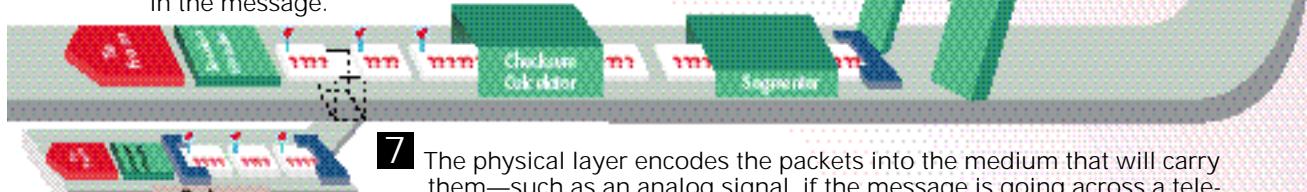
- 9** At the receiving node, the layered process that sent the message on its way is reversed. The physical layer reconverts the message into bits. The data-link layer recalculates the checksum, confirms arrival, and logs in the packets. The network layer re-counts incoming packets for security. The transport layer recalculates the checksum and reassembles the message segments. The session layer holds the parts of the message until it is complete and sends it to the next layer. The presentation layer decrypts, expands, and translates the message. The application layer identifies the recipient, converts the bits into readable characters, and directs the data to the correct application.



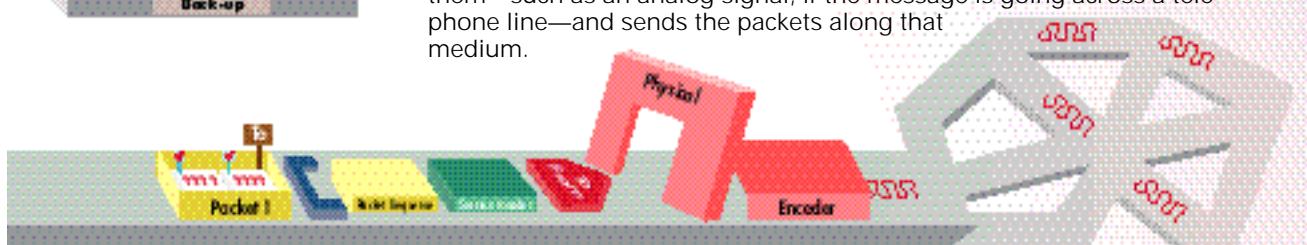
- 3 The session layer opens communications. It sets boundaries (called brackets) for the beginning and end of the message, and establishes whether the message will be sent half-duplex, with each computer taking turns sending and receiving, or full duplex, with both computers sending and receiving at the same time. The details of these decisions are placed into a session header.



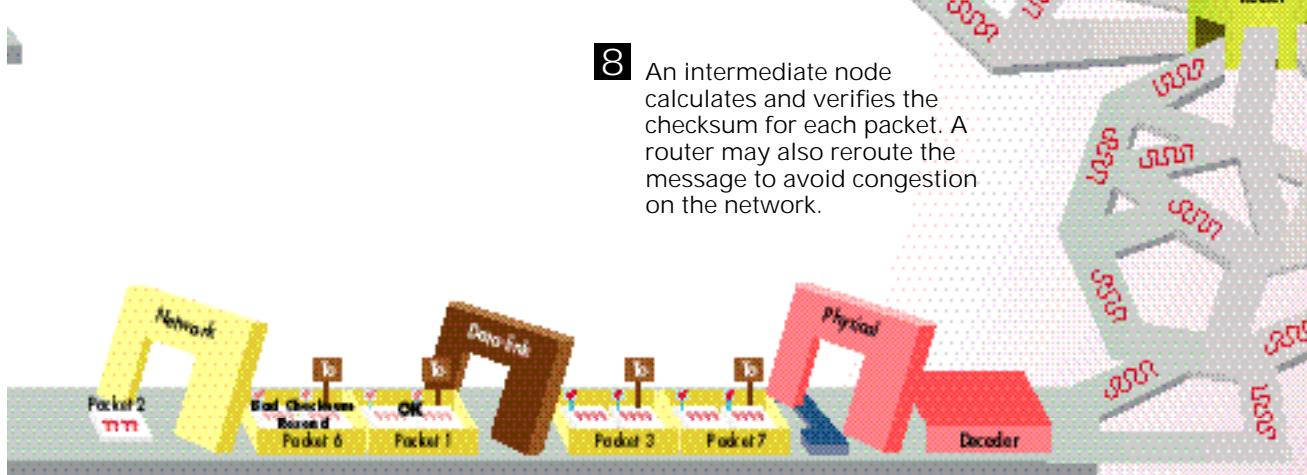
- 4 The transport layer protects the data being sent. It subdivides the data into segments, creates checksum tests—mathematical sums based on the contents of data—that can be used later to determine whether the data was scrambled. It also makes backup copies of the data. The transport header identifies each segment's checksum and its position in the message.



- 7 The physical layer encodes the packets into the medium that will carry them—such as an analog signal, if the message is going across a telephone line—and sends the packets along that medium.



- 8 An intermediate node calculates and verifies the checksum for each packet. A router may also reroute the message to avoid congestion on the network.



# How Data Travels the Internet

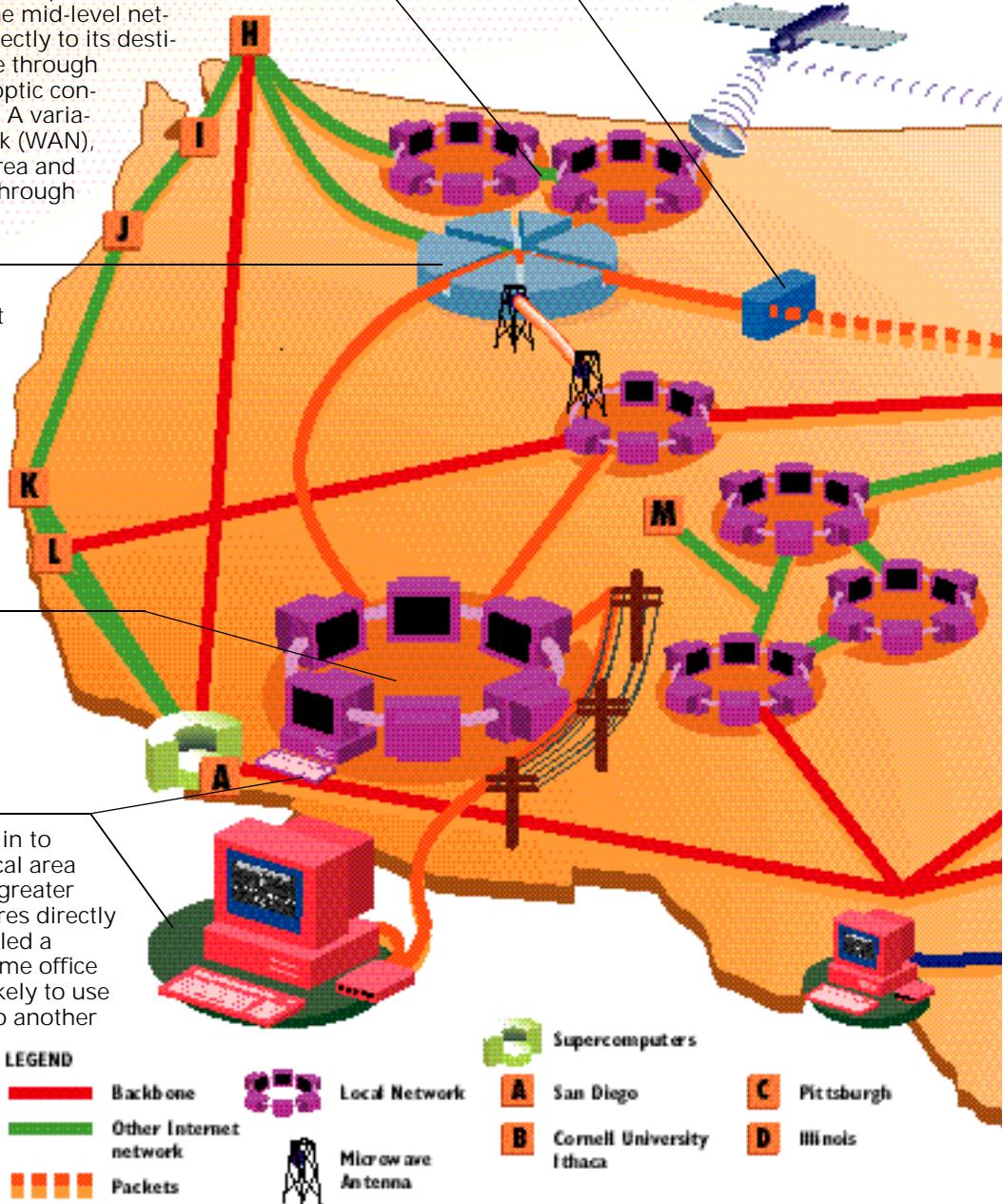
**4** Several networks in the same region may be grouped into a mid-level network. If your request is destined for another system within the same mid-level network, the router sends it on directly to its destination. This is sometimes done through high-speed phone lines, fiber-optic connections, and microwave links. A variation, called a wide area network (WAN), covers a larger geographical area and may incorporate connections through orbiting satellites.

**3** A router is a device on a network that connects networks. It inspects your request to determine what other part of the Internet it's addressed to. Then, based on available connections and the traffic on different parts of the Net, the router determines the best path to set the request back on its track to the proper destination.

**2** Your local host network makes a connection on another line to another network. If the second network is some distance away, your host LAN may have to go through a router.

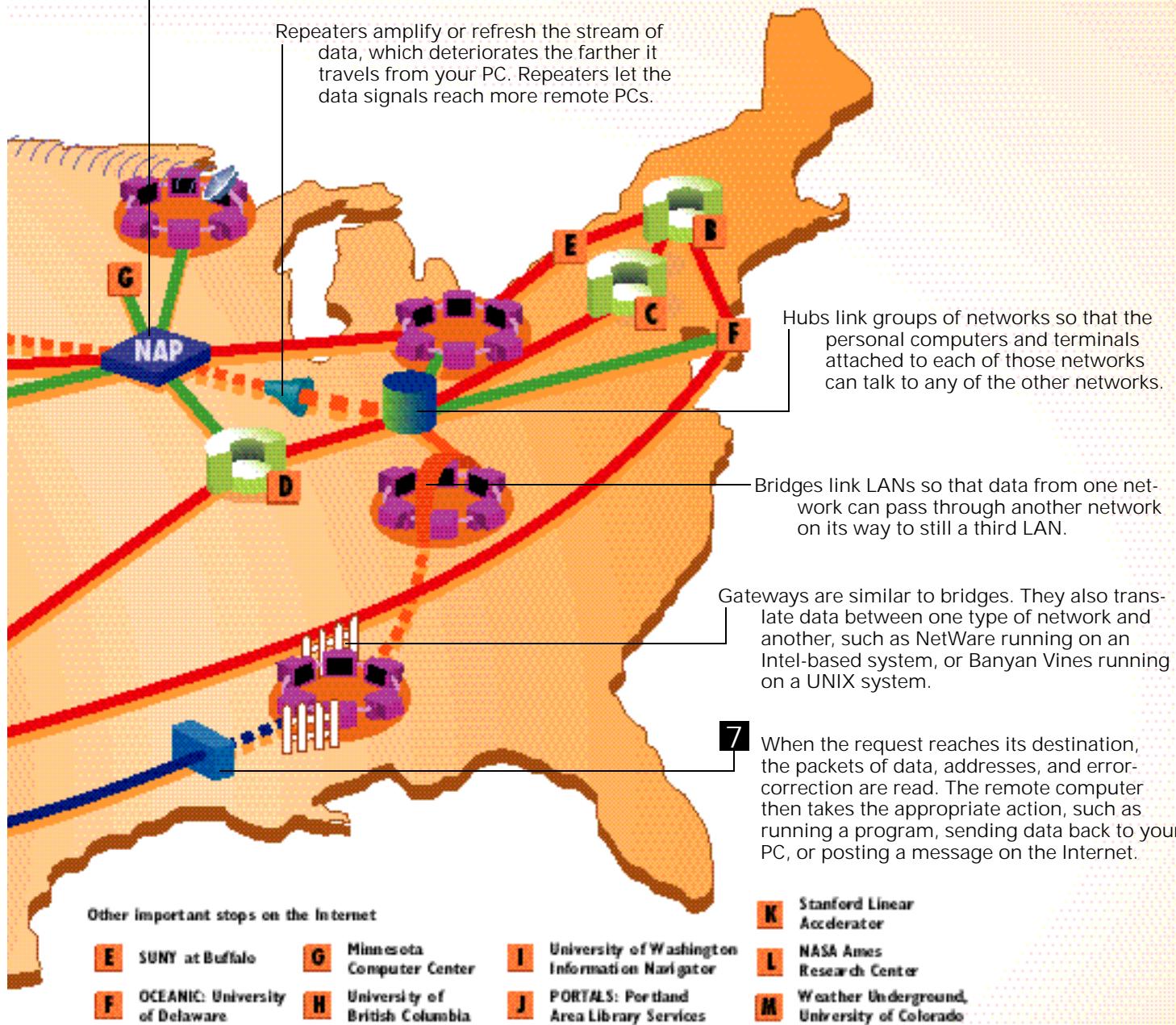
**1** In the office, a PC typically jacks in to the Internet by being part of a local area network (LAN) that is part of the greater Internet. The network, in turn, wires directly to the Internet through a port called a T-connection. In a small office/home office (SOHO) situation, a PC is more likely to use a modem to connect via phone to another modem that is wired directly to a network. Either way, through your browser you ask to see a page of information, and maybe multimedia, located on another computer somewhere else on the Internet.

**5** As the request passes from network to network, a set of protocols, or rules, creates packets. Packets contain the data itself as well as the addresses, error checking, and other information needed to make sure the request arrives intact at the destination.



**6**

If the destination for your request isn't on the same mid-level network or WAN as your host network, the router sends the request to a network access point (NAP). The pathway may take any of several routes along the Internet's backbone, a collection of networks that link extremely powerful supercomputers associated with the National Science Foundation. The Internet, however, isn't limited to the United States. You can connect to computers on the Net in virtually any part of the world. Along the way, your request may pass through repeaters, hubs, bridges, and gateways.



CHAPTER

34

# How E-Mail Works



**ELECTRONIC** mail is everywhere. Many people in business, government, and education now use e-mail more than the telephone to communicate with their colleagues. And in their private lives, many use e-mail as an inexpensive but quick method of keeping in touch with friends and family scattered around the world.

Although e-mail has been around since the formative years of the Internet, electronic mail first achieved mass popularity on local area networks. LAN-based e-mail systems allow people in an office or campus to resolve problems without holding meetings, and to communicate with others without the dreariness of formal hard-copy memos. Today, the proprietary e-mail systems of LANs are being replaced by e-mail client software and server software used on the public Internet, creating popular, broad-based standards that make e-mail easier for everyone.

Internet e-mail uses two main standards, the simple mail transfer protocol, or SMTP, to send messages, and the post office protocol, or POP, to receive messages. Because these standards are universal, the software sending and receiving e-mail, as well as the servers that handle the messages, can run on a variety of normally incompatible computers and operating systems.

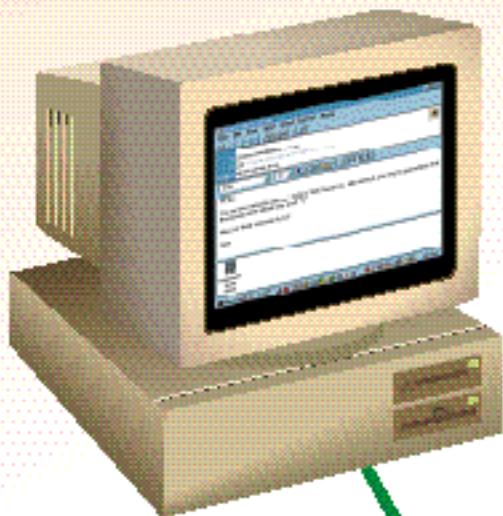
The country's largest Internet service provider, America Online, doesn't use SMTP or POP, but instead makes use of its own proprietary protocols to send and receive e-mail. The reason users of AOL can communicate with people outside AOL is that America Online uses gateway software that translates between different e-mail protocols.

The body of an e-mail message is where you type your note. But e-mail has moved beyond mere words to encompass the ability to enclose complex document files, graphics, sound, and video. You can add these types of data to the body of a message by enclosing or attaching a file. When you enclose a file, your e-mail software encodes it, turning all the multimedia data into ASCII text. At the receiving end, the user's software turns this ASCII gibberish back into meaningful data—if the software at both the user and the sender ends uses the same encoding scheme. The most popular of these is MIME, the multipurpose Internet mail extensions. MIME doesn't care what type of data is being enclosed with a message.

Another set of e-mail standards can be seen as a great enabler of communication or an annoying source of inundating junk mail. Electronic mailing lists automatically send messages to a large number of users, either on a one-time basis or on a regular schedule. A mail reflector is server software that distributes mail to members of a mailing list. With a list server, individuals send e-mail messages to subscribe and unsubscribe to the mailing list just as one would subscribe to a newsletter or magazine. And somewhere in the future, you can expect artificial intelligence agents that, like a trained assistant, will go through your e-mail before you do, tagging the messages you really want to see and tossing unwanted junk mail into the Recycle Bin.

# Sending E-Mail

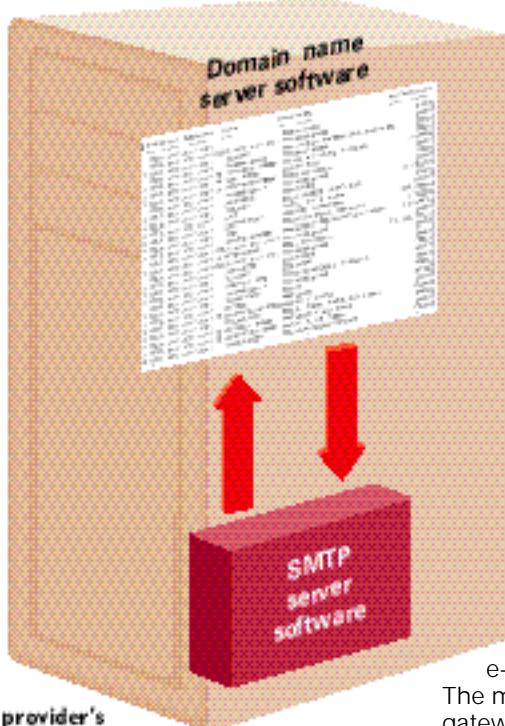
**1** Using e-mail client software, Jane creates a message to go to Bob. She also attaches a digitized photograph of herself, which is encoded using a standard algorithm, such as MIME, uuencode, or BINHEX. Just as easily, Jane could enclose a word processing document, spreadsheet, or program.



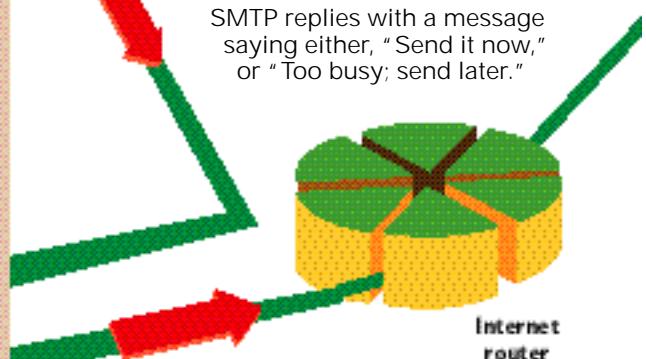
**2** The encoding turns the data making up the photograph into ASCII text, which computers commonly use for unformatted, simple text. The e-mail software may also compress the enclosure before attaching it to the message so it takes less time to send.



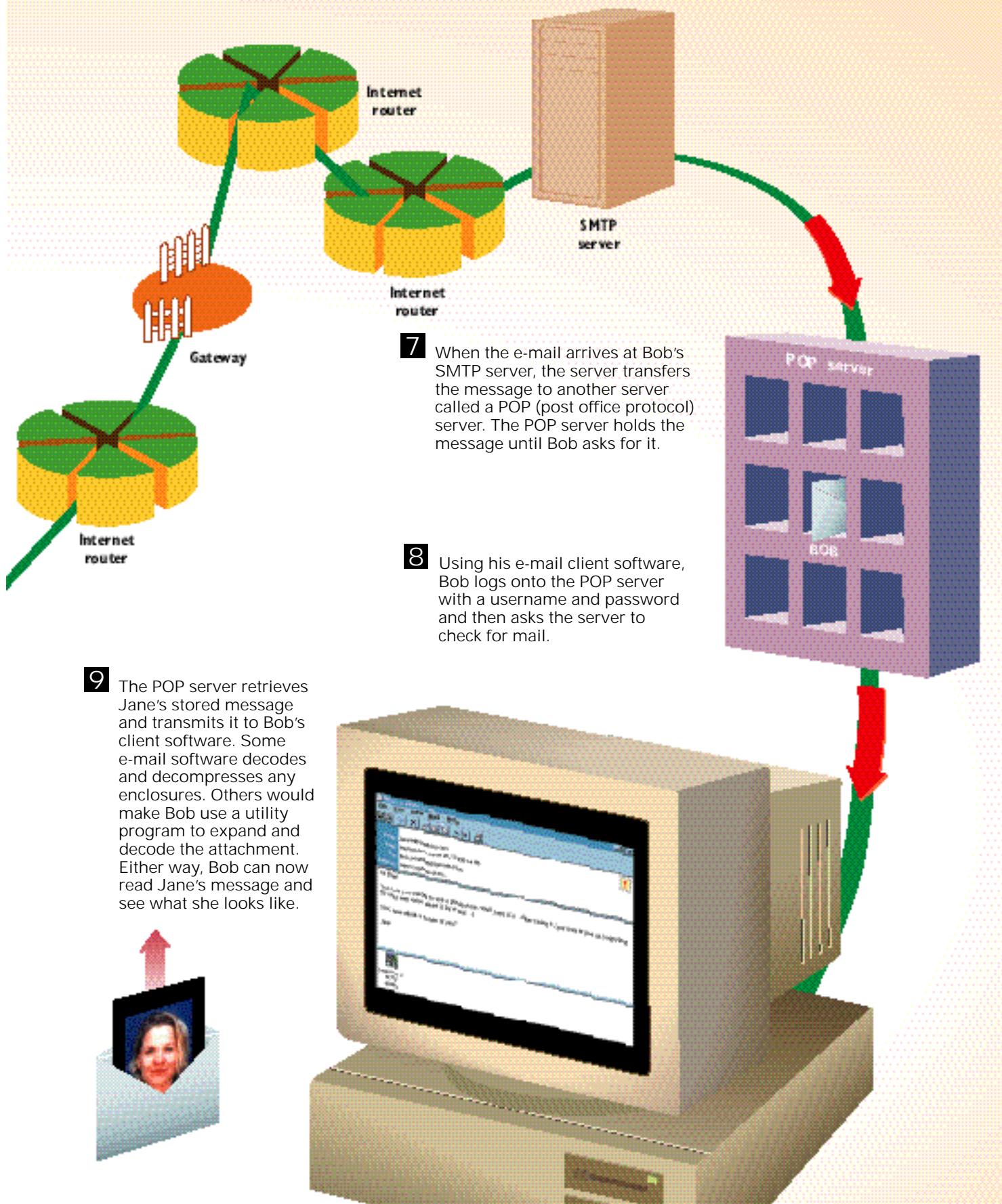
**4** The client sends the message to the SMTP server and asks for confirmation. The server confirms that it has received the message.



**3** The client software contacts the Internet service provider's computer server over a modem or network connection. The client software connects to a piece of software called an SMTP server, short for simple mail transfer protocol. The server acknowledges that it has been contacted, and the client tells the server it has a message to be sent to a certain address. The SMTP replies with a message saying either, "Send it now," or "Too busy; send later."



**6** After the SMTP sends the message, the mail travels through various Internet routers. Routers decide which electronic pathway to send the e-mail along based on how busy the routes are. The message may also pass through one or more gateways, which translates the data from one type of computer system—such as Windows, UNIX, and Macintosh—to the type of computer system that's the next pass-through point on the route.



CHAPTER

35

# How Internet Video and Audio Work



**TODAY** you use the World Wide Web to listen to Mozart and watch Super Bowl highlights, but the Internet began as a text-only medium. In fact, the Internet was a multimedia late-bloomer compared to personal computers. That's because the problem of how to handle the necessarily huge amounts of data involved in graphics, audio, and videos was easier to solve on the desktop than on the Internet.

Conveniently, there's a name for the problem: bandwidth. The term is used to describe how much data you can push across a network, a computer bus, or any of the many other data pathways that let components work with the same information. The wider the bandwidth, the better.

And luckily, computers have several ways to overcome the Internet's bandwidth limitations. The most common solution is compressing data to make multimedia files smaller. But compression used on CD-ROM or DVD doesn't work well enough for the slower Internet. So content providers also usually deliver less quality than you would get from a CD-ROM. This means reducing the frequency range of sound—the number of high and low pitches—and the number of colors and frames for video. It also means restricting the size of the window and the length of the multimedia clip.

The first audio and video on the Internet were short clips because a computer had to download them completely to its hard drive before it could play them. A newer technology called streaming extends the length of a multimedia clip from seconds to hours. Streaming enables your PC to play the file as soon as the first bytes arrive, instead of forcing the PC to wait for an entire multimedia file to finish downloading.

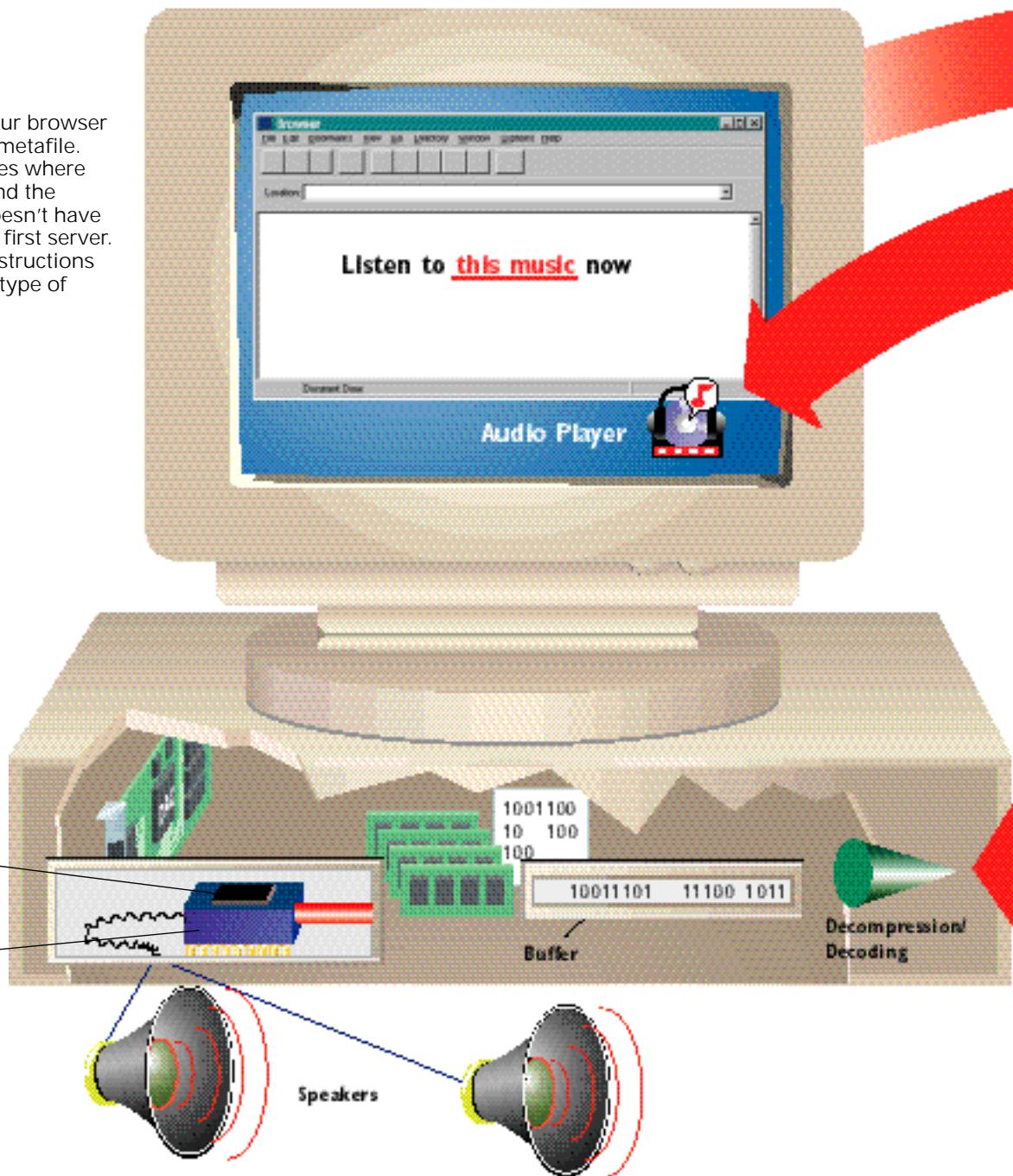
Streaming doesn't send files as other files are sent on the Internet. That is, it doesn't use the same protocol. A protocol is the rules governing how two computers connect to each other—how they break up data into packets, and synchronize sending them back and forth. Instead of the transmission control protocol (TCP) used for most Internet transactions, streaming calls on the user database protocol (UDP). The crucial difference between the two protocols is how they check for errors. If you're downloading a hot new game off the Internet and a passing electrical interference garbles one packet, TCP suspends the download while it asks the sending computer to resend the bad packet. But with audio and video, if you miss a frame or word here or there, the loss isn't crucial. You may not even notice it. But you would notice if the protocol took the extra time to enforce the retransmission. That's why UDP lets the connection lose occasional packets without fussing.

Because of streaming technology, you can listen to a live radio station located across the country, select from an archive of interviews with celebrities and scholars, or hear concerts so remote you could never attend in person. Streaming technology proves that audio and video are more than a fancy way to dress up a Web page. They add new dimensions to Internet communications.

# How Streaming Audio Works

- 1 When you click on a word or picture linked to an audio source, the Web browser contacts the Web server holding the current Web page. (See Chapter 37.)

- 2 The server sends your browser a small file called a metafile. The metafile indicates where your browser can find the sound file, which doesn't have to be located on the first server. Your PC also gets instructions on how to play that type of audio.

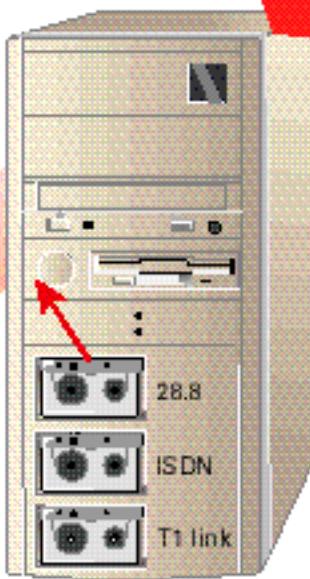




- 3 The metafile tells the Web browser to launch the appropriate audio player. The players are plug-ins, mini-programs designed to work with a particular browser such as Netscape Navigator or Microsoft Internet Explorer. Plug-ins let software developers who don't work for Netscape or Microsoft write code to expand the browsers' abilities. If you have not previously set up a particular plug-in, at this point you may have to download and install it before continuing.

- 4 The audio player contacts the audio server providing the sound file and tells this server how fast your Internet connection is.

- 6 When the packets arrive at your PC, your system decompresses and decodes them, and sends the results to a buffer, a small portion of RAM that holds a few seconds of sound.



- 5 Based on the speed of your connection, the audio server chooses one of several versions of the audio file. It sends higher-quality sound, which requires a wider bandwidth, over faster links, and lower-quality sound over slower connections. The server sends the audio files to the PC as a series of packets in user database protocol (UDP), which permits the occasional packet to get lost without critically disrupting the transmission.

- 7 When the buffer fills up, the audio player starts to process the file through your sound card, turning file data into voices, music, and sounds while the server continues to send the rest of the audio file. This process can continue for several hours. The buffer can temporarily empty if it doesn't receive enough data to replenish it. This happens if you access another Web page, or you have a poor connection, or Internet traffic is heavy. If the buffer empties, the audio replay pauses for a few seconds until your PC accumulates enough data to resume playing. If the sound source is live, the player will skip portions of the audio program. If the sound source is prerecorded, the player will continue from the point it stopped.

# How Internet Video Works



Recorded video (VHS tape)

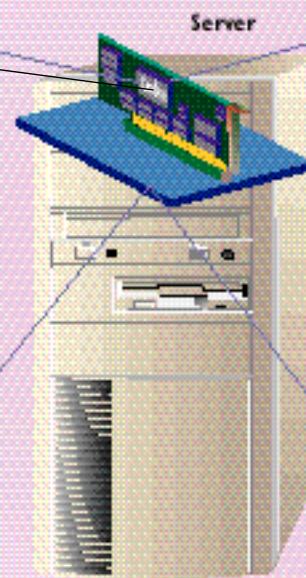
1

Inside the computer that acts as a server for Internet video, a video capture expansion card receives the ordinary analog video signal from its source, either a live feed or recorded tape. The capture board turns the analog signal into digital information at a maximum rate of 30 frames a second.

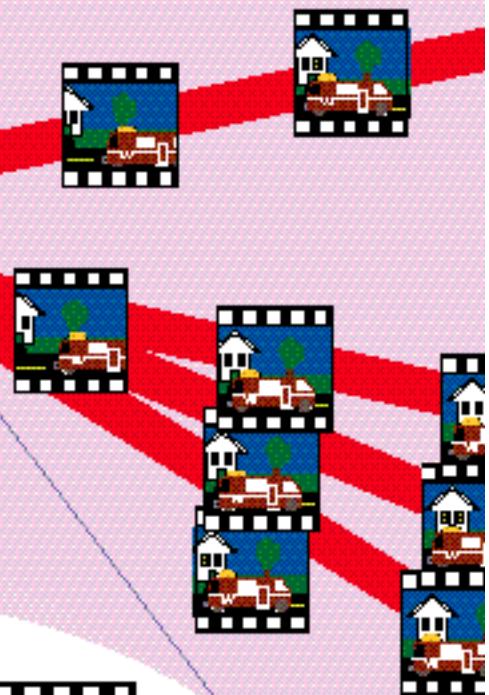


Live video (camera)

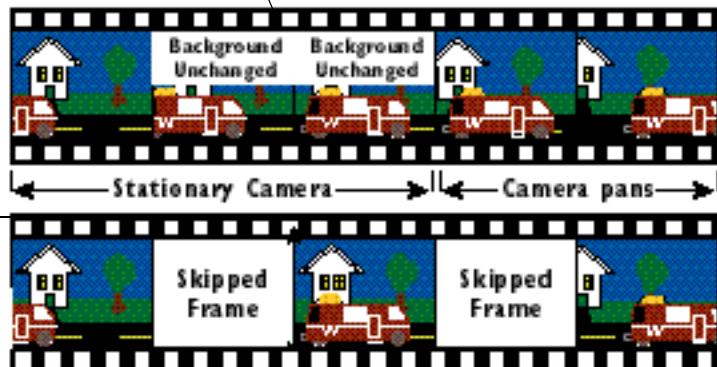
- 2 The capture board sends the digital information through a codec, a compression/decompression algorithm. Different codecs use several methods to compress the video.



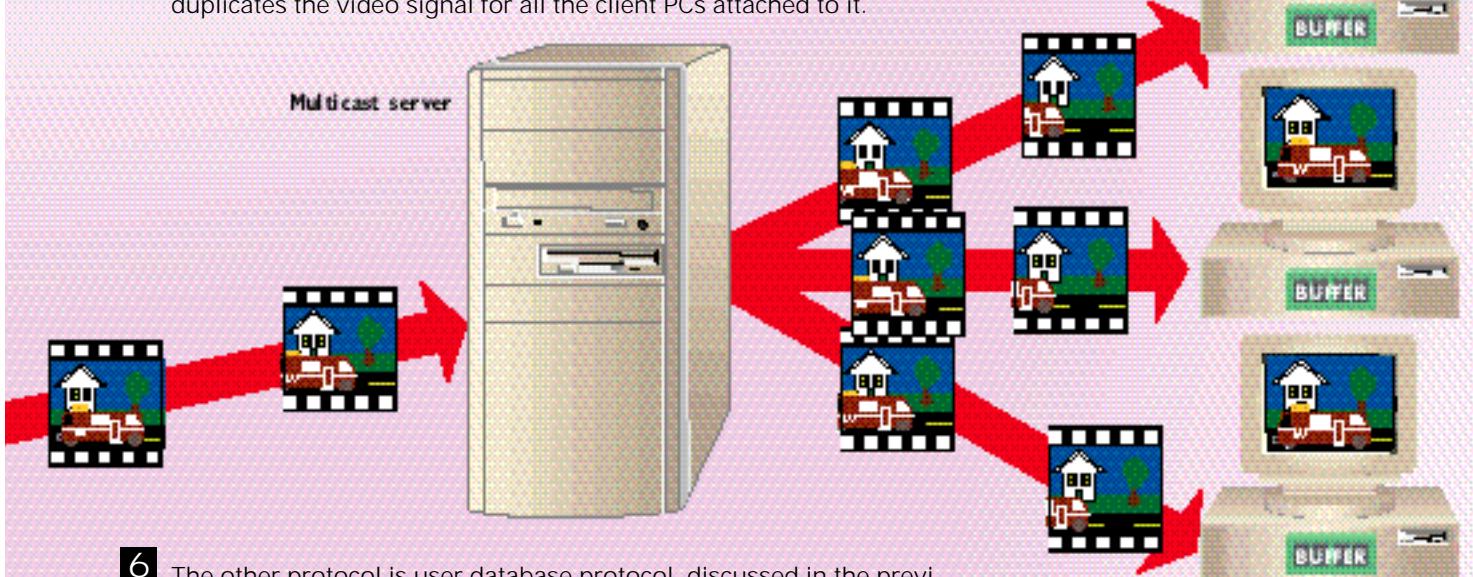
- 3 Interframe compression compares adjacent frames and transmits only those pixels that change from one to the other. When the camera is still, the background is not transmitted after a key frame has established what the background looks like. When the camera pans, causing the background to change, the entire frame is transmitted, creating another key frame. Thus, a still camera generally creates less data to transmit than a camera that is always moving.



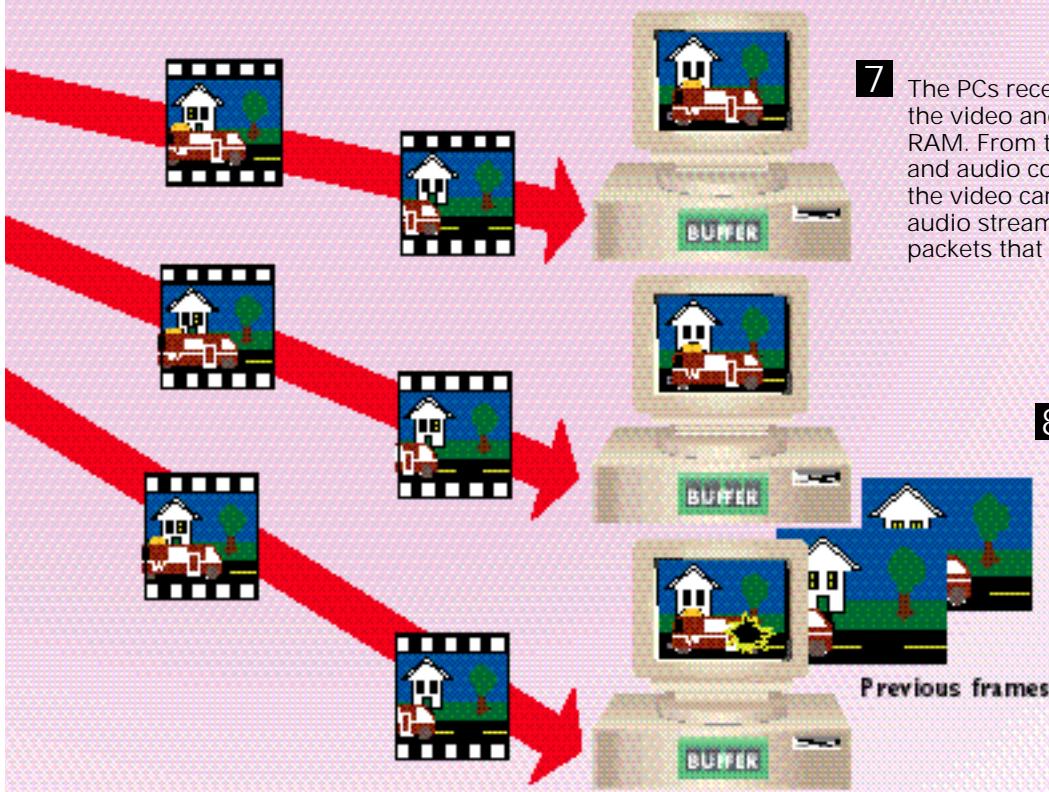
- 4 Codecs also skip frames for slower Internet links. The faster your connection, the more frames you receive, and the smoother the video is.



- 5 The server breaks up compressed video into one of two types of packets for two types of transmission protocols. One type is called IP (Internet provider) multicast packets. IP multicast uses less bandwidth, which is helpful when transmitting the same video to several people at different PCs. The video server sends a single signal to a computer acting as a multicast server, which duplicates the video signal for all the client PCs attached to it.



- 6 The other protocol is user database protocol, discussed in the previous illustration. UDP video delivery is more common because it doesn't require special network hardware, such as a multicast server. UDP packets must be sent to every client PC, which uses more bandwidth, but is more efficient in preventing gaps or pauses in the audio part of the signal.



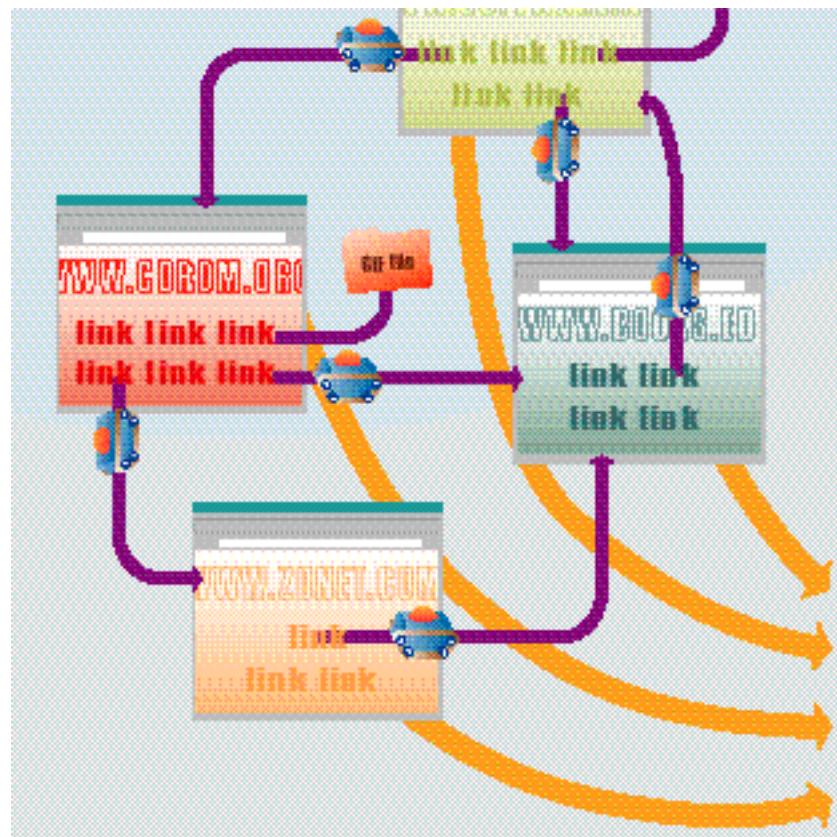
- 7 The PCs receiving the signals decompress the video and load it into a small buffer in RAM. From there, the signal splits into video and audio components, which are sent to the video card and sound card. As with pure audio streaming, video streams simply skip packets that they can't handle in real time.

- 8 But unlike audio, a corrupted video packet can cause a defect that carries over to other frames. To correct this, the software compares new frames with other ones to detect errors and correct them by using visual information from an untampered frame.

CHAPTER

36

# How the World Wide Web Works



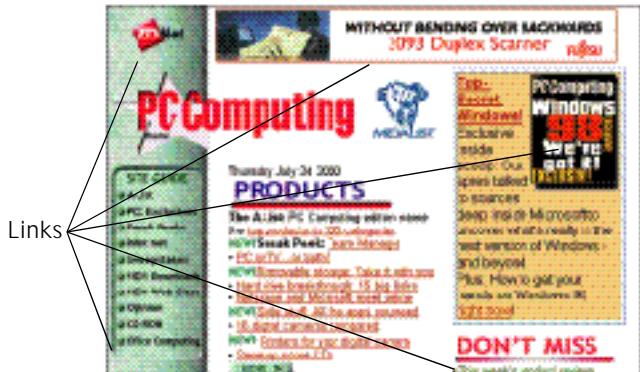
FOR now, the Internet—and its most popular manifestation, the World Wide Web—is a Model-A Ford. It gets us to where we want to go in cyberspace, and it sure is fun and a lot faster than that old horse, postal mail. But it's cantankerous, slow, and no one today can imagine what the Web may become—what enormous, pervasive effect the Internet is going to have by the end of the next century—any more than someone watching a Model-A chugging down a dirt road nearly a century ago could imagine the importance of the automobile to modern society.

It has already had an impact on education, business, and personal communications. Ultimately, it has the ability to finally fulfill the promise of Gutenberg—every person can be a publisher. Anyone can present ideas to the world without owning expensive presses, having a publisher, or, for that matter, having anything worth saying. The Internet also could make us a species of potato—anemic creatures with flabby muscles and weak bones who never leave their houses because everything, from a job to worship to shopping to socializing to sex, will be available on the Web. Actually, chances are it's going to influence us in ways we've yet to imagine. I doubt that first Model-A inspired ideas of drive-in movies and take-out food. Those whose main concern at the time was what effect the car would have on the buggywhip business weren't even close.

So I'm not going to predict either utopian or dystopian worlds forged by the World Wide Web. But it doesn't take any insight to see that it's one of those quantum leaps that will change society irrevocably.

But for right now, here's what the World Wide Web is: a fast, cheap way to send messages and even get around paying for long-distance calls. It's a great way to find information to settle a bet, do a term paper, or psyche out your competitors. You can get lots of nifty software from the Web either free or nearly so. You can meet and even become friends with people you will never see in person your entire life. Just those few things ain't bad. And in this chapter, we're going to look at two of the most common uses of the Web—browsing and searching. The two sound quite a bit alike. But as we'll see, browsing often is a casual, sequential activity, the Net equivalent of channel surfing, full of serendipitous surprises. Searching is pulling out all the stops, earnest, systematic, with the capabilities of hundreds of assistants.

# How a Web Browser Works



1

One way to begin a jump through the cyberspace of the World Wide Web is to click on a hyperlink. A hyperlink is a text phrase or a graphic that conceals the address of a site on the Web. A site is a collection of files, documents, and graphics that someone has made generally available to others through the Internet. When the hyperlink is text, it is underlined and in a different color. The text doesn't have to be the actual address. [These three words](#) could hide a link to the home page of ZDNet that actually looks like this: <http://www5.zdnet.com/>.

2

Another way to direct the browser to a site is to type its universal resource locator (URL) into the address space on your browser's toolbar.

For example, typing in

<http://www.cdron.com> aims your browser at my own Web site, where I have a database of CD-ROM reviews. Each part of the URL means something.

Address <http://www.cdron.com>

http identifies the site as one on the World Wide Web using HTML, or hypertext markup language. If the part before the slashes is ftp, that means the site is one that uses file transfer protocol; ftp exists primarily for plain text listings of files available for downloading, unadorned by the graphics and pizzazz of a Web page.

:// alerts your browser that the next words will be the actual URL, which is broken up by periods. Each period is usually referred to as dot.

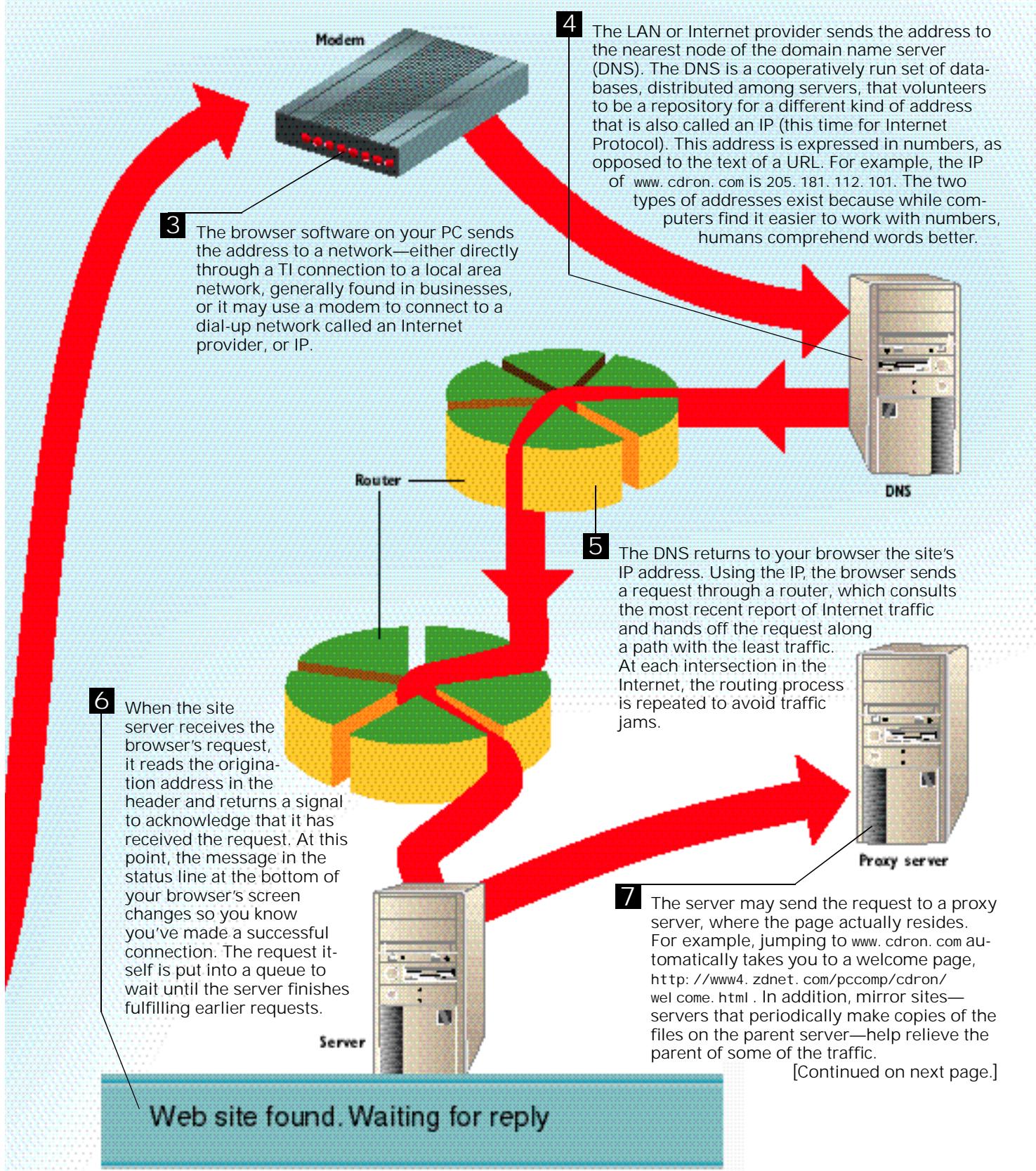
www identifies the site as part of the World Wide Web. The Web is a subset of the Internet that uses text, animation, graphics, sound, and video.

cdron is the domain name. This is a unique name that must be registered with a company called Network Solutions, which has exclusive authority to register domain names under an agreement with the National Science Foundation.

com is the top-level domain name. In the United States, it indicates the purpose of the sponsors of the site. For example, com says that cdron is a commercial operation. Other top-level names include edu for schools, gov for government offices, and the all-purpose org for organizations. Outside the United States, the top-level name may refer to a country, such as uk for United Kingdom.

welcome.html is a specific page at the site, and the html tells the browser that the page uses the hypertext markup language—simple codes that determine the page's onscreen look.

<http://www.cdron.com/welcome.html>



# How a Web Browser Works

(Continued)

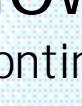
**8** Stored on the server, the Web page itself consists of an HTML text file. HTML is a collection of codes that control the formatting of text in the file.

**9** The codes may also include the URLs of graphics, videos, and sound files that exist elsewhere on the server or on a different site entirely.

**10** When the server is free to respond to the browser request, it sends the HTML document back over the Internet to your browser's Internet provider address. The route it uses to get to your PC may differ vastly from the route your request followed to reach the server.



```

<HTML>
<HEAD>
<TITLE>CD-ROM Highlights</TITLE>
<META NAME="FIND_COPY_DATE" CONTENT="19970711">
<HEAD>
<BODY BGCOLOR="#FFFFFF" BACKGROUND="http://www.pbs.org/itvs/nature/01.htm">
<table width="500" border=0>
<tr>
<td width="150" rowspan="2" style="vertical-align: top; padding-right: 10px;">


<td style="padding-left: 10px; font-family: sans-serif; font-size: 10pt; color: #000000; background-color: #FFFFCC; border-bottom: 1px solid black; height: 15px; margin-bottom: 5px; vertical-align: bottom; text-align: right; padding-right: 5px; font-weight: bold; letter-spacing: 1px;">CD-ROM HIGHLIGHTS

<tr>
<td style="padding-left: 10px; font-family: sans-serif; font-size: 8pt; color: #000000; background-color: #FFFFCC; border-top: 1px solid black; border-bottom: 1px solid black; height: 15px; vertical-align: top; padding-top: 5px; padding-bottom: 5px;">The CD-ROM version of the show includes a CD-ROM with all the highlights of the series. It includes a CD-ROM player, a menu, and a search function. To play the CD-ROM, click on the "CD-ROM" button on the menu. To search for a specific topic, type it into the search bar and click "Search".

<tr>
<td colspan="2" style="text-align: center; padding-top: 10px; padding-bottom: 10px; font-size: 10pt; color: #000000; font-weight: bold; letter-spacing: 1px;">Site Outline

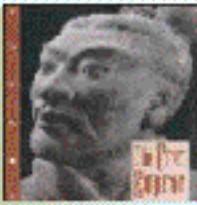
<tr>
<td colspan="2" style="text-align: center; padding-top: 10px; padding-bottom: 10px; font-size: 10pt; color: #000000; font-weight: bold; letter-spacing: 1px;">VALIDATE



```

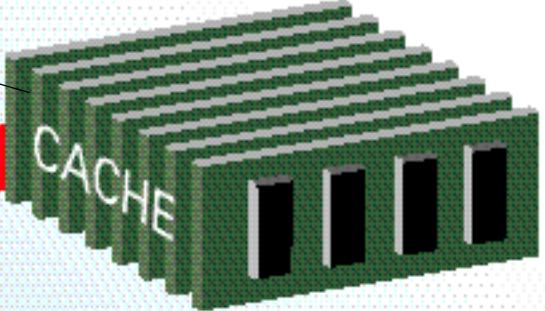
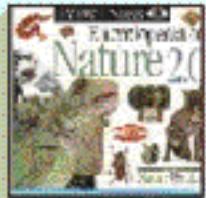
**11** At the same time, the server sends instructions to the sites that contain the graphics, sound, and video files identified in the page's HTML coding, telling those sites to send those files to your PC.





12

As the different parts of the page arrive at your PC, they are stored in a cache, a combination staging area and reservoir in the computer's RAM. If later, your browser requests the same page or any of the elements on the page, such as a graphic, the browser retrieves it from the cache rather than going back over the Internet to the original sources.

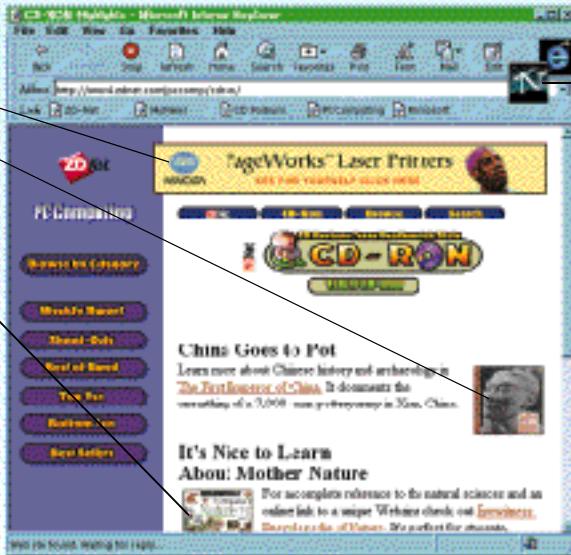


13

If a sound, music, or video file, such as a wave, MIDI, or AVI recording, is part of the page, the browser waits until all of the file has arrived in the cache, and then it feeds the wave file to Windows's Media Player, which uses your sound card to reproduce the sound.

14

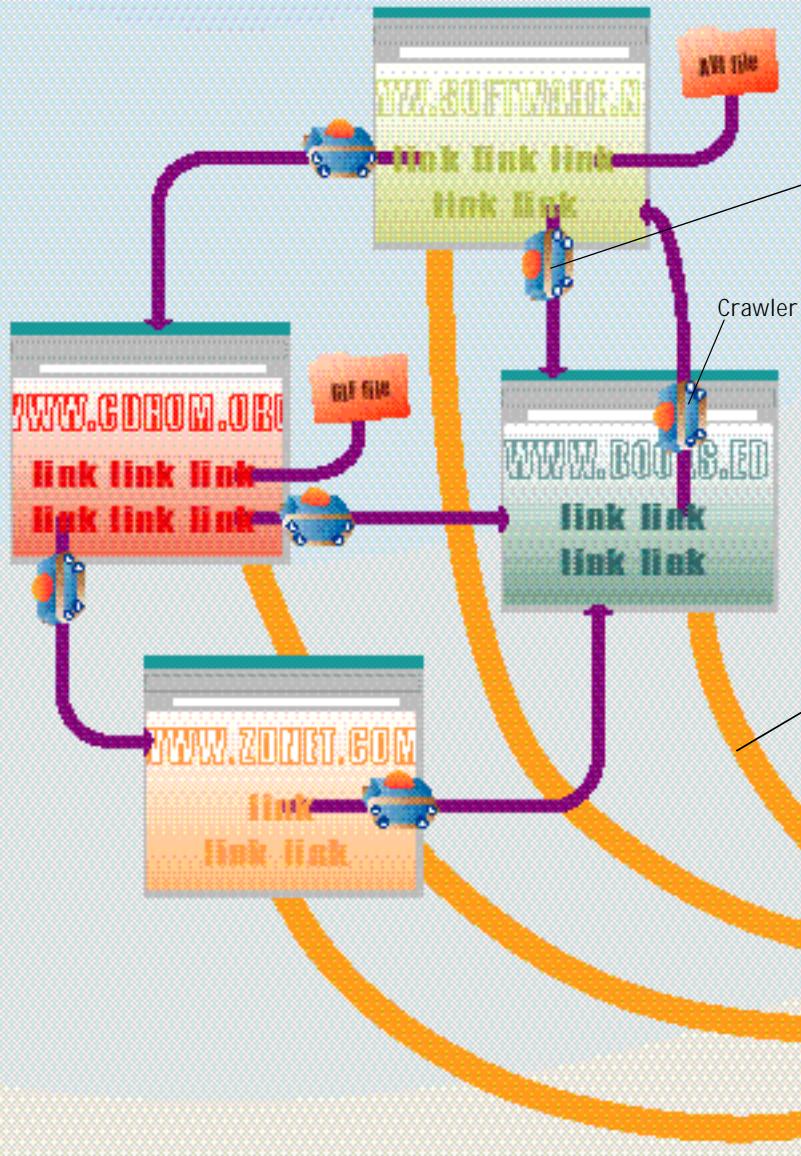
Meanwhile, the browser begins using the elements in the cache to reassemble the Web page onscreen, following the hidden HTML codes in the main document to determine where to place text, graphics, or videos onscreen. Because not all portions of the Web page arrive at your PC at the same time, different parts of it appear onscreen before others. Text, which is the simplest element to send, usually appears first, followed by still and animated graphics, sounds and music, and videos.



15

Icons in the upper-right corners of both Netscape Navigator and Microsoft Internet Explorer are animated while parts of the page are still being received. When all elements have arrived and have been added onscreen, the animations freeze into still images, telling you that the entire Web page has now been successfully retrieved by the browser.

# How a Web Search Engine Works

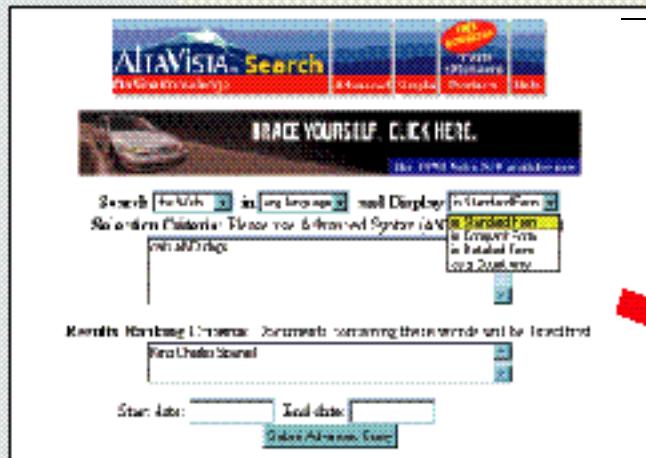


1 A Web site that includes a search engine, such as Alta Vista, Yahoo!, or Excite, regularly runs programs called Web crawlers, or spiders, to gather information about what's available on the Internet. The crawlers travel across the World Wide Web by following hypertext links they encounter. Some Web crawlers follow every link they find. Others follow only certain types of links, ignoring those that lead to graphics, sound, and animation files.

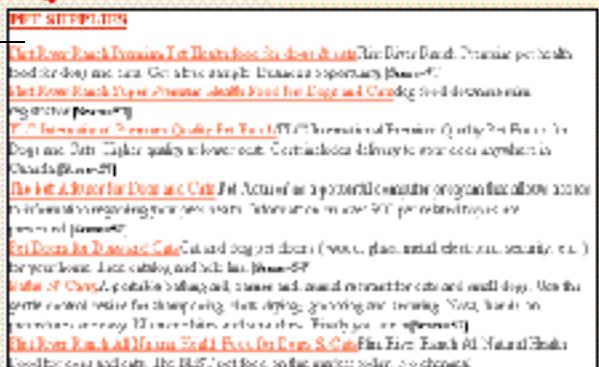
2 When a crawler encounters a document, it sends the address of the document—its universal resource locator, or URL—along with the text from the document, back to the search engine's indexing software.

INDEXING SOFTWARE			
Searching for #3 AND #4 Press any key to interrupt processing			
Set number 6 resulted in 250 references found in 88 classes			
Comma/HELP/ME [New 2]: c			
Find Set number [14] to Comma/HELP/ME [CAMEI]: 5			
And/Or/Mate/All/Cancel [New 3]: or			
New Set number [16] to Comma/HELP/ME [CANCEI]: 6			
And/Or/Mate/All/Cancel [New 4]:			
Searching for #5 AND #6 Press any key to interrupt processing			
Set number 3 resulted in 348 references found in 110 classes			
Comma/HELP/ME [New 5]:			
SET 16 MILES	LOCATION	SEARCH HITS	SET CATEGORIES
1	511	2209	Software - Arts
2	3180	14448	Abuse - Corporations
3	1021	4421	Abuse - Government
4	45466	10672	Software - Sports
5	31	135	#1 AND #2
6	88	889	#3 AND #4
7	118	246	#5 OR #6

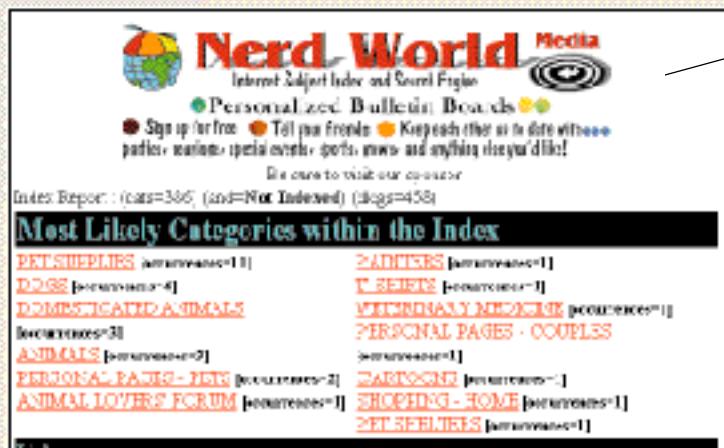
3 The indexing software extracts information from the documents, organizes it into a database, and, based on the frequency of different words found in the document, indexes the information. Like the index in the back of this book, a Web index makes it quicker to locate a specific piece of information. The type of information indexed depends on the search engine. It could include every word in each document; the most frequently used words; words in its title, headings, and subheadings; and the document's size and date of creation.



- 4 When you visit the search engine's Web page, you launch a search of its database by typing in keywords that describe the information you're looking for. Some search engines allow you to use Boolean qualifiers to restrict or widen the results of a search. "Cats AND dogs," for example, would find Web pages that include both "cats" and "dogs." The keywords "cats OR dogs" find pages that include either one of the words, but not necessarily both. When you click the search button, your search request is sent to the database.



- 5** The search engine does not go directly to the Web to search for your keywords. Instead, it looks for the keywords in the index for the database it has already constructed. The search engine then creates a new HTML page on-the-fly, listing the results of your search. Most search engines display the URLs and titles of the documents. Some also list the first few sentences of the documents as well, and some rank the sites for relevancy according to how closely they match your request.



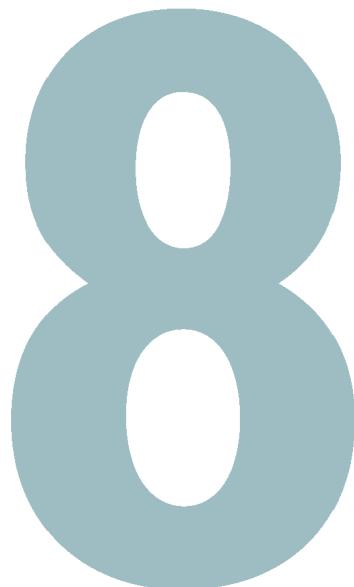
- 6** The results page comes complete with hypertext links for the Web pages contained in the database. To go to the actual page, you merely have to click on the link.

## Tricking Search Engines

Some documents include metatext, words that don't appear onscreen but that are designed to trick the indexing software into giving a site a higher relevance rating in a search. For example, by repeating the word "candy" 100 times, the document is more likely to turn up in a search that includes "candy" as a keyword. Better search engines are aware of this trick and make the appropriate adjustments to give the page a more competitive relevancy rating.



P A R T



# HOW PRINTERS WORK

Chapter 37: How Fonts Work

278

Chapter 38: How a Dot-Matrix Printer Works

284

Chapter 39: How a Laser Printer Works

288

Chapter 40: How Color Printing Works

292

**I**N the early days of personal computers, someone came up with the idea that all this computerized data would lead to the “paperless office.” We’re on our second decade in the personal computer revolution and more trees than ever are giving their lives to produce hard copies of everything from company budgets complete with full-color graphs to homemade greeting cards. Not only are we creating more printouts than ever before, but computer printing has turned into a fine art. The very essence of a whole new category of software—desktop publishing—is the accomplishment of better and better printed pages.

Whoever made that erroneous prediction about a paperless office missed an important fact. That person was probably thinking about how offices used paper in the age of the typewriter. Then there wasn’t much you could put on paper except black letters and numbers—most often in an efficient but drab typeface called Courier. If all those ugly memos and letters had been replaced by electronic mail, the world would not have suffered a great loss. But what forecasters didn’t see is that software and printing technology would make possible fast, easy, colorful hard-copy of reports, photographs, newsletters, graphs, and, yes, company budgets and greeting cards. Even IBM’s best Selectric could never come close to producing these things.



Speed and ease were the first improvements in printing. Where a simple typo on a typewriter might just be whited out or hand-corrected with a pen, today—because of the speed of printers—it’s easier just to correct a mistake on screen and print a fresh, flawless copy.

Graphics were the next big advance. The day of the all-text document ended with the first software that could print even the crudest line graph on a dot-matrix printer. Now anything that’s visual, from line art to a photograph, can be printed on a standard office printer.

Today, color is the current frontier being conquered with office printers. The quality and speed of color printers is increasing as their cost is decreasing. Because they can double as black-and-white printers, you can expect to see a color printer become more and more the only printer in the office and home.

And paper hasn’t disappeared from the office. Instead, it’s taken on a whole new importance. The lowly printer that used to turn out crude approximations of characters is now one of the most important components of a computer system.

## KEY CONCEPTS

**Ben Day dots** The individual dots of ink that make up a printed graphic.

**bitmapped font** Characters created by using records of where each and every dot of ink is placed in a dot matrix. Sometimes referred to as raster fonts, they are most often used in impact and low-end printers. The use of bitmapped fonts is decreasing because of the versatility of outline fonts used in Windows.

**CYMK** Cyan, yellow, magenta, and black (K)—the four colors most often used in color printing.

**dithering** A process in which the frequency and placement of ink are used to create shades of gray and hues of color.

**dot matrix** The grid of horizontal and vertical dots that make up all the possible dots—most often as many as 900,000—that could be included in the bitmap of a single character. Don't confuse dot matrix with impact printers. An ink-jet printer also creates bitmapped letters using a matrix.

**font/typeface** A typeface is a design for the alphabet distinguished by its use of such elements as serifs, boldness, and shape. Times Roman, Helvetica, and Courier are typeface. So a font is a typeface of a particular size and a particular variation, such as italic. Courier 12-point bold and Courier 12-point italic are different fonts in the Courier typeface family.

**impact printer** A printer that uses a quick blow to press ink from a ribbon onto paper.

**ink-jet printer** A printer that forms graphics and text by ejecting tiny dots of ink onto paper.

**page description language (PDL)** A software language used with printers to control complex and sophisticated print jobs. PostScript and TrueType are the two most common page description languages.

**point** 1/72 of an inch, a traditional measure for typefaces.

**print head** The mechanism that actually transfers ink from the printer to paper.

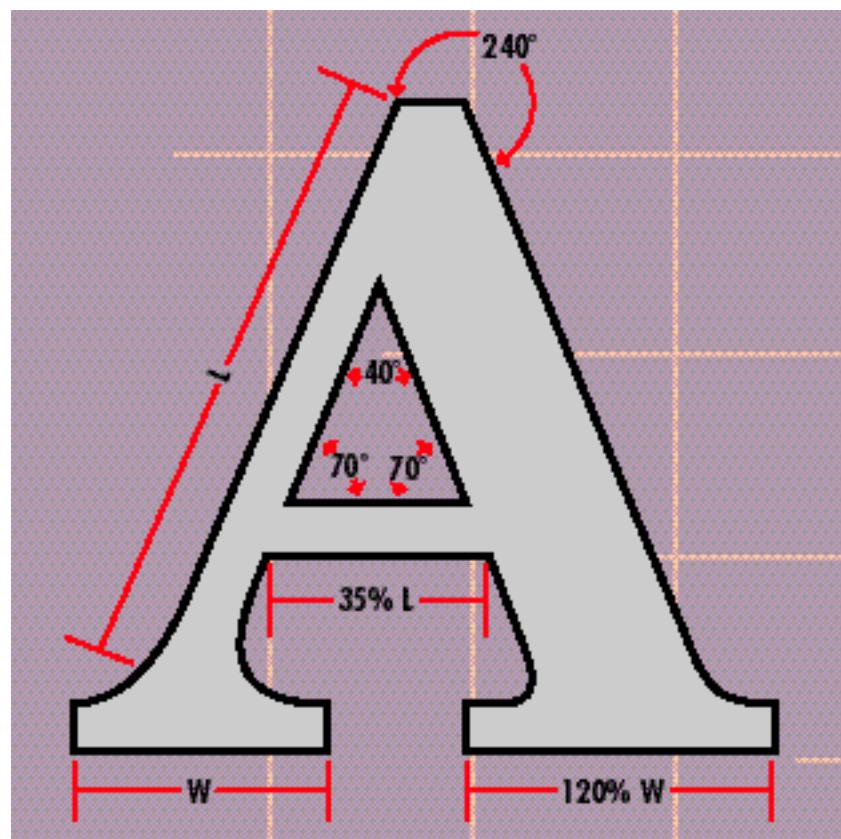
**resolution** The quality of the text and images on hard copy are dependent largely on the resolution of the printer, which is determined by the number of dots of ink it would take to make a one-inch line. 300 dpi (dots per inch) is most common, although 600 and even 1,200 dpi printers are becoming more commonplace.



CHAPTER

37

# How Fonts Work



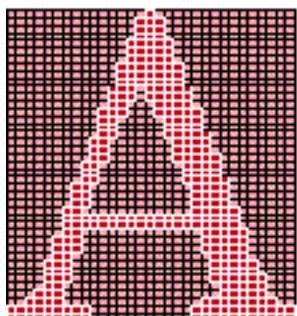
**All** printers, whether dot-matrix, ink-jet, laser, dye sublimation, or solid ink, accomplish essentially the same task: They create a pattern of dots on a sheet of paper. The dots may be sized differently or composed of different inks that are transferred to the paper by different means, but all the images for text and graphics are made up of dots. The smaller the dots, the more attractive the printout.

Regardless of how the dots are created, there must be a common method for determining where to place the dots. The most common schemes are bitmaps and outline fonts. Bitmapped fonts come in predefined sizes and weights. Outline fonts can be scaled and given special attributes, such as boldfacing and underlining. Each method has its advantages and disadvantages, depending on what type of output you want.

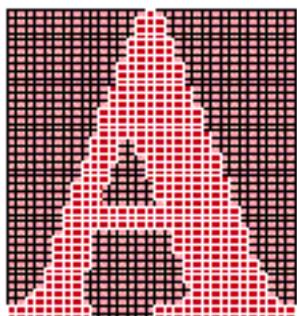
Bitmapped images are generally limited to text and are a fast way to produce a printed page that uses only a few type fonts. If the hard copy is to include a graphic image in addition to bitmapped text, then to create the graphic, your software must be able to send the printer instructions that it will understand. Printers still come with a few bitmapped fonts stored on a chip, but the fonts of choice are increasingly outline fonts.

Outline, or vector fonts, are used with a page description language (PDL), such as Adobe PostScript or Microsoft TrueType. A PDL treats everything on a page—even text—as a graphic. The text and graphics used by the software are converted to a series of commands that the printer's page description language uses to determine where each dot is to be placed on a page. Page description languages generally are slower at producing hard copy, but they are more versatile at producing different sizes of type with different attributes or special effects, and they create more attractive results. Outline fonts don't need to be a part of a printer. They can be stored on a PC's hard drive and used by the page description language as needed. With the increase in computing horsepower that makes the demands of outline fonts less of a consideration and with the spreading use of the Windows interface, outline fonts are becoming the rule where they used to be the exception.

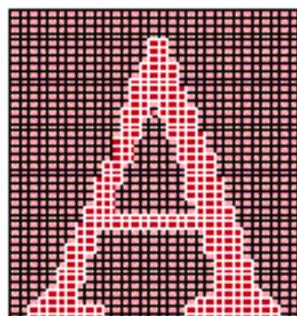
# Bitmapped Fonts



36 pt. medium

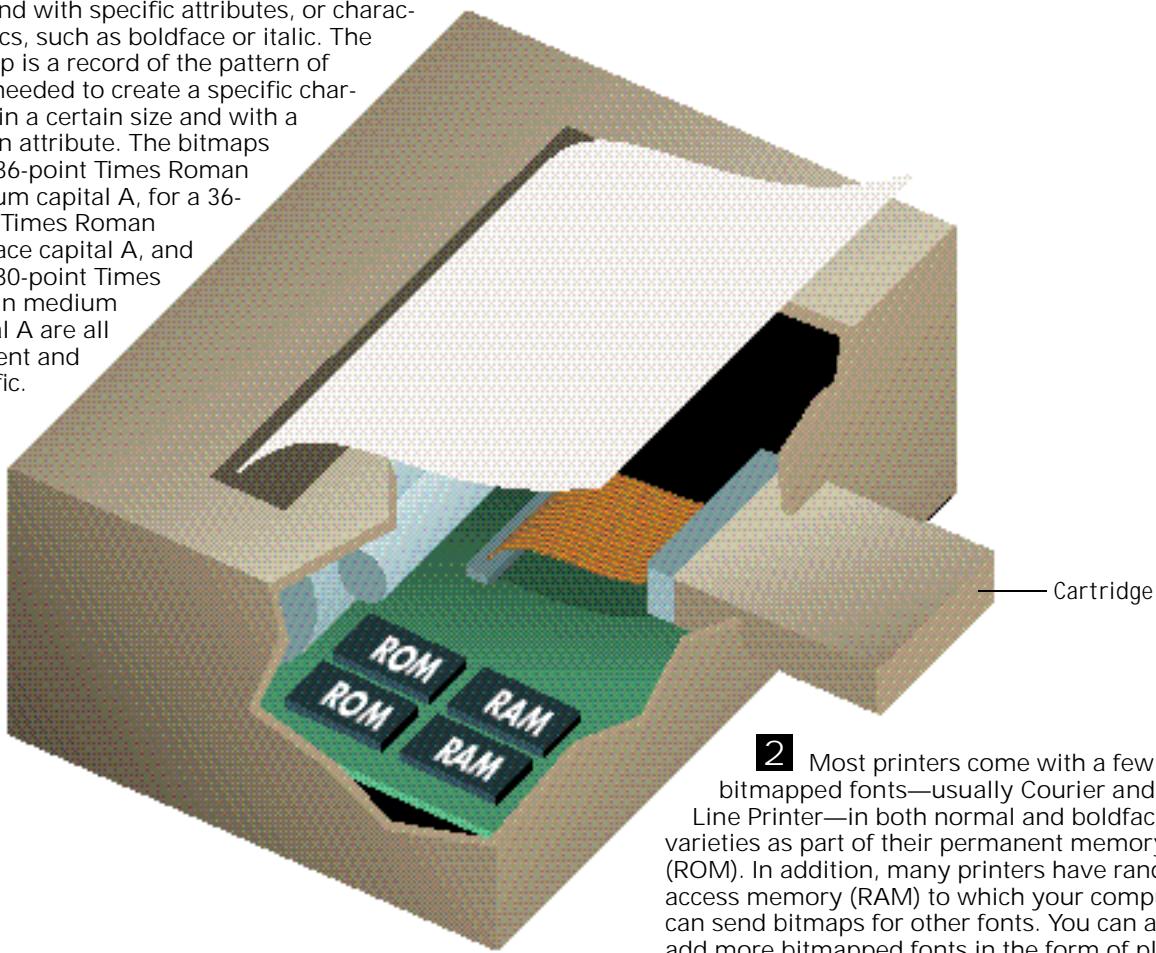


36 pt. bold

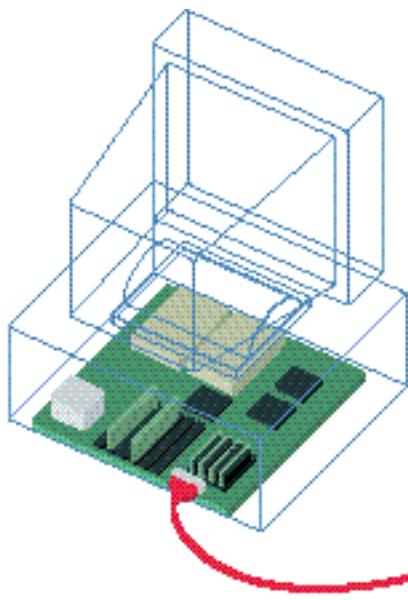


30 pt. medium

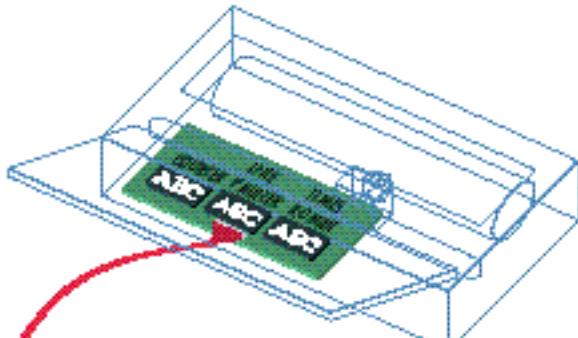
- 1** Bitmapped fonts are typefaces of a specific size and with specific attributes, or characteristics, such as boldface or italic. The bitmap is a record of the pattern of dots needed to create a specific character in a certain size and with a certain attribute. The bitmaps for a 36-point Times Roman medium capital A, for a 36-point Times Roman boldface capital A, and for a 30-point Times Roman medium capital A are all different and specific.



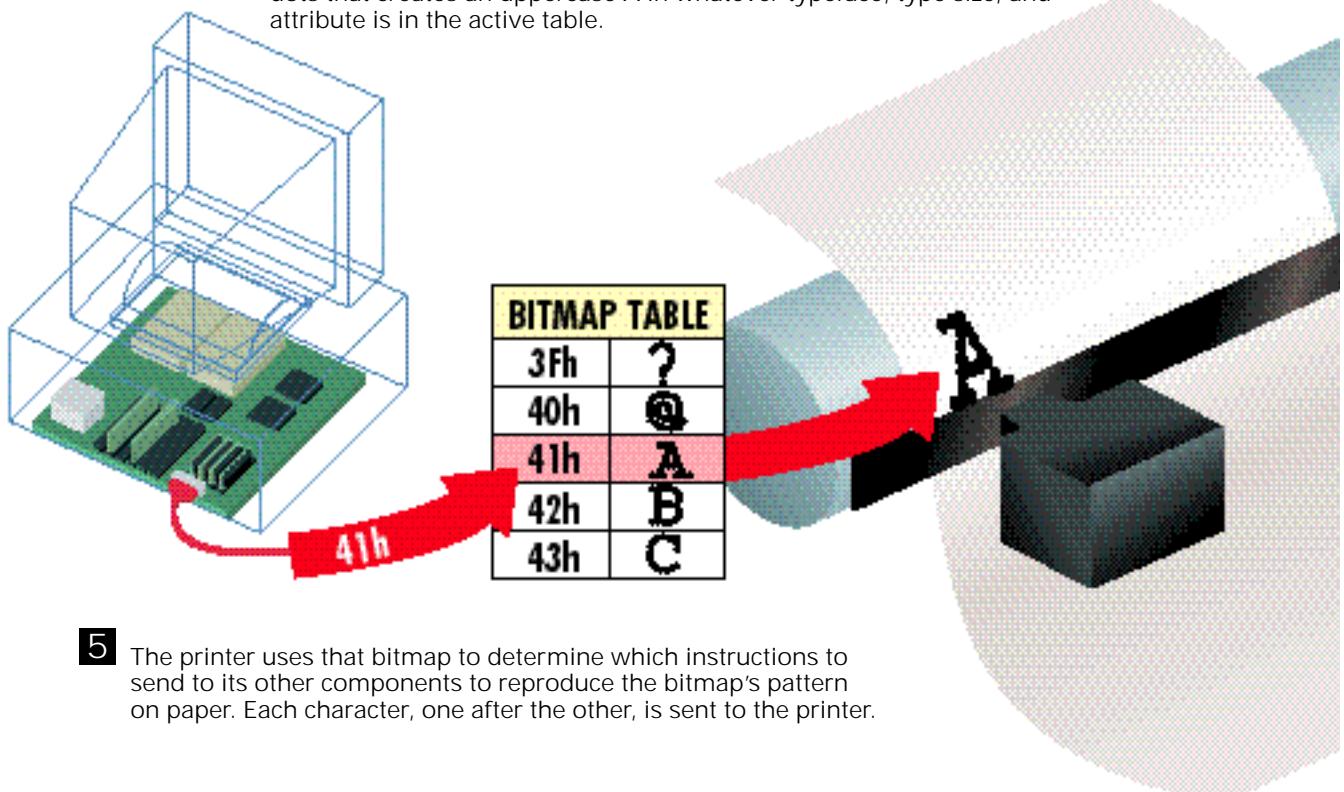
- 2** Most printers come with a few bitmapped fonts—usually Courier and Line Printer—in both normal and boldface varieties as part of their permanent memory (ROM). In addition, many printers have random access memory (RAM) to which your computer can send bitmaps for other fonts. You can also add more bitmapped fonts in the form of plug-in cartridges used by many laser printers.



- 3** When you issue a print command—either from your operating system or from within your application software—to a printer using bitmapped fonts, your PC first tells the printer which of the bitmap tables contained in the printer's memory it should use.



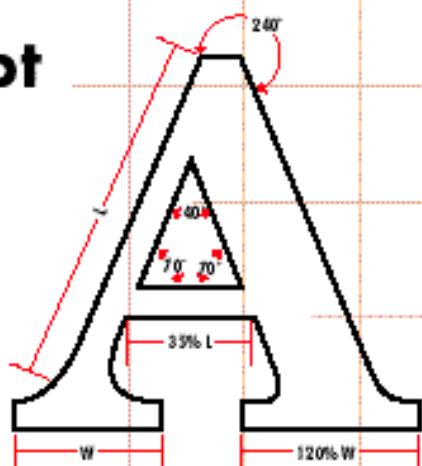
- 4** Then for each letter, punctuation mark, or paper movement—such as a tab or carriage return—that the software wants the printer to create, the PC sends an ASCII code. ASCII codes consist of hexadecimal numbers that are matched against the table of bitmaps. (Hexadecimal numbers have a base of 16—1, 2, 3, 4, 5, 6, 7, 8, 9, 0, A, B, C, D, E, F—instead of the base 10 used by decimal numbers.) If, for example, the hexadecimal number 41 (65 decimal) is sent to the printer, the printer's processor looks up 41h in its table and finds that it corresponds to a pattern of dots that creates an uppercase A in whatever typeface, type size, and attribute is in the active table.



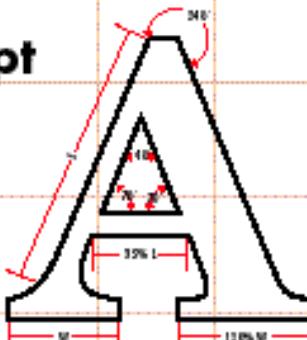
- 5** The printer uses that bitmap to determine which instructions to send to its other components to reproduce the bitmap's pattern on paper. Each character, one after the other, is sent to the printer.

# Outline Fonts

**36pt**

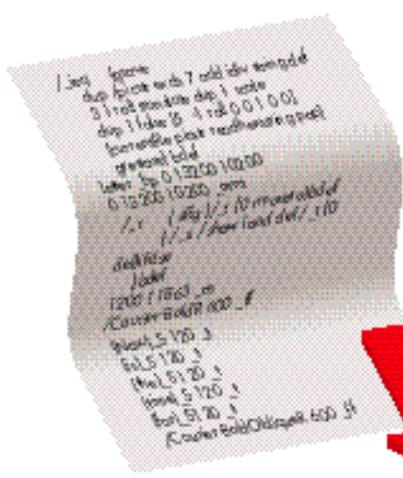


**24pt**



1

Outline fonts, unlike bitmapped fonts, are not limited to specific sizes and attributes of a typeface. Instead, they consist of mathematical descriptions of each character and punctuation mark in a typeface. They are called outline fonts because the outline of a Times Roman 36-point capital A is proportionally the same as that of a 24-point Times Roman capital A.

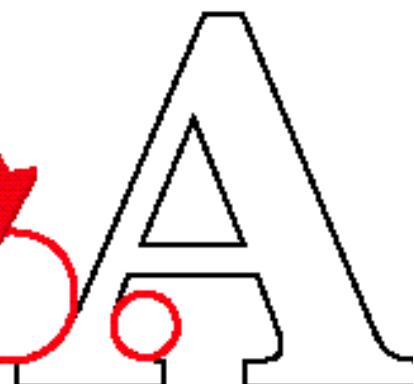
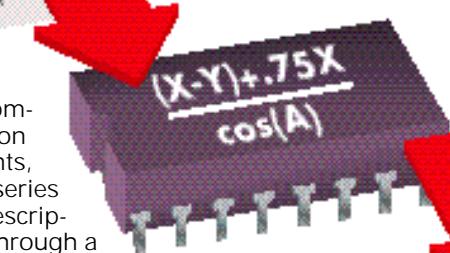


2

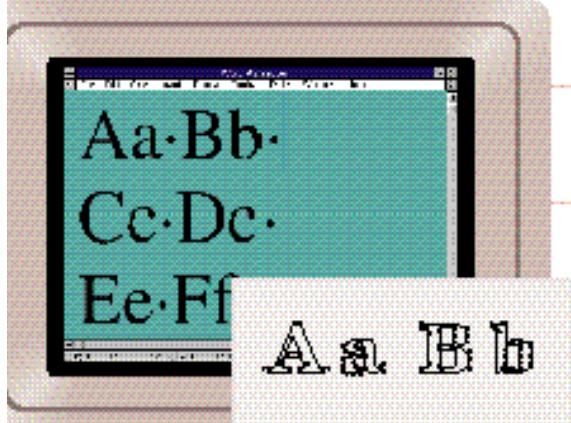
Some printers come with a page description language, most commonly PostScript or Hewlett-Packard Printer Command Language, in firmware—a computer program contained on a microchip. The language can translate outline font commands from your PC's software into the instructions the printer needs to control where it places dots on a sheet of paper. For printers that don't have a built-in page description language, Windows printer drivers translate the printer language commands into the instructions the printer needs.

3

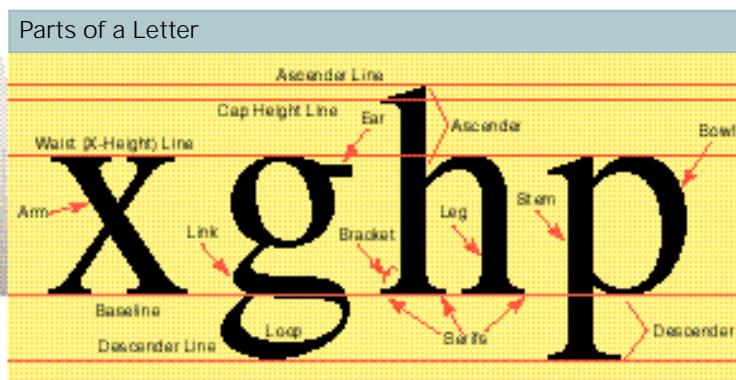
When you issue a print command from your application software using outline fonts, your application sends a series of commands the page description language interprets through a set of algorithms, or mathematical formulas. The algorithms describe the lines and arcs that make up the characters in a typeface. The algorithms for some typefaces include hints, special alterations to the details of the outline if the type is to be either extremely big or extremely small.



- 4 The commands insert variables into the formulas to change the size or attributes of the outline font. The results are commands to the printer that say, in effect, "Create a horizontal line 3 points wide, which begins 60 points from the bottom and 20 points to the right." The page description language turns on all the bits that fall inside the outline of the letter—unless the font includes some special shading effect within the outline.



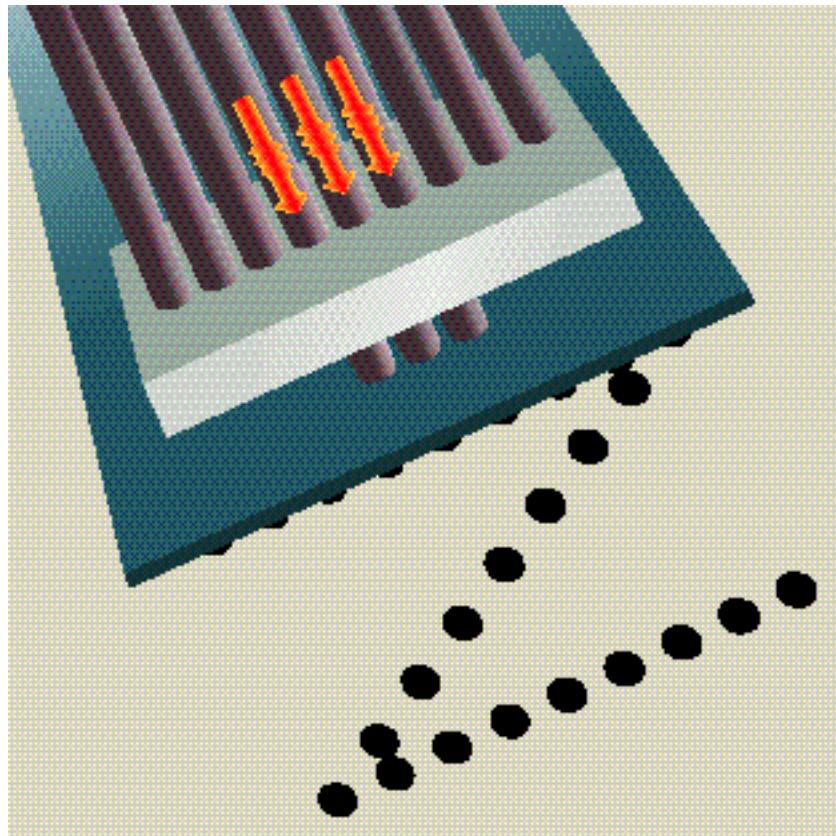
- 5 Instead of sending the individual commands for each character in a document, the page description language sends instructions to the printing mechanism that produces the page as a whole. Under this scheme, the page is essentially one large graphic image that may also happen to contain text; text and graphics are treated the same here. Treating a page as a graphic rather than as a series of characters generally makes producing text slower with a page description language than with a bitmap.



CHAPTER

38

# How a Dot-Matrix Printer Works

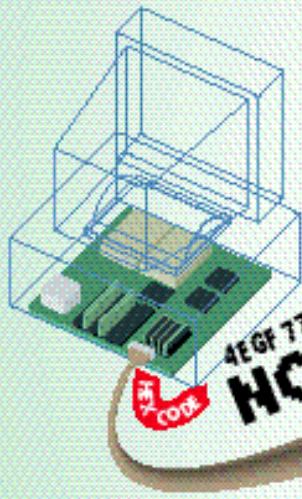


THE dot-matrix printer—noisy, slow, and crude—is a dying technology. Laser printers are faster and produce more attractive documents. Ink-jet printers cost little more than a dot-matrix, but produce beautiful color and resolution. The only thing a dot-matrix printer has on its cousins is that it can handle multilayer forms—carbons—whereas the others can't. That's because a dot-matrix printer is, like a typewriter, an impact printer. It prints by pounding ink onto paper. But carbons being, themselves, a sort of retro-technology, even that advantage is dubious.

That's sort of a shame. The dot-matrix is like an amusement park's old, wooden roller coaster. Sure, there are newer rides that fling you into multiple Gs, upside-down, and sideways. But the creaking wood and steel of a classic roller coaster has a mechanical appeal that's also found in the dot-matrix printer. It lets you see the letters as they appear behind the rushing print head and you hear the whine that tells you it's hard at work. There's something exciting about it that newer printers lack—the same seat-of-the-pants involvement you get from a sport car's stick shift.

Enjoy a dot-matrix printer while you can. Someday you'll be able to brag that, a long time ago, you actually used a printer that worked by committing miniature violence against a ribbon smeared with soot. But they are not really as primitive as they sound. Impact printers with 24 or more pins produce documents that rival the resolution of laser printers, and some dot-matrix printers can interpret commands from PostScript or another page description language. But most impact printers are simple things, designed to work with another ancient technology, bitmapped type controlled by ASCII codes sent to the printer from a PC.

# How a Dot-Matrix Printer Works



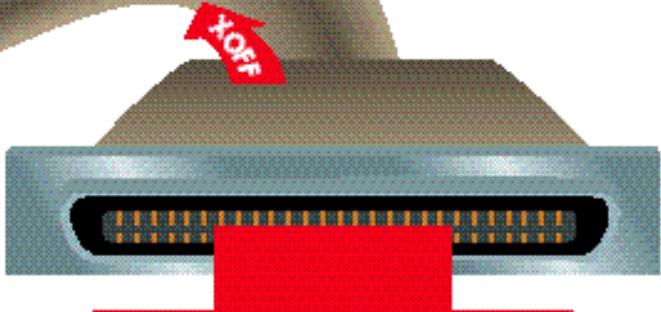
1 Your PC sends a series of hexadecimal ASCII codes that represent characters, punctuation marks, and printer movements such as tabs, carriage returns, line feeds, and form feeds that control the position of the print head in relation to the paper.

2 The ASCII codes are stored in a buffer, which is a special section of the printer's random access memory (RAM). Because it usually takes longer for a dot-matrix printer to print characters than it takes a PC and software to send those characters to the printer, the buffer helps free up the PC to perform other functions during printing. When the buffer gets full, the printer sends an XOFF control code to the computer to tell it to suspend its stream of data. When the buffer frees up space by sending some of the characters to its processor, the printer sends an XON code to the PC, which resumes sending data.

3 Among other codes are commands that tell the printer to use a certain font's bitmap table, which is contained in the printer's read-only memory chips. That table tells the printer the pattern of dots that it should use to create the characters represented by the ASCII codes.

4 The printer's processor takes the information provided by the bitmap table for an entire line of type and calculates the most efficient path for the print head to travel. Some lines may be printed from right to left. The processor sends the signals that fire the pins in the print head, and it also controls the movements of the print head and platen.

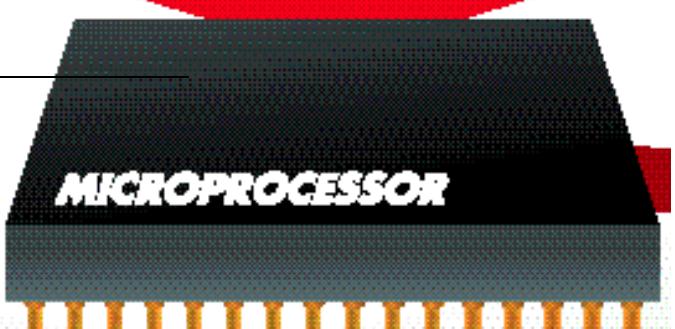
4E GF 77 20 69 73 20 74 68 65 20 69 6D 65 20 66 6F 72 20 61 6C 6C 20  
2E 2E 2E ...  
**HOW IS THE TIME FOR ALL**

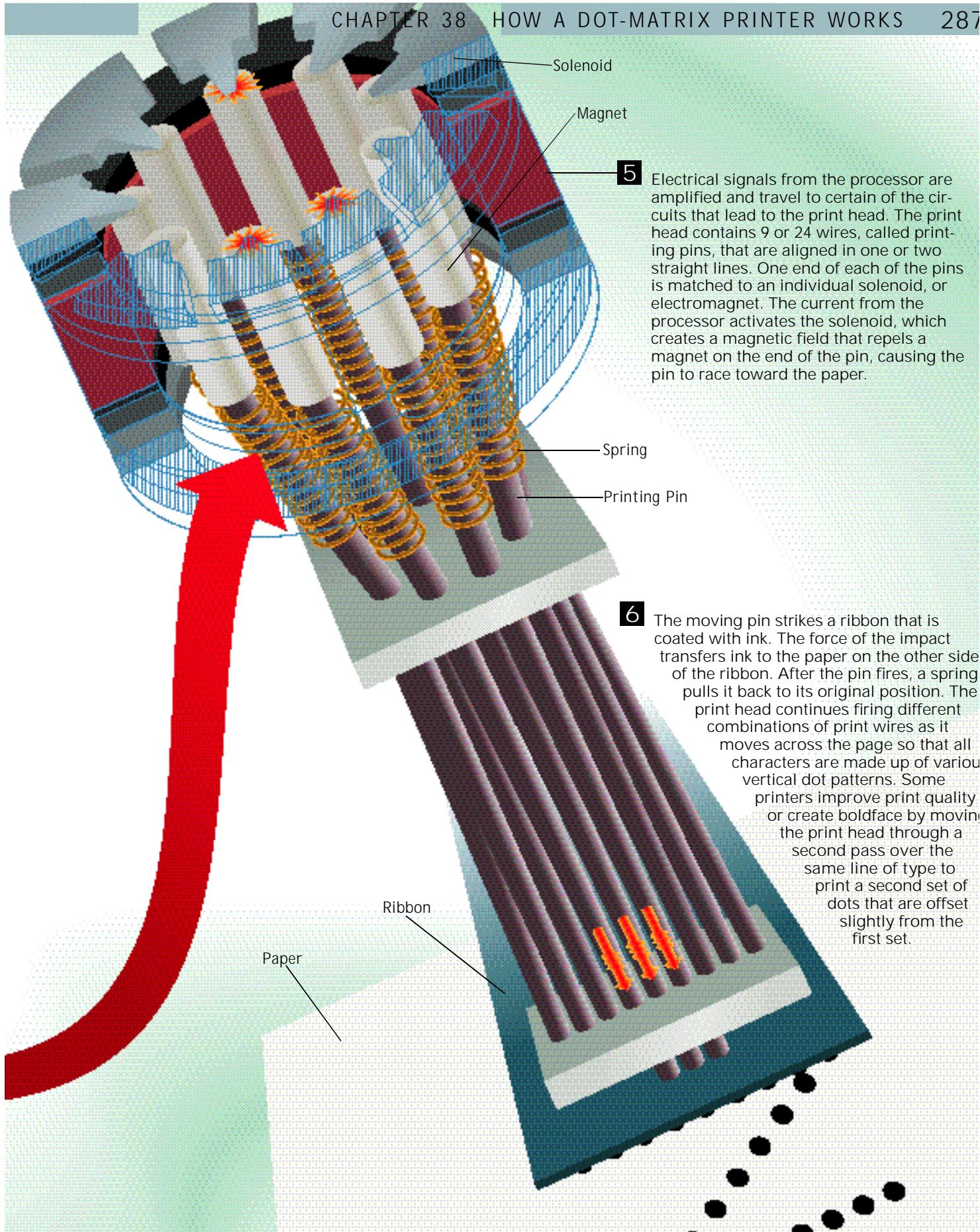


4E GF 77 20 69 73 20 74 68 65  
20 74 69 6D 65 20 66 6F 72 20  
61 6C 6C 20  
**BUFFER**

**COURIER TABLE**  
41h= A  
42h= B  
43h= C

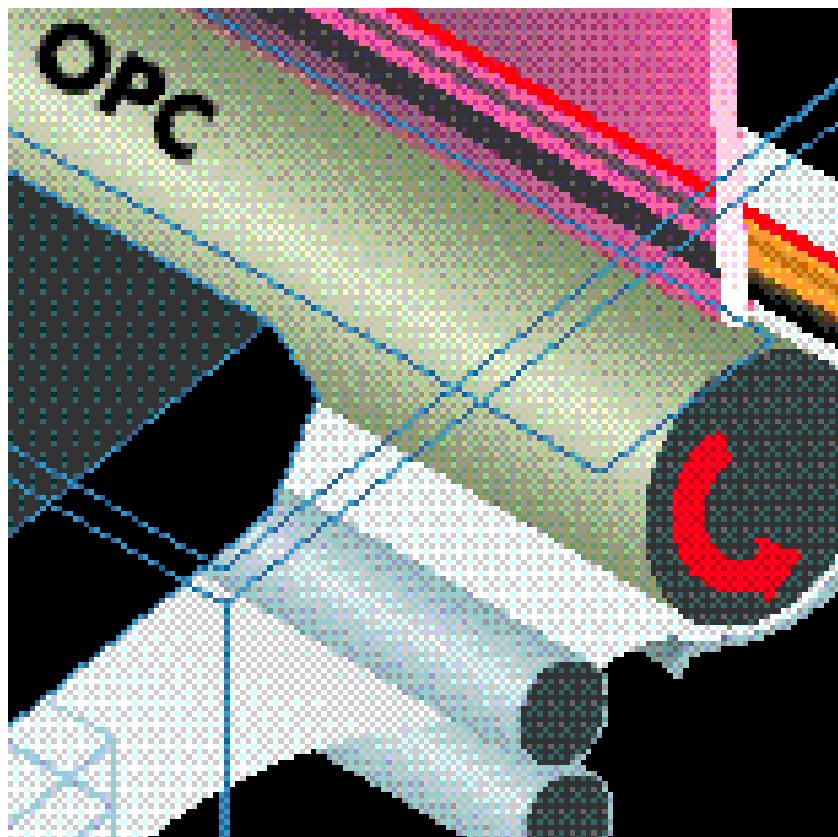
**LINE PRINTER TABLE**  
41h= A  
42h= B  
43h= C





CHAPTER  
**39**

# How a Laser Printer Works



**EVERY** time you send a page to your laser printer, you're setting in motion a complex, interlocked series of steps as efficiently organized as a factory and as precisely choreographed as a ballet.

At the heart of the printer is the print engine—the mechanism that transfers a black powder to the page—which is a device that owes its ancestry to the photocopier. Its operation includes technologies Gutenberg never imagined, including laser imaging, precise paper movement, and microprocessor control of all its actions.

To create the nearly typeset-quality output that is characteristic of a laser printer, the printer must control five different operations at the same time: (1) It must interpret the signals coming from a computer, (2) translate those signals into instructions that control the firing and movement of a laser beam, (3) control the movement of the paper, (4) sensitize the paper so that it will accept the black toner that makes up the image, and (5) fuse that image to the paper. It's a perfectly coordinated five-ring circus.

The result is no-compromise printing. Not only does the laser printer produce hard copy faster than most any other type of printer, but the laser-printed pages are more sharply detailed. With the introduction of color laser printers, the five-ring circus turns into a 20-ring bazaar. For the foreseeable future, the laser is the standard for high-end, day-in, day-out office printing.

# How a Laser Printer Works

**1**

Your PC's operating system and software send signals to the laser printer's processor to determine where each dot of printing toner is to be placed on the paper. The signals are one of two types—either a simple ASCII code or a page description language command. (See Chapter 37, "How Fonts Work.")

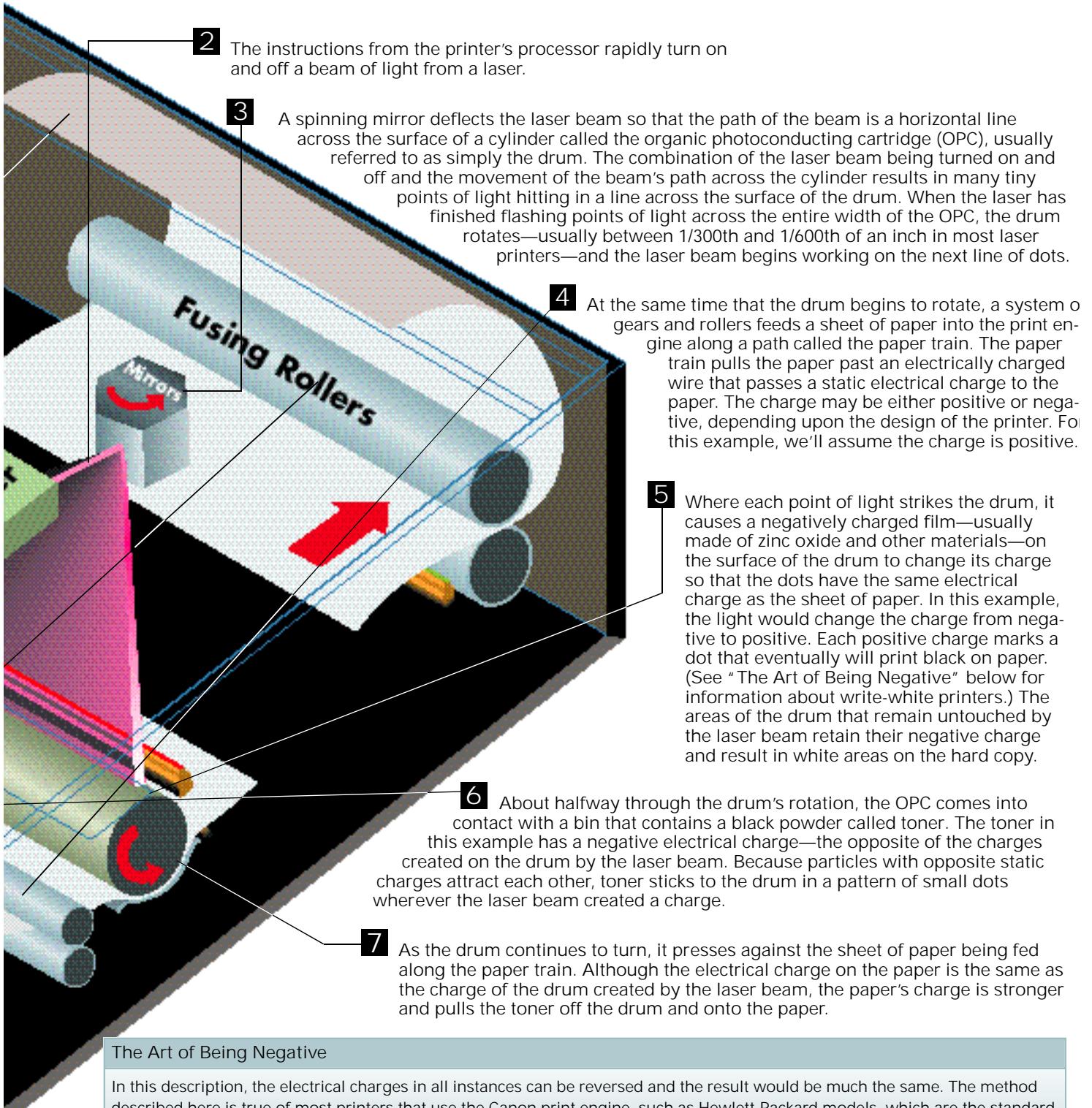
- 10** The paper train pushes the paper out of the printer, usually with the printed side down so that pages end up in the output tray in the correct order.

**9**

Another set of rollers pulls the paper through a part of the print engine called the fusing system. There, pressure and heat bind the toner permanently to the paper by melting and pressing a wax that is part of the toner. The heat from the fusing system is what causes paper fresh from a laser printer to be warm.

**8**

The rotation of the drum brings its surface next to a thin wire called the corona wire. It's called that because electricity passing through the wire creates a ring, or corona, around it that has a positive charge. The corona returns the entire surface of the drum to its original negative charge so that another page can be drawn on the drum's surface by the laser beam.



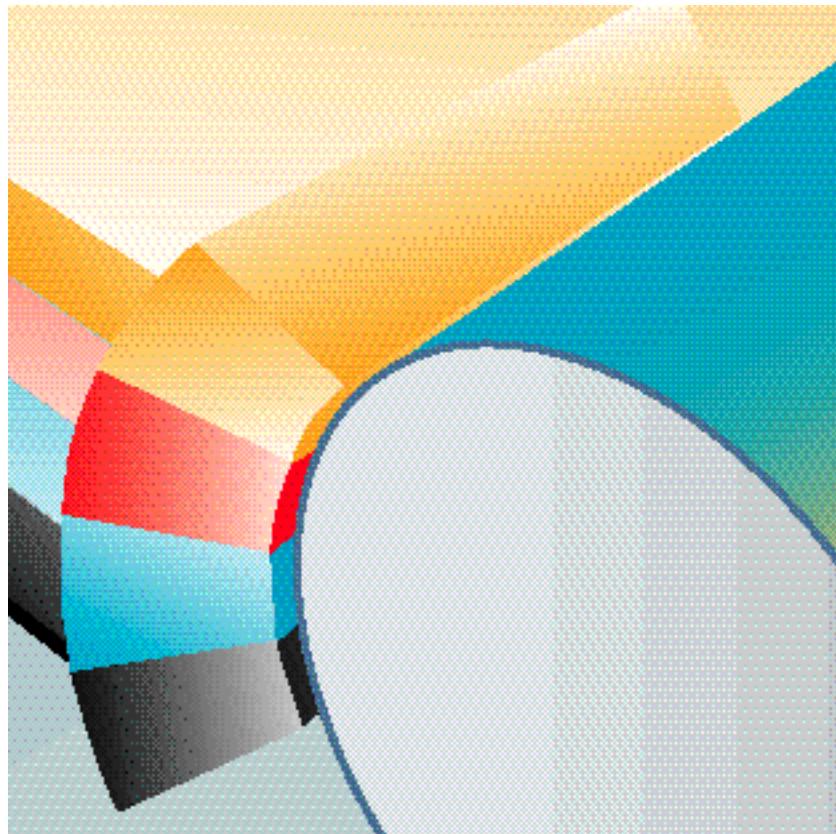
#### The Art of Being Negative

In this description, the electrical charges in all instances can be reversed and the result would be much the same. The method described here is true of most printers that use the Canon print engine, such as Hewlett-Packard models, which are the standard among laser printers. This approach is called write-black because every dot etched on the printer drum by the laser beam marks a place that will be black on the printout. There is, however, an alternative way that a laser printer can work and that way produces noticeably different results. The other method, used by Ricoh print engines, is called write-white because everywhere the laser beam strikes, it creates a charge the same as that of the toner—the toner is attracted to the areas not affected by the beam of light. Write-white printers generally produce darker black areas, and write-black printers generally produce finer details.

CHAPTER

40

# How Color Printing Works



**THERE** have been two revolutions in computer printing in the last decade. One was the laser printer, which brings typeset-quality printing of text and graphics to the masses. The second was the development of inexpensive, fast, high-quality color printing.

The complexity of color printing, of course, means trade-offs. At the low-price end is the color ink-jet printer. It is in some ways a dot-matrix printer without the impact and with four times the colors. A color ink-jet costs barely more than a black-and-white ink-jet. The visual detail approaches that of laser printers, in some printers surpassing it. But ink-jet technology is relatively slow and you always have to fuss with cleaning and replacing the ink-filled print heads. Color ink-jets are the ideal printer for the home, where printing volume is small, a budget may be nonexistent, and the flash of color in a school report or a greeting card is worth the extra wait.

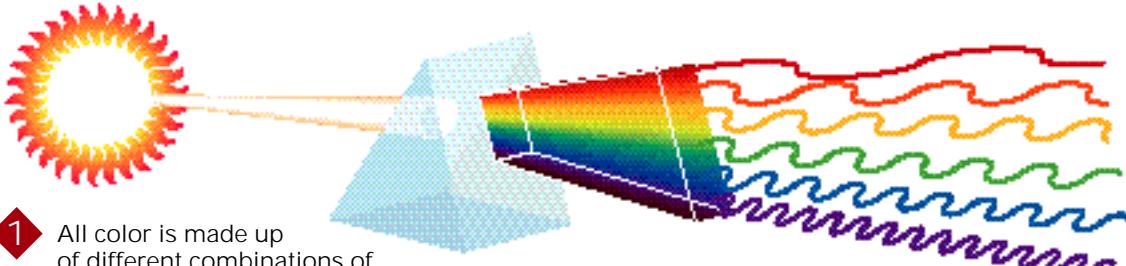
For the office, there are different color printing solutions that match the budget of a small business or home office and solutions that give the most fussy graphic artists the speed, color-matching, and details they need to create professional results. The crucial difference among color printers is how they get ink on the paper. Because it involves four colors of ink to achieve full color printing, a printer must either make multiple passes over the same sheet of paper—as happens with laser and thermal wax color printers—or it must manage to transfer all the colors more or less simultaneously, which is what happens with ink-jet and solid-ink printers.

A common office color-printing device is the color thermal printer. The process provides vivid colors because the inks it uses don't bleed into each other or soak into specially coated paper. But its four-pass method is slow and wastes ink. The color laser printer provides the most precise detail but is slow, complicated, and expensive because it requires four separate print engines that must each take their turn to apply colored toner to the page.

Two other color printing methods provide speed and photographic dazzle: dye-sublimation—also called dye diffusion thermal transfer (D2T2)—and solid-ink. By controlling not only how many dots of color they put on the page but the intensity of the dots, they produce continuous-tone printing. The result is virtually indistinguishable from a color photograph, even though its actual resolution may be no more than the 300 dots per inch of the old laser printer. If the results you're trying to get with color printing are really important, these technologies are well worth the cost.

In this chapter, we'll look at how color printing tricks the eye into seeing hues and shades that aren't really there, and how ink-jet, color thermal, dye-sublimation, and solid-ink color printers work. To learn how four-color laser printers work, turn to Chapter 39 on black-and-white laser printers and read it four times.

# How Printed Colors Are Created

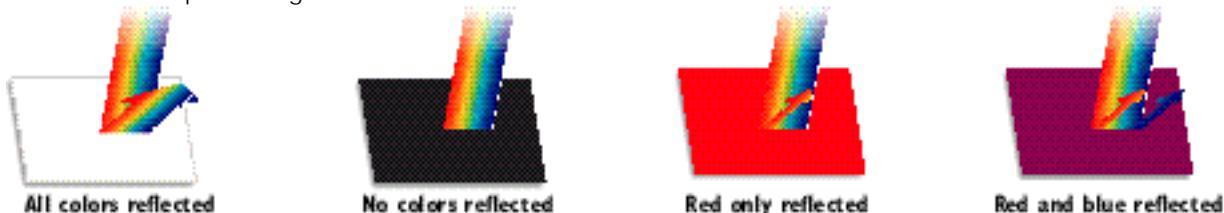


- 1** All color is made up of different combinations of light. We see all the wavelengths that make white when we pass white light through a prism, breaking it into the spectrum. Although the spectrum is a continuous blending of colors, color is reproduced by using only a few of those colors. Different mixtures of those primary colors re-create virtually any color from the spectrum by either adding colors or subtracting them.



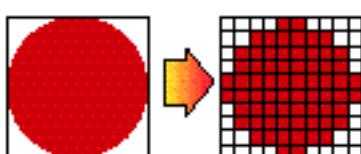
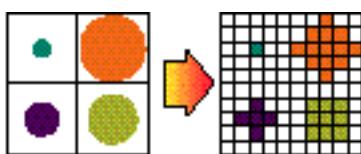
- 2** Additive color is used to create colors on televisions, computer monitors, and in movies. Three colors—red, green, and blue—are emitted to produce all other colors and white by adding various intensities of those primaries. Each time a color is added, it increases the number of colors the eye sees. If red, green, and blue are all added at their most saturated shades, the result is white.

- 3** Subtractive color is the process used when light is reflected—rather than emitted as in additive color—from colored pigments. Each added color absorbs (subtracts) more of the shades of the spectrum that makes up white light.

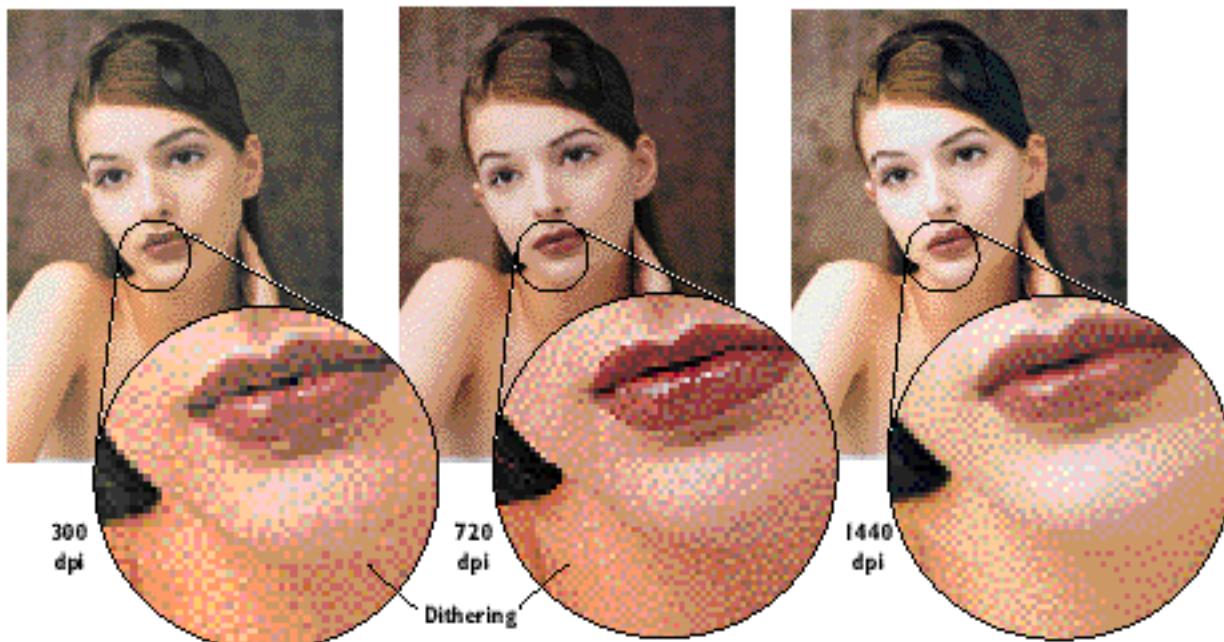


- 4** Color printing uses four pigments: cyan (blue-green), yellow, magenta (purple-red), and black. This system is called CYMK. (K stands for black.) Some low-end color ink-jet printers save the cost of a black-ink print head by using equal portions of magenta, yellow, and cyan to produce black. But the resulting black lacks density, which is why better personal printers include a print head for black ink.

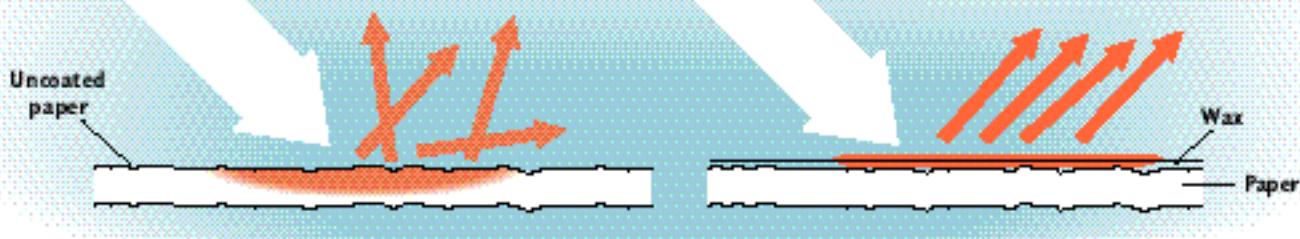




- 5** All color printers use tiny dots of those four inks to create various shades of color on the page. Lighter shades are created by leaving dots of unprinted white. Some printers, such as dye-sublimation, control the size of the dots and produce continuous-tone images that rival photography. But most printers create dots that are essentially the same size no matter how much of a particular color is needed. The most common color printers create up to 300 dots of color per inch, for a total of 8 million dots per page. Many printers can create about 700 dpi, and a few printers can create up to 1,440 dpi.



- 6** For all shades beyond the eight that are produced by overlaying the primaries, the printer generates a varied pattern of differently colored dots. For example, the printer uses a combination of 1 magenta dot to two of cyan to produce a deep purple. For most shades of color, the dots of ink are not printed on top of each other. Instead, they are offset slightly, a process called dithering. The eye accommodatingly blends the dots to form the desired shade as it hides the jagged edges, or jaggies, produced by the dots. Dithering can produce nearly 17 million colors.

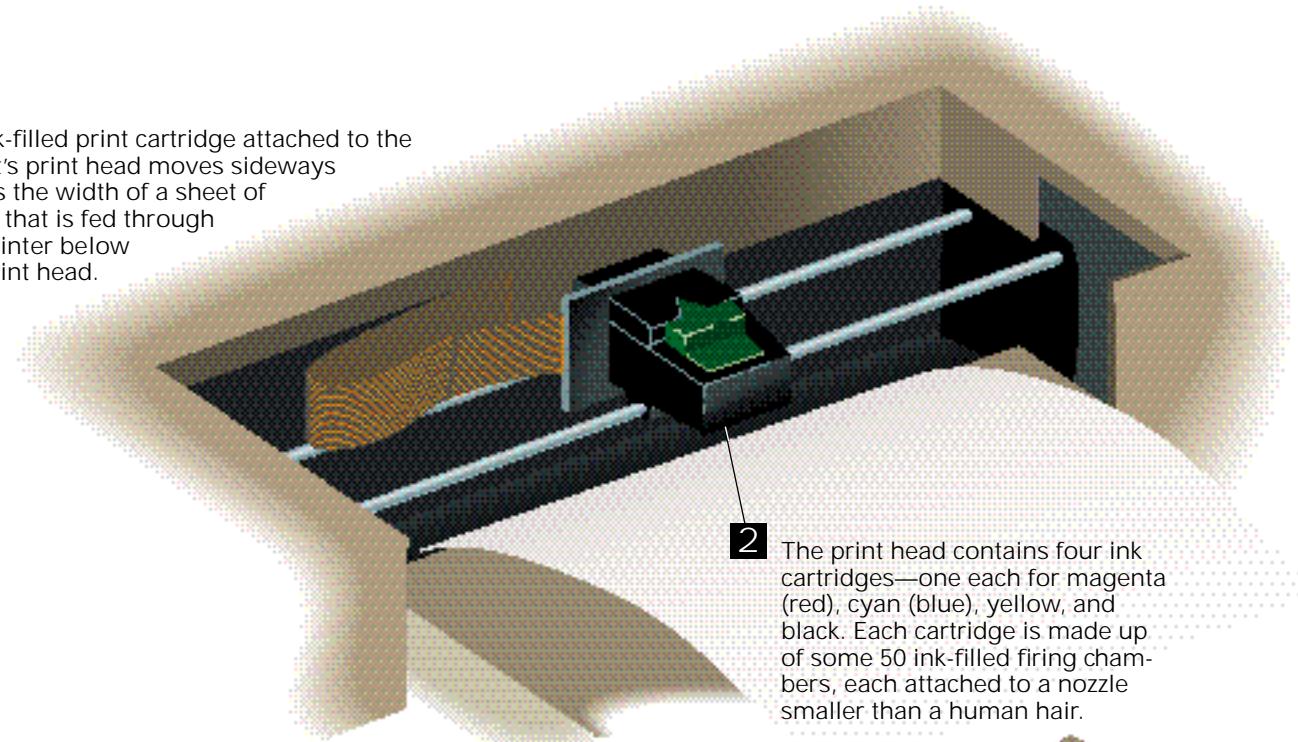


- 7** The type of paper used in color printing affects the quality of the hard copy. Uncoated paper, the type used with most black-and-white office machines, has a rough surface that tends to scatter the light, reducing the brightness, and it tends to absorb ink, which slightly blurs the image.

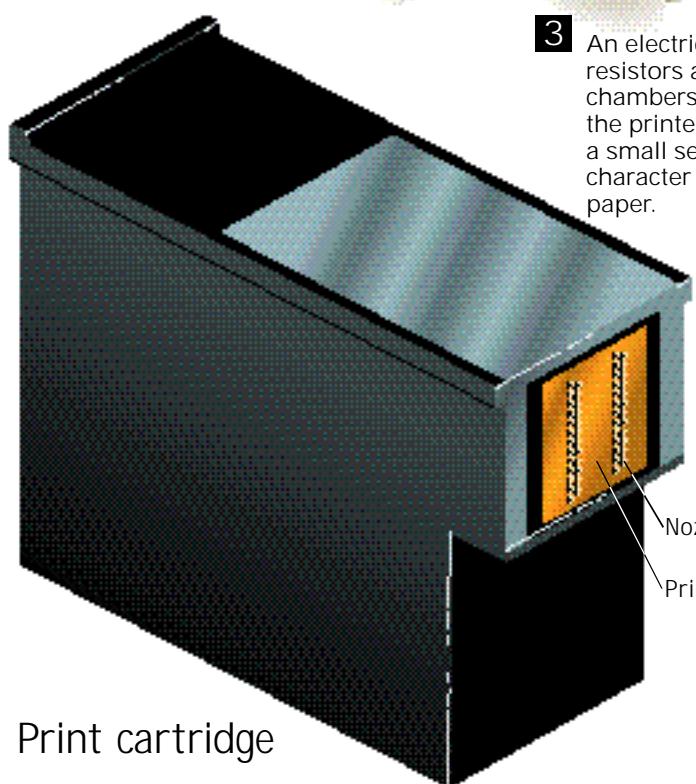
Paper coated with a fine varnish or wax takes applications of ink more evenly so that the ink dries with a smooth surface that reflects more of the light hitting it. The coating also helps prevent the ink being absorbed by the paper, for a sharper image.

# How a Color Ink-Jet Printer Works

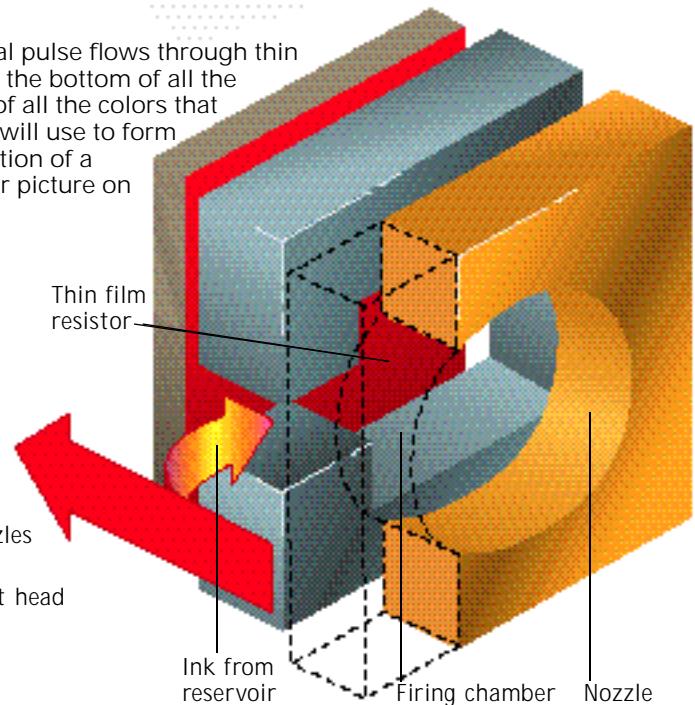
- 1 An ink-filled print cartridge attached to the ink-jet's print head moves sideways across the width of a sheet of paper that is fed through the printer below the print head.



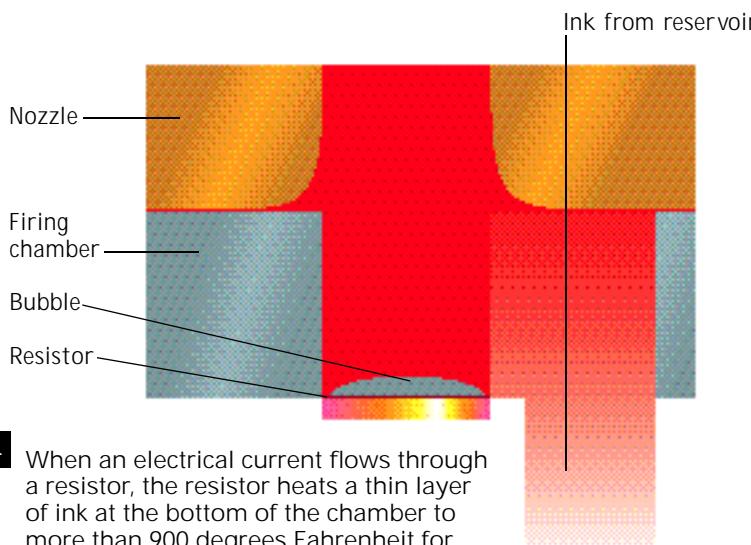
- 2 The print head contains four ink cartridges—one each for magenta (red), cyan (blue), yellow, and black. Each cartridge is made up of some 50 ink-filled firing chambers, each attached to a nozzle smaller than a human hair.



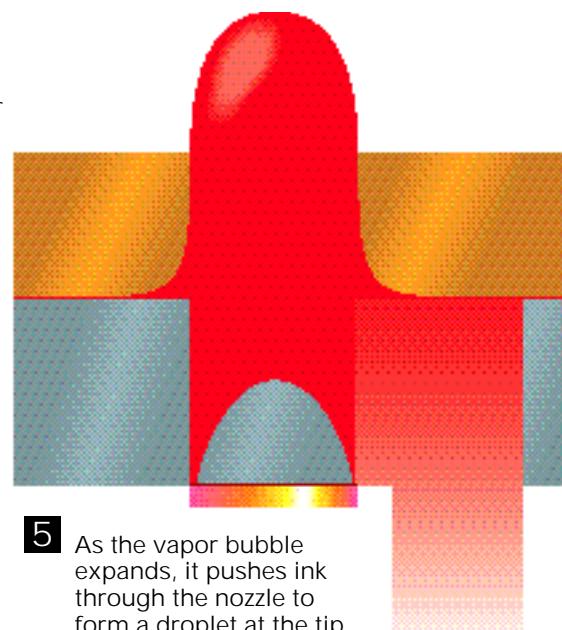
- 3 An electrical pulse flows through thin resistors at the bottom of all the chambers of all the colors that the printer will use to form a small section of a character or picture on paper.



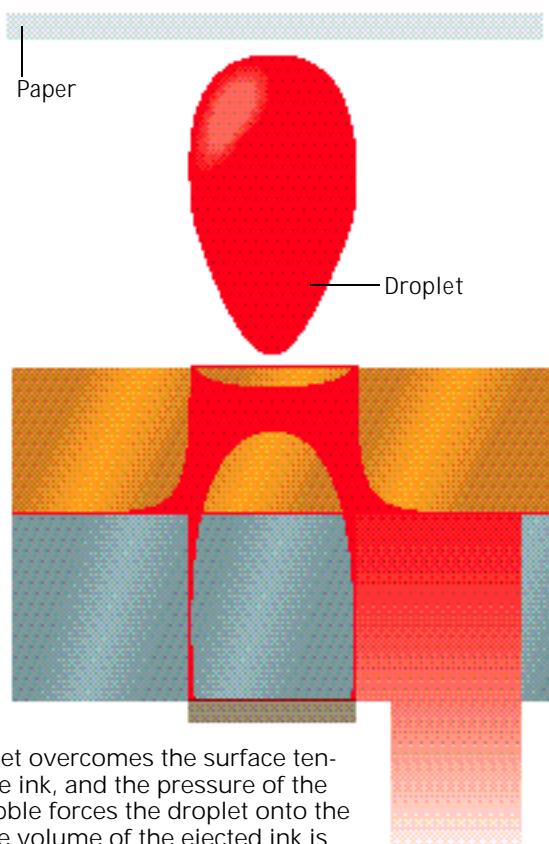
Nozzle cross-section



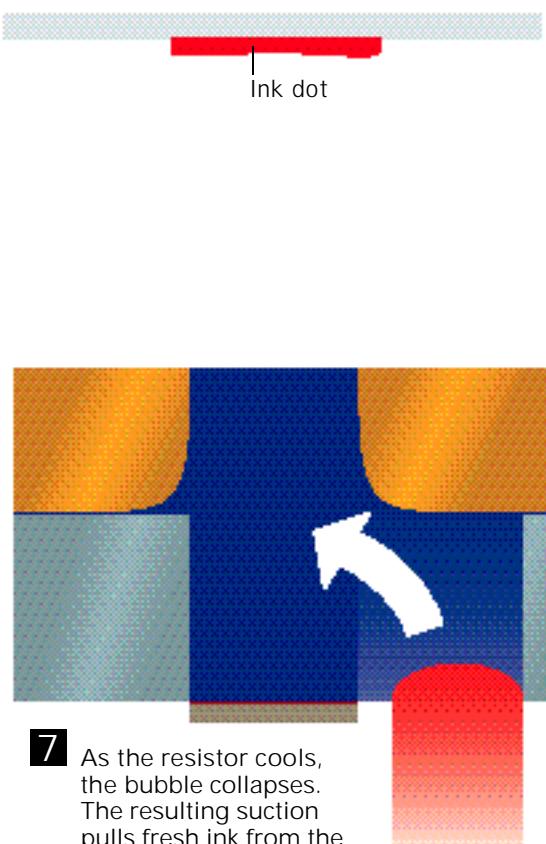
- 4** When an electrical current flows through a resistor, the resistor heats a thin layer of ink at the bottom of the chamber to more than 900 degrees Fahrenheit for several millionths of a second. The ink boils and forms a bubble of vapor.



- 5** As the vapor bubble expands, it pushes ink through the nozzle to form a droplet at the tip of the nozzle.

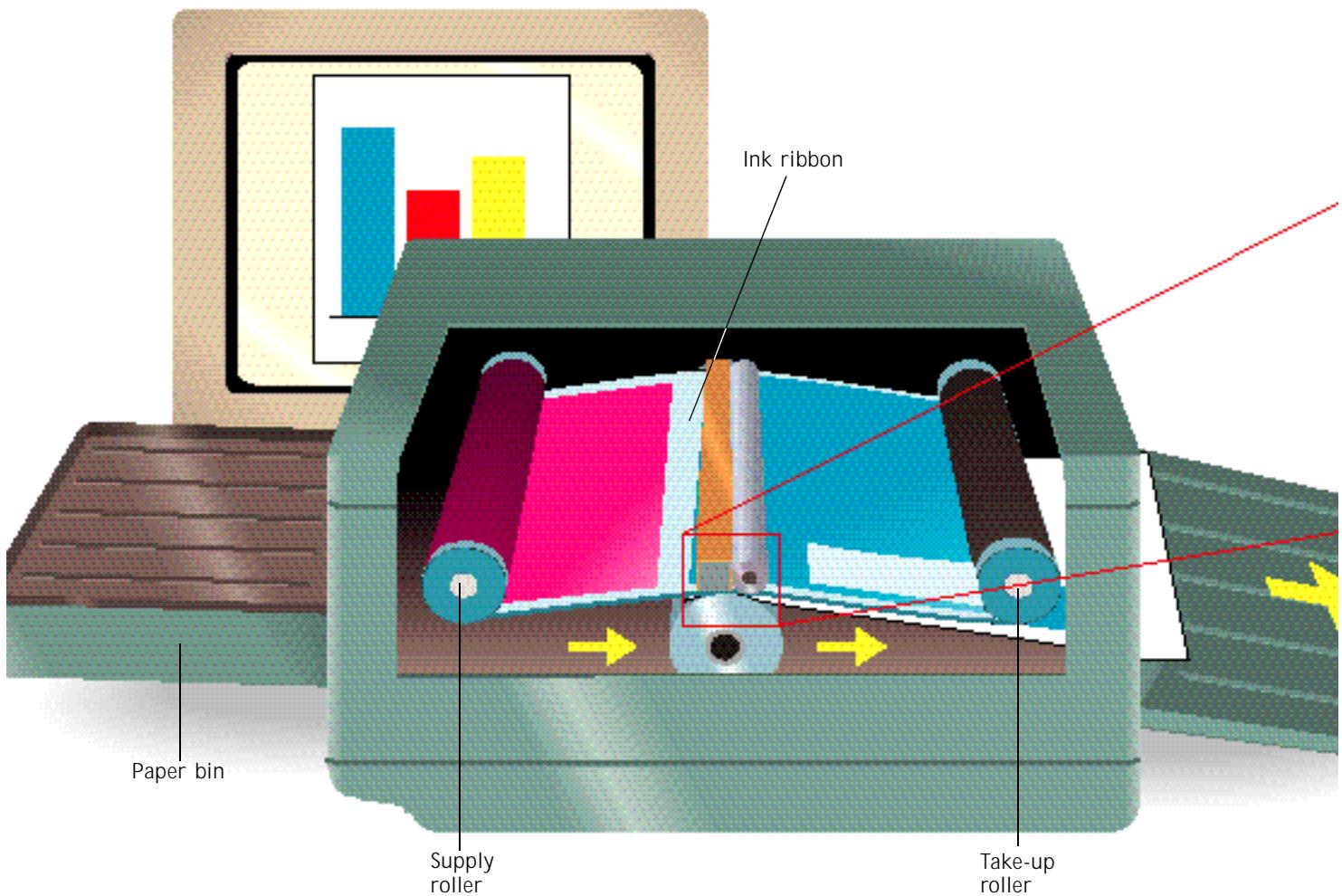


- 6** The droplet overcomes the surface tension of the ink, and the pressure of the vapor bubble forces the droplet onto the paper. The volume of the ejected ink is about one millionth that of a drop of water from an eyedropper. A typical character is formed by an array of these drops 20 across and 20 high.

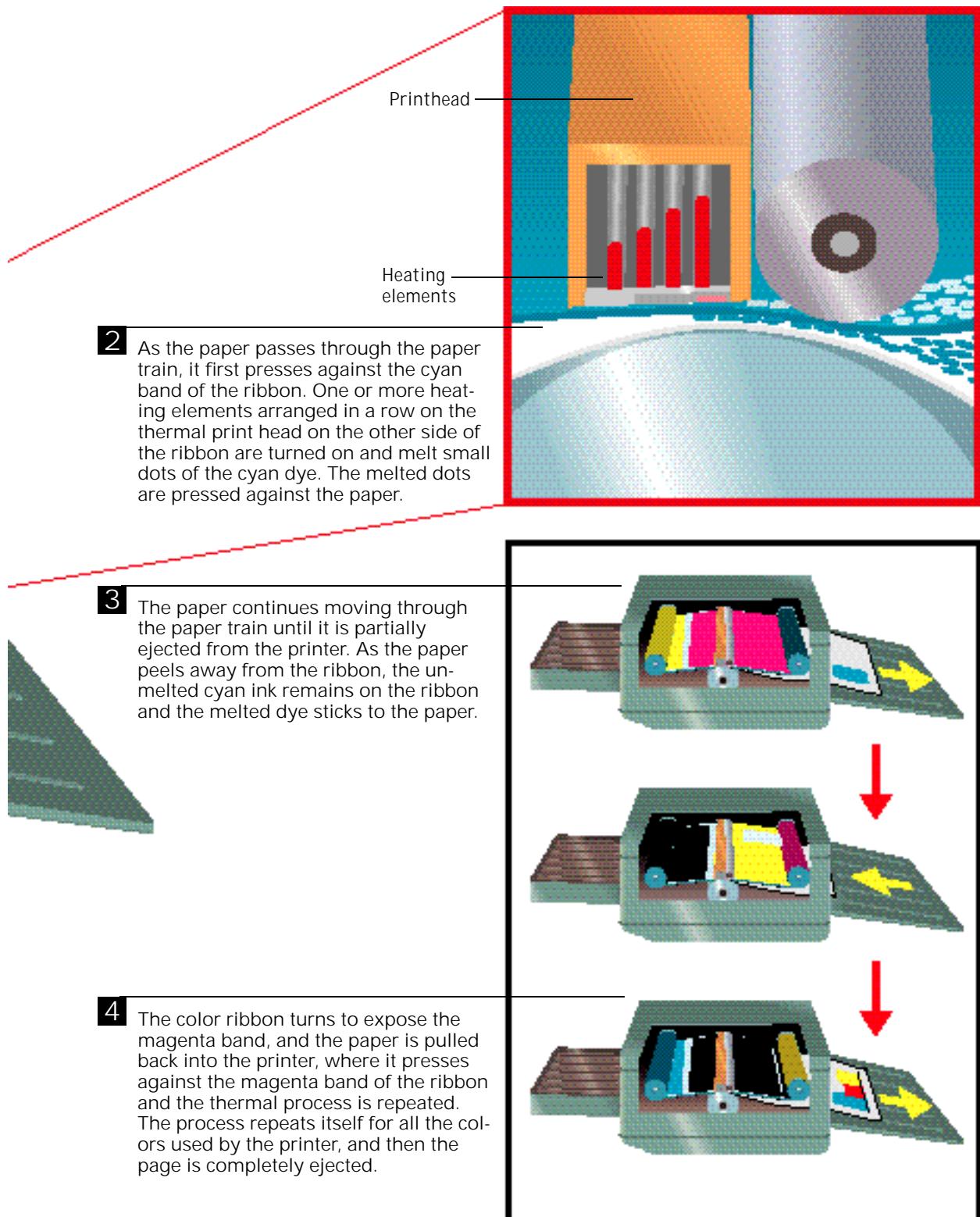


- 7** As the resistor cools, the bubble collapses. The resulting suction pulls fresh ink from the attached reservoir into the firing chamber.

# How a Color Thermal Printer Works

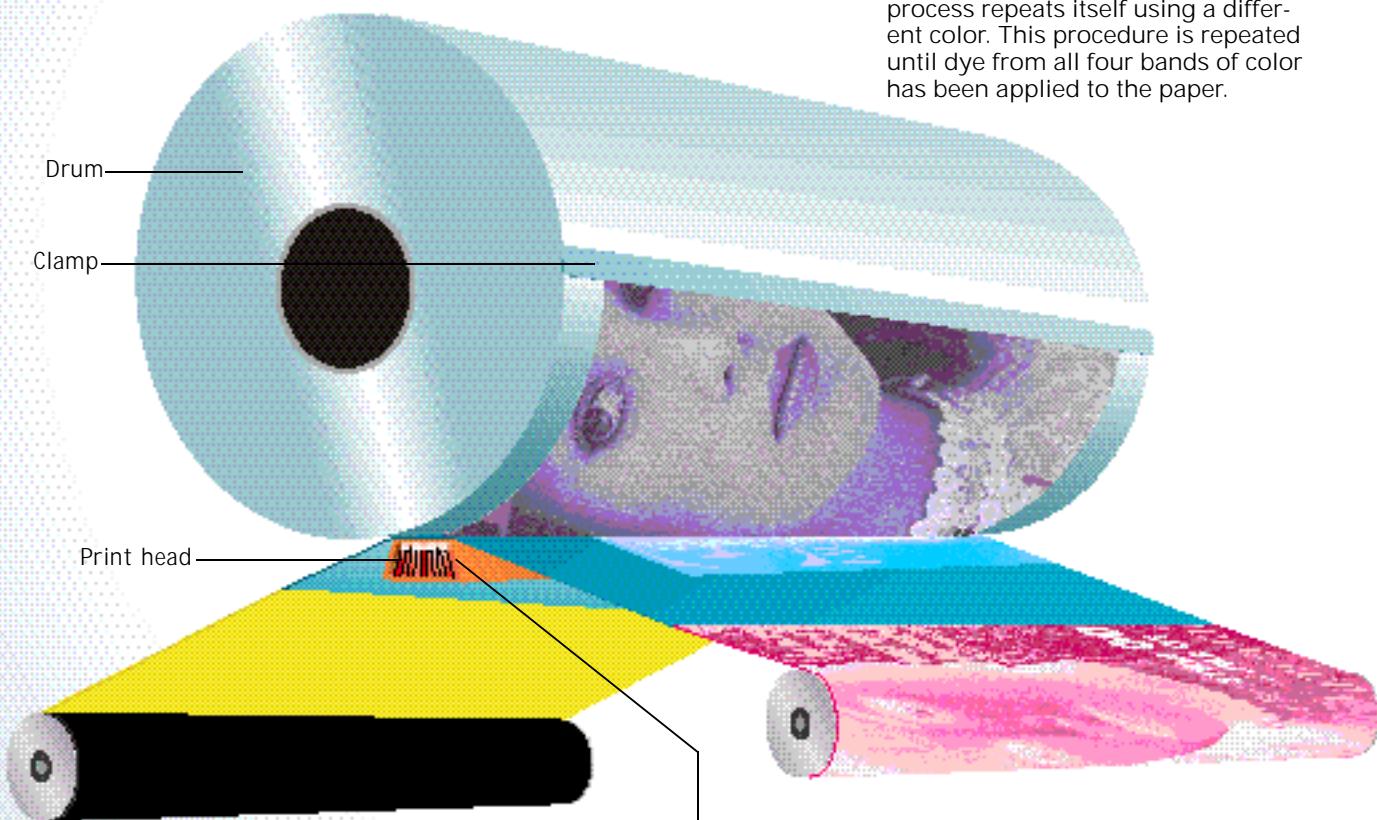


- 1 The color thermal printer feeds a sheet of specially coated paper from its bin into the print engine, where the paper is held on one side by a roller that presses the paper against a wide ribbon coated with colored inks mixed with wax or plastic. The ribbon contains a band of each of the composite printing colors—cyan, magenta, yellow, and, if it's used, black. Each color band covers a large area—the width and length of the sheet of paper.



# How a Photo-Quality Printer Works

**1** A dye-sublimation printer, which produces photographic quality hard copy, uses a paper specially made to accept the printer's colors. A sheet of the paper is clamped to a drum to ensure that all the colors, which are applied in separate passes, precisely line up with each other.



**2** As the drum turns, other gears in the printer turn a transfer roll of plastic film. The film—the equivalent of a ribbon in other printers—contains cyan (blue), magenta (red), yellow, and black dyes in bands the same width and height as the sheet of paper. Moving at the same speed as the surface of the drum holding the paper, the film passes along the surface of the paper.

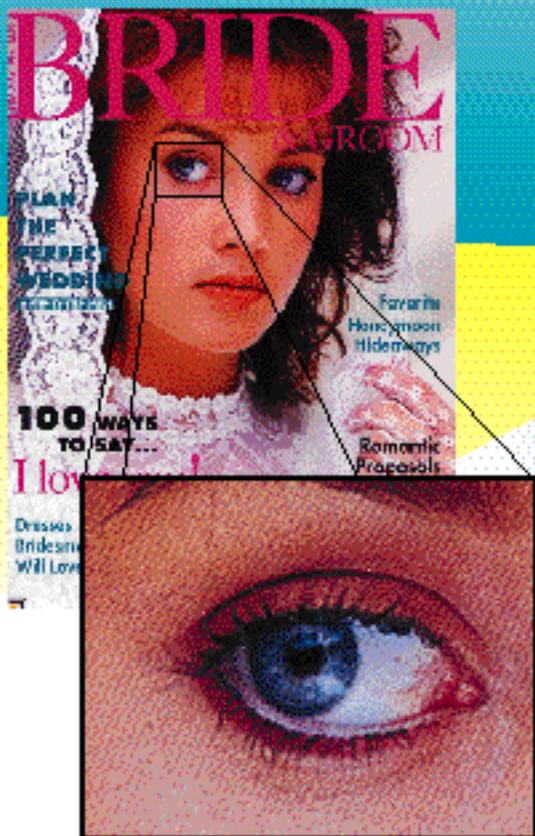
**4** After the dye for one color has been transferred to the paper, the drum reverses direction, returning the paper to its start position. Then the process repeats itself using a different color. This procedure is repeated until dye from all four bands of color has been applied to the paper.

**3** At the point where the transfer roll is nearest the surface of the paper, thousands of heating elements cause the dye to sublimate—a scientific term for a solid melting to a gas without going through an intervening liquid stage. The gaseous dye is absorbed into the fiber of the paper.

- 5 Each of the heating elements produces any of 256 different temperatures. The hotter the temperature, the more that dye is sublimated and transferred to the paper, producing a different shade of the dye's color. In effect, this gives the dye-sublimation printer the ability to create 16 million colors without resorting to the dithering used by other printers to mix colors.



360 percent enlargement of thermal wax transfer hard copy



360 percent enlargement of dye-sublimation hard copy

- 6 Sublimation also accounts for the resolution looking higher than it is. A dye-sublimation printer may produce only 300 dots per inch, but, unlike a thermal wax transfer printer at 300dpi, the dots are not all the same size. A heating element that has a cooler temperature produces a smaller dot than a heating element with a hotter temperature. The result is that the apparent resolution of the hard copy is of photographic quality.

# How a Solid-Ink Color Printer Works

**1** A solid-ink color printer such as the Tektronix Phaser 350, which prints six pages a minute on ordinary paper, uses inks that are solid at room temperatures. They are loaded into the printer as wax-like blocks for each of the four colors the printer uses. The holes for loading the blocks are shaped differently for each color to prevent loading a color into the wrong hole.

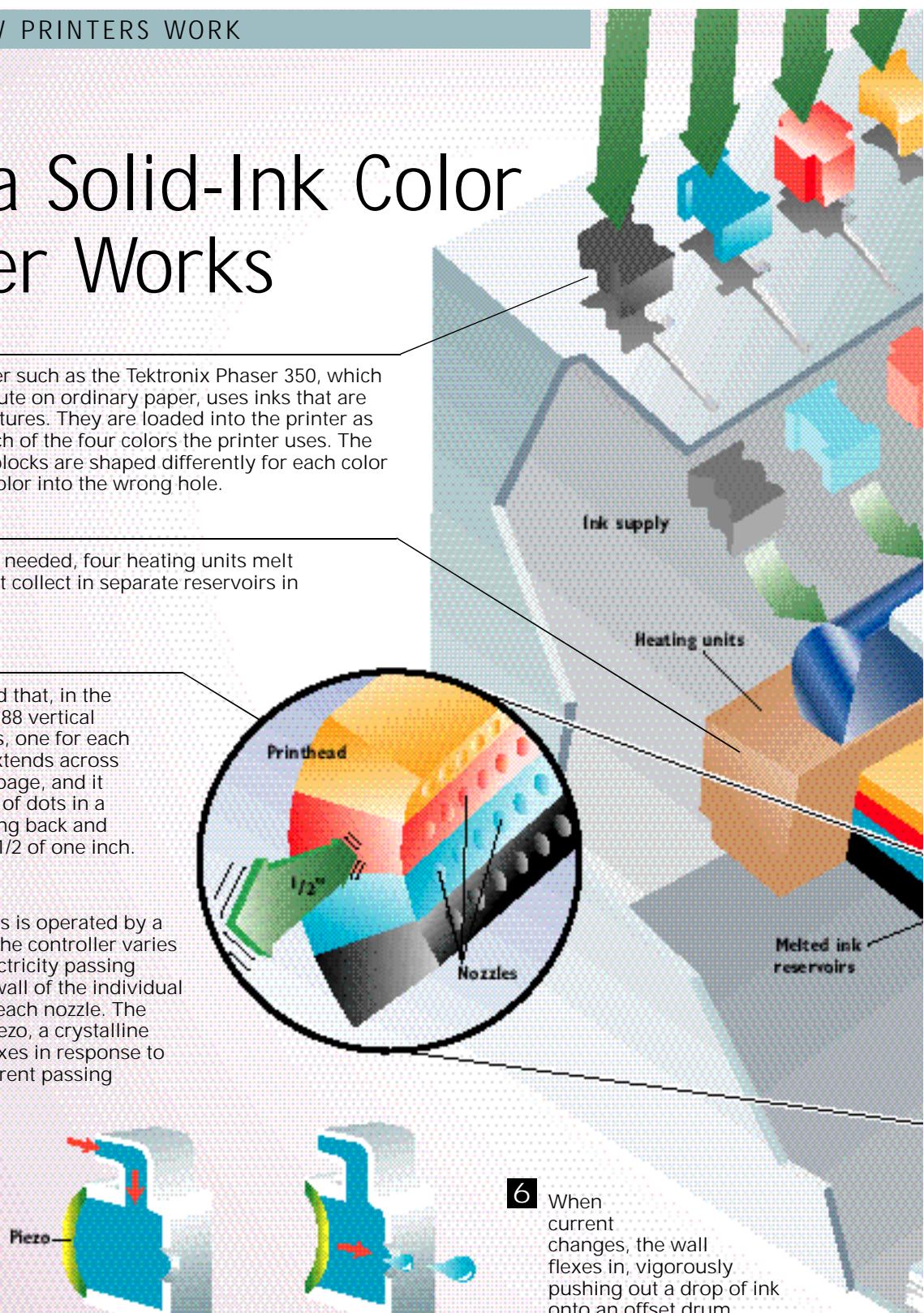
**2** As each of the colors is needed, four heating units melt the inks into liquids that collect in separate reservoirs in the print head.

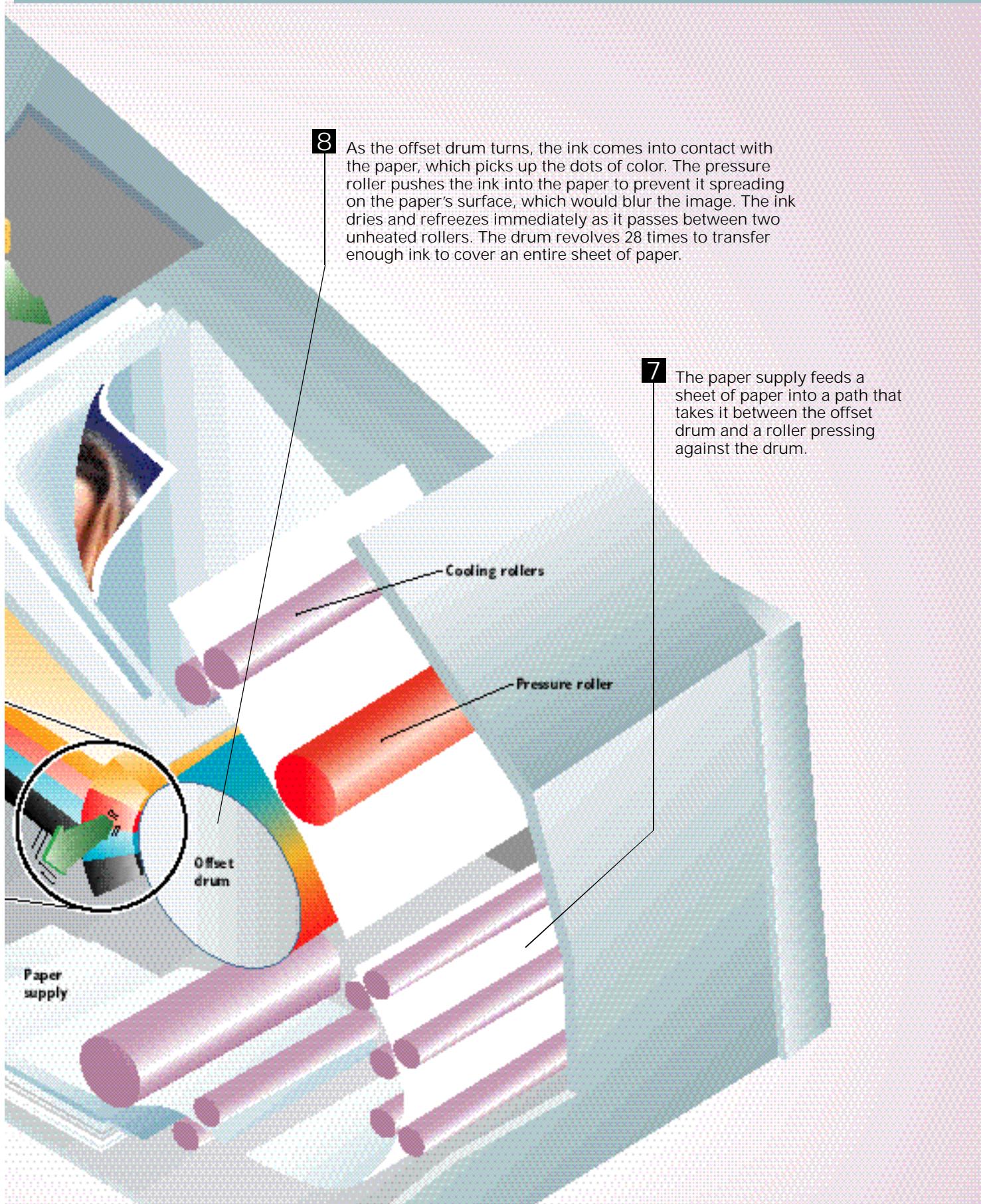
**3** Ink flows to a print head that, in the Phaser 350, consists of 88 vertical columns of four nozzles, one for each color. The print head extends across the entire width of the page, and it produces an entire line of dots in a single pass while moving back and forth horizontally only 1/2 of one inch.

**4** Each of the nozzles is operated by a piezo controller. The controller varies the amount of electricity passing through the rear wall of the individual ink chambers for each nozzle. The wall is made of piezo, a crystalline substance that flexes in response to the amount of current passing through it.

**5** As the wall moves back, it sucks ink in from the reservoir and back from the mouth of the nozzle. The farther the wall flexes, the more ink it pulls in, allowing the controller to change the amount of ink in a single dot.

**6** When current changes, the wall flexes in, vigorously pushing out a drop of ink onto an offset drum coated with a silicone oil. The drum is slightly warm to keep the ink in a liquid state.





# Index

## Symbols



- ACK line (acknowledge line), 137
- ATN, (attention line), 136
- BSY, see busy line
- REQ line, see request line
- .AVI (audio/video interleave), 226-227
- .BMP file extension, 144
- .JPG file extension, 144
- .PCX file extension, 144
- .TIF file extension, 144
- //, URLs, 268
- 16-bit expansion cards, see ISA cards
- 16-bit PCs, binary numbers, 39
- 24-bit graphics, 145
- 3-D cards, 237
- 3-D graphics
  - coprocessors, 237
  - creating characters, 236
  - virtual reality, 232-237
- 3.5-inch disks, 83-84
- 3.5-inch floppy drives, 83-85
- 32-bit EISA (Extended Industry Standard Architecture) expansion cards, 123
- 32-bit MCA (Microchannel Architecture) expansion cards, 122
- 32-bit PCI (Peripheral Component Interconnect) Local-Bus expansion cards, 123
- 32-bit PCs, binary numbers, 39
- 32-bit VESA Local-Bus (VL-Bus) expansion cards, 123
- 5.25-inch
  - floppy disks, 84
  - floppy drives, 83
- 56K modems, analog local loop, 176

- 8-bit expansion cards, 122
- 8-bit graphics, 145
- 80-bit registers, FPUs, 65

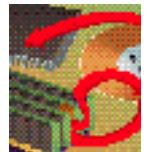
## A

- A-law codec, 56K modems (in Europe), 177
- AA (Auto Answer), modems, 175
- absolute timing in pregroove, see atip
- accelerated 3-D graphics (virtual reality), 232-237
- accelerated graphics port (AGP), 237
- accelerated graphics port, (AGP) set, 125
- accelerometers (virtual reality controllers), 230
- access time, speed of hard drives, 95
- acknowledge line (-ACK line), 137
- active-matrix color, LCDs (Liquid Crystal Displays), 152-153
- adapters, see expansion cards
- ADCs (analog-to-digital converters)
  - 56K modems, 177
  - digital photography, 199
  - flatbed scanners, 185
  - hand-held scanners, 187
  - joysticks, 167
  - sound cards, 218-219
  - video, 227
- additive color, 143
  - colors, 294
- address electrodes, digital light processing, 156
- address lines, RAM (random access memory), 44
- AFIS (automated fingerprint identification system), 197
- AGP (accelerated graphics port), 237
- AGP (accelerated graphics port) set, 125

algorithms  
     print commands, 282  
     sound cards, 221  
 Alpha blending, 235  
 ALUs (arithmetic logic units), 59, 61  
     Pentium MMX CPUs, 65  
 amplitude, 56K modems, 177  
 analog current, 225  
 analog devices, telephone systems, 171  
 analog local loop, 56K modems, 176  
 analog-to-digital converters, see ADCs  
 AND gates, logic gates, 53  
 animation, full polygon, 236  
 annealing phase, data, erasing (DVD-RAM), 213  
 AOL (America Online), e-mail, gateway software, 257  
 application layers, network communications, 252  
 arithmetic logic units, see ALUs  
 ASCII, 43  
     fonts, 281  
     notation representations, 45-47  
 ASCII codes, bitmapped fonts, 281  
 atip (absolute timing in pregroove), CD-R (CD-Recordable), 208  
 attaching, files to e-mail, 257  
 attention line (-ATN), 136  
 attributes, bitmapped fonts, 280  
 audio  
     compression, 227  
     Internet, 261  
         bandwidth, 261  
         streaming, 261-263  
 audio/video interleave, see .AVI  
 avoiding conflicts, 27

## B

backups, tape backups, 109  
 bandwidth  
     Internet audio and video, 261  
     universal serial bus (USB), 133  
 Baseball 3-D (Microsoft), 236  
 basic input/output system, see BIOS  
 baud, see bits per second  
 Ben Day dots, 143  
 bias/reset bus, 157  
 bilinear filtering, 235  
 binary numbers, 43  
     16-bit PCs, 39  
     32-bit PCs, 39  
     bits, 59  
     transistors, 39  
 BIOS (basic input/output system), 6-7, 23-24  
     operating systems, 19, 25  
     read-only memory (ROM), 23  
 bitmap tables, printers, 286  
 bitmapped fonts, 279-281  
     ASCII, 281  
     attributes, 280  
     print commands, 280-281  
 bitmaps, 143-144  
     color  
         24-bit, 145  
         8-bit, 145  
         bits per pixel requirements, 145  
         RLE (run-length encoding), 146  
     file extensions, 144  
     header signature ID, 144  
     pixels, 143  
         DAC (digital-to-analog converters), 147  
         locations, 144  
         storing values in adapter frame  
             buffer, 147  
         turning on and off, 144





bits, 39, 44-47, 75  
 binary numbers, 59  
 color bitmap pixel requirements, 145  
 reading, on disks, 74  
 writing, to disks, 74  
 bits per second, modems, 173  
 BIU (bus interface unit), Pentium Pro, 63  
 blocks, see clusters  
 Boolean logic, 39  
 Boolean XOR operation, see XOR operations  
 boot disk, process, 10-13  
 boot program, 6  
 boot record, 10-11  
 boot, see boot-up  
 boot-up, 5-7  
   operating systems, 9  
   POST (power-on self-test), 5, 9  
 brackets, session layers, 253  
 branch predictor unit, CPU (central processing unit), 61  
 branch target buffers (BTB), Pentium Pro, 63  
 bridges, Internet, data transfer, 255  
 browsing, World Wide Web (WWW), 267  
 buffers  
   BTB (branch target buffers), 63  
   TLB (translations lookaside buffers), 64  
 buses, 119  
   EISA (Extended Industry Standard Architecture), 120  
   expansion cards, 119  
   ISA (Industry Standard Architecture), 119  
   local bus, 120  
     PCI (Peripheral Component Interconnect), 119, 121, 124  
     Video Electronics Standards Association (VESA), 119, 121  
   MCA (Microchannel Architecture), 120  
   Pentium processors, 60  
   traces, 119

universal serial bus, see universal serial bus  
 bus interface unit (BIU), Pentium processors, 60  
 bus networks, see Ethernet networks  
 busy line, 136  
 bytes, 47

## C

caches, 271  
   data cache (D-cache), 63  
   I cache (instruction cache), 63  
   L1 (level 1), 63  
   L2 (level 2), Celeron and Xeon processors, 67  
 Pentium II processors  
   L1 (level 1), 66  
   L2 (level 2), 66  
 Pentium MMX processors, 64  
 Pentium processors, 60  
 TLB (translation lookaside buffers), 64  
 caching, hard drives, 95  
 caching texture maps, 237  
 cameras, see digital photography  
 capacitance, touchpads, 162  
 capacitors, 167  
   closed, 45-47  
   switches, 45-47  
 capture cards, Internet video, 264  
 Carrier Detect signal, see CD signal  
 Carrier signal, modems, 172  
 cathode-ray tubes, 150  
 CCD, see charge-coupled device, 167  
 CCDs (charge-coupled devices), digital photography, 198  
 CD (Carrier Detect) signal  
   modems, 172, 174-175

- CD jukeboxes, 214-215  
 CD-R (CD-Recordable), 205, 208-209  
 CD-ROMs, 205
  - constant linear velocity, 207
  - drives, 206-207
  - see also CD-R (CD-Recordable)
 CD-RW (CD-Rewriteable), 205, 212-213  
 CDB (command descriptor block), SCSI (Small Computer System Interface), 137  
 Celeron processors (Pentium II), 67  
 central processing units, see CPUs  
 Centronics ports, see parallel ports  
 channel hot electron injection, flash memory transistors, 48  
 characters, creating for 3-D worlds, 236  
 charge-coupled device (CCD), 167
  - hand-held scanners, 186
 circuit boards, 3.5-inch floppy drives, 85  
 CISC (complex instruction set computing) processors, 55
  - commands, 56
 Clear to Send (CTS) signal, 174  
 client software, e-mail, 257-258  
 clients, networks, 243  
 clock cycles, 62  
 closed capacitors, 45-47  
 clusters, 96
  - contiguous, 100
  - slack, 96
  - tracks, 76
  - writing to disks, 79
 CMOS, 134  
 CMOS chip, 7  
 codecs (compression/decompression), 264
  - corrupted video packets, 265
  - interframe video compression, 264
  - IP multicast video packets, 265
  - symbols, 56K modems, 177
  - UDP video packets, 265
 color bitmaps
  - 24-bit, 145
  - 8-bit, 145
  - bits per pixel requirements, 145
  - pixels
    - DAC (digital-to-analog converters), 147
    - storing values in adapter frame buffer, 147
    - RLE (run-length encoding), key bytes, 146
 color ink-jet printers, 296-297  
 color laser printers, 293  
 color printers, see printers  
 color printing, 294
  - dots, 295
 color thermal printers, 293, 298-299  
 colors
  - additive color, 143, 294
  - creating, 294-295
  - digital light processing, 157
  - dithering, 295
  - LCDs, polarizing filters, 152-153
  - monitors, phosphor current streams, 147
  - subtractive color, 294
  - super-VGA monitors, 150
 command descriptor block, see CDB commands
  - CISC (complex instruction set computing) processors, 56
  - RISC (reduced instruction set computing) processors, 57
 common ground connection, serial ports, 130  
 complex instruction set computing, see CISC  
 compressed volume file (CVF), 97  
 compression
  - audio signals, 227
  - bitmaps, RLE (run-length encoding), 146
  - disk compression, 96-97
  - file compression, 98-99
  - hard drives, 95

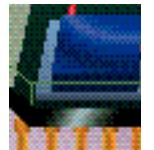




- lossless compression, 98
  - lossy compression, graphics, 99
  - MPEG (Motion Pictures Expert Group), 225
  - video, 226-227, 264-265
  - conflicts, 27
  - connections, 134
  - constant angular velocity, magnetic disks, 207
  - constant linear velocity, CD-ROMs, 207
  - contact switches, joysticks, 167
  - contiguous clusters, 100
  - continuous-tone printing, 293
  - control gates, flash memory transistors, 49
  - controller board, 3.5-inch floppy drives, 85
  - controllers, hard drives, 88
  - convergence, 191
  - cookies
    - 3.5-inch floppy drives, 84
    - Zip disks, 104
  - cooperative multitasking, memory, 33
  - coprocessors, 3-D-graphics, 237
  - corrupted packets, 265
  - CPUs (central processing units), 5, 59
    - branch predictor unit, 61
    - Pentium II
      - cache, 66
      - Celeron and Xeon processors, 67
      - MMX units, 67
    - Pentium III, 66
    - Pentium MMX, 64
      - ALUs, 65
      - FPU (floating point units), 65
      - internal caches, 64
      - SIMD process (single instruction/multiple data), 65
    - TLB (translation lookaside buffers), 64
    - program counter, 6
  - CRTs (cathode-ray tubes), 150
  - CTS (Clear to Send) signal, modems, 174
  - Curie point, magneto-optical drives, 106
  - CVF (compressed volume file), 97
  - Cyberpuck, 165
  - Cyclic redundancy check (CRC), quarter-inch cartridge (QIC) tape backup drive, 110
- D**
- D-cache (data cache), 63
  - D2T2 (dye diffusion thermal transfer) printers, 293
  - DAC (digital-to-analog converters), 227
    - 56K modems, 177
    - converting pixel values, 147
    - S-VGA adapters, 150
    - sound cards, 218
  - Daisy chains, 135
  - DAT (digital audio tape), restoring files, 113
  - data
    - erasing, annealing phase (DVD-RAM), 213
    - reading, RAM (random access memory), 46-47
    - writing, to RAM (random access memory), 44-45
  - data bits, modems, 173
  - data cache (D-cache), 63
  - data gloves (virtual reality), 231
  - data lines, RAM (random access memory), 44
  - data packets, modems, 173
  - Data Set Ready signal, (DSR) signal, 172
  - Data Terminal Ready signal, (DTR) signal, 172
  - data transfers
    - over Internet, 251, 254-255
    - packets, 254
    - priorities, 133
    - protocols, 254
  - data-link layer, network communications, 252
  - decimal numbers, 75
  - decode unit, 56-57
  - Pentium processors, 61

decompression, hard drives, 95  
 defragmenting  
     disks, 100-101  
     hard drives, 95  
 deleting, files from disks, 78  
 Delta video, 226  
 demodulating, modems, 171  
 detectors, CD-ROMs, 206  
 device drivers, 23-25, 29  
 dictionary, file compression, 98  
 digital audio-tape (DAT) backup drive, tape  
     drives, 112-113  
 digital devices, PCs, 171  
 digital light processing, 156-157  
 digital photography, 191  
     ADCs (analog-to-digital converters), 199  
     CCDs (charge-coupled devices), 198  
     DSPs (digital signal processors), 199  
     image storage, 199  
 digital signal processing, 179  
 digital signal processors, see DSPs  
 digital video discs, (DVDs), 205-211  
 digital-to-analog converter, see DAC  
 digitizing tablets, pointing devices, 159  
 DIMMs (Dual In-line Memory Modules), 47  
 diodes  
     3.5-inch floppy drives  
         light-emitting, 85  
         photo-sensitive, 85  
     flatbed scanners, 185  
 DIP switches, PC Cards, 180  
 direct memory access, see DMA  
 disk caching, see caching  
 disk defragmenting, 100-101  
 disk operating system, see DOS  
 disks  
     3.5-inch disks, 83  
     5.25-inch floppy disks, 84  
     deleting files, 78

formatting, 76  
 reading  
     bits, 74  
     files from, 80-81  
 writing, 78  
     bits, 74  
     clusters, 79  
     end of file marker, 79  
 dispatch/execution unit, uops, (Pentium Pro), 63  
 displays  
     digital light processing, 156-157  
     gas plasma, 154-155  
     LCDs (Liquid Crystal Displays), 152-153  
         on palm PCs, 192-193  
 monitors  
     additive colors, 143  
     Ben Day dots, 143  
     bitmaps, 143-147  
     LCDs (Liquid Crystal Displays), 154-155  
         phosphor current streams, 147  
         super-VGA, 150-151  
         see also gas plasma displays  
 dithering, colors, 295  
 DMA (direct memory access), system  
     resources, 27  
 DNS (Domain Name System), Web  
     browsers, 269  
 domain names, 251  
     URLs, 268  
 domain name server, e-mail, 258  
 DOS (disk operating system), 73  
     FAT (file allocation table), 76  
 dot pitch, CRTs (cathode-ray tubes), 151  
 dot-matrix printers, 285-287  
     print head, printing pins, 287  
 dots, color printing, 295  
 double buffering, 237  
 drain, silicon, 40





DRAM (dynamic random access memory), 47  
 drawing, vector graphics, 148-149  
 drive arrays  
     hard drives, 87  
     mirrored drive arrays, 90-91  
     striped drive arrays, 92-93  
 drivers, 23  
 drives, CD-ROMs, 206-207  
 drum, see OPC (organic photoconducting cartridge)  
 DSPs (digital signal processors), 199  
     MIDI (musical instrument digital interface), 220-221  
     sound cards, 219  
 DSR (Data Set Ready) signal, modems, 172  
 DTR (Data Terminal Ready) signal, modems, 172  
 Dual In-line Memory Modules (DIMMs), 47  
 dual-ported memory, 237  
 duplexes, modems, 173  
 DVD-RAM, 212-213  
 DVDs (digital video discs), 205, 210-211  
 dye diffusion thermal transfer (D2T2)  
     printers, 293  
 dye-sublimation printers, 300-301  
 dynamic random access memory (DRAM), 47

## E

e-mail, 257-259  
 ECCs (Error Correcting Codes), 47  
 EDO RAM (Extended Data Out random access memory), 47  
 EIDE (Enhanced Integrated Device Electronics), 127  
 EIDE (Enhanced Integrated Device Electronics) connections, 134  
 EISA (Extended Industry Standard Architecture) bus, 120

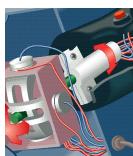
electronic mail, see e-mail, 257  
 encoders, mouse, 160  
 End of file marker, writing to disk, 79  
 Enhanced Intregrated Device Electronics, see EIDE  
 enumerators  
     operating systems, 28  
     programmable registers, 29  
 EPROM (erasable, programmable, read-only memory), BIOS (basic input/output systems), 25  
 EPROM memory, replacing with flash memory, 48-49  
 erasable, programmable, read-only memory, see EPROM  
 eraserheads, pointing devices, 159  
 erasing, data, annealing phase (DVD-RAM), 213  
 ergonomics, keyboards, 139  
 Error Correcting Codes (ECCs), 47  
 Error Correction (EC)  
     parity bits, modems, 173  
     tape drives, 111  
 Ethernet networks, 243-245  
     nodes, 245  
     transceivers, 244  
 execution unit, Pentium processors, 61  
 expansion bus, see buses  
 expansion cards, 122-123  
     buses, 119  
 Expansion slots, 119  
     ISA (Industry Standard Architecture)  
         expansion slots, 120  
 Extended Data Out random access memory (EDO RAM), 47  
 Extended Industry Standard Architecture, EISA bus, 120

# F

FAT (file allocation table), 76, 89  
 DOS, 76  
 feature extraction, OCR (optical character recognition), 189  
 fetch/decode unit, Pentium Pro, 63  
 FID (Fingerprint Identification) software, 196-197  
 fields, CRT raster scanning, 151  
 file allocation table, see FAT  
 file compression  
   dictionary, 98  
   LZ (Lempe and Ziv) adaptive dictionary-based algorithm, 98  
   zipping files, 98  
 file server, 243  
 files  
   attaching, to e-mail, 257  
   bitmapped  
     color, 145  
     extensions, 144  
     header signature IDs, 144  
     pixel location data, 144  
     pixels, 147  
     RLE (run-length encoding), 146  
   compressing, videos, 264-265  
   deleting, from disks, 78  
   fragmented, 100  
   reading, from disks, 80-81  
   restoring  
     backups, 111  
     from DAT (digital audio tape), 113  
   zipping, 98  
 fills, drawing vector graphics, 149  
 filters, polarizing, color LCDs, 152-153  
 fingerprint recognition, 191  
   IVS (identity verification system), 197  
 software, 196

firing chambers, color ink-jet printers, 296-297  
 firmware, fonts, printers, 282  
 flash memory, 48-49  
   RAM (random access memory), 43  
   transistors, floating gates, 48-49  
 flat memory, Windows 95 and 98, 31  
 flatbed scanners, 183-185  
 floating gates, flash memory transistors, 48  
   channel hot electron injection, 48  
   control gates, 49  
 floating point numbers, processors, 59  
 floating point unit, Pentium processors, 61  
 floppy disks, see disks  
 floppy drives, 83  
 floptical disks, 107  
 floptical drives, 107  
 FM synthesis, sound cards, 221  
 focusing coil, CD-ROMs, 206  
 fogging, 235  
 fonts  
   ASCII, 281  
   attributes, 280  
   bitmapped, 279-281  
   outline, 279, 282-283  
   printer commands, 283  
   printers, firmware, 282  
 force-feedback joysticks, 165, 168-169  
 formatting  
   disks, 76  
   magnetic disks, 73  
 FPUs (floating point units)  
   Pentium MMX processors, 65  
   registers, 65  
 fragmented files, 100  
 frame buffers, video adapter memory, 147  
 full polygon animation, 236





full-adders, 39  
gates, 53  
full-duplex, modems, 173  
functions of  
disk compression, 96-97  
file compression, 98-99  
hard drives, 88-89  
mirrored drive arrays, 90-91  
read/write heads, 74-75  
striped drive arrays, 92-93  
fusing system, print engines, laser  
printers, 290

## G

game controllers, 165  
see also joysticks  
gamma correction, hand-held scanners, 187  
gas plasma displays, monitors, 154-155  
see also LCDs (Liquid Crystal Displays)  
gates  
full-adders, 53  
half-adders, 53  
transistors, 52  
gateway software, e-mail, (America Online), 257  
gateways  
e-mail, 258  
Internet, data transfer, 255  
GDI (Graphics Device Interface), 32  
gesture recognition, 191  
global positioning systems (GPSs), 191, 194  
broadcast signal timing, 195  
radio receivers, 195  
GPSs (global positioning systems), 191, 194  
broadcast signal timing, 195  
radio receivers, 195  
graffiti character set, PalmPilot (3Com), 193

graphics  
3-D, see 3-D graphics  
bitmapped files, 144  
bitmaps, 145  
compression, lossy compression, 99  
vector, 148  
changing attributes, 149  
point locations, 148  
viewing transform, 149  
Graphics Device Interface (GDI), 32  
group coding, modems, 173

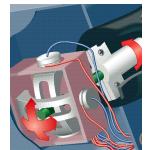
## H

half-adders, 39  
gates, 53  
half-duplex, modems, 173  
hand-held scanners, 183, 186-187  
ADC (analog-to-digital converters), 187  
gamma correction, 187  
handshake, modems, 173  
handwriting recognition, palm PCs, 193  
hard disks, see hard drives  
hard drives  
access time, 95  
caching, 95  
compression, 95  
decompression, 95  
defragmenting, 95  
drive arrays, 87  
functions of, 88-89  
history of, 87  
platters, 87  
speed of, 95  
storage, 95  
hardware tree, 29  
head actuator, hard drives, 88

headers, bitmaps, signature IDs, 144  
 helmets (virtual reality), 231  
 hidden views, 233-234  
 history of, hard drives, 87  
 horizontal positions, vector graphics, 148  
 host hub, universal serial bus (USB), 132  
 hotswapping, Plug and Play, 27  
 HS (High Speed), modems, 175  
 HTML, URLs, 268  
 http, URLs, 268  
 hubs  
   3.5-inch floppy drives, 84  
   Internet, data transfer, 255  
   star networks, 248  
 hyperlinks, World Wide Web (WWW), 268

I cache (instruction cache), 63  
 IDE (integrated drive electronics), 24  
   connections, 134  
   hard drives, 88  
 images, storing  
   digital photography, 199  
   vector, 148  
     changing attributes, 149  
     point locations, 148  
     viewing transform, 149  
 immersion (virtual reality), 229-230  
 infrared lasers, CD-RW (CD-Rewriteable), 213  
 initiator, SCSI controllers, 136  
 ink-jet printers, 293  
   see also color ink-jet printers  
 input/output controllers, 119  
 instruction cache (I cache), 63  
 instruction prefetch buffer, Pentium  
   processors, 61  
 instructions, SIMD (Pentium MMX  
   processors), 65

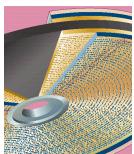
integrated drive electronics, see IDE  
 integrated services digital network, see ISDN  
   (integrated services digital network)  
 interframe video compression, 264  
   corrupted packets, 265  
   IP multicast packets, 265  
   UDP packets, 265  
 interlacing, super-VGA monitors, 151  
 Internet, 251  
   audio, 261  
     bandwidth, 261  
     streaming, 261-263  
   data transfers, 251, 254-255  
   video, 261, 264-265  
     bandwidth, 261  
     capture cards, 264  
     corrupted packets, 265  
     IP multicast packets, 265  
     streaming, 261  
     UDP packets, 265  
   World Wide Web (WWW), Internet, 267  
 Internet access, 254  
 Internet provider (IP), 269  
 interrupt, 20-21  
 interrupt controller, 20-21  
 interrupt return (IRET), 21  
 interrupt table, 21  
 interrupt transfers, second highest  
   priority, 133  
 interrupt vectors, 21  
 interrupt (IRQ), 27  
 IP (Internet provider), Web browsers, 269  
 IP multicast packets, 265  
 IRET (interrupt return), 21  
 IRQ, 27  
 ISA (Industry Standard Architecture)  
   bus, 119  
   expansion cards, 122  
   expansion slots, 120



ISDN (integrated services digital network), 227  
 modems, 171  
 isochronous, highest priority, 133  
 IVS (Identity Verification System), fingerprint recognition, 197

**J**

Jot character set, Windows CE, 193  
 joysticks, 159, 165-167  
   force-feedback, 165, 168-169  
   sensing positions, piezo-electric sensor, 167  
   top hat, 167  
   X-Y axes coordinates, 166  
   Y-adapter, 166  
   yokes, 166  
 jukeboxes, 214-215  
 jump execution unit, Pentium Pro, 62  
 jumpers, PC Cards, 180

**K**

key bytes, RLE, 146  
 keyboards, 139-141  
   ergonomics, 139  
   scan codes, 140-141

**L**

L1 (level 1), caches, 63  
 L1 cache, Pentium II processors, 66  
 L2 cache  
   Celeron and Xeon processors, 67  
   Pentium II processors, 66

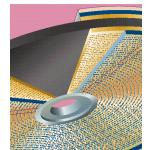
LAN (local area network), 243, 254  
   e-mail, 257  
 landing pads, digital light processing, 157  
 lands  
   CD-ROMs, 206  
   DVDs, 210  
 laptops  
   gas plasma displays, 154-155  
   LCDs (Liquid Crystal Displays), 152-153  
 laser printers, 289-291  
   OPC (organic photoconducting cartridge), 291  
   print engines, 289  
   fusing systems, 290  
 lasers, 103  
 LCDs (Liquid Crystal Displays), 152-153, 230  
   colors, 152-153  
 palm PCs, 192  
   handwriting recognition, 193  
   showing ink, 193  
 projectors, digital light processing, 156  
   see also gas plasma displays; monitors  
 level 1 (L1), caches, 63  
 light-emitting diodes (LED)  
   3.5-inch floppy drives, 85  
   force-feedback joysticks, 169  
   hand-held scanners, 186  
 light-sensing diodes, CD-ROMs, 207  
 Liquid Crystal Displays (LCDs), 152-153, 230  
 list server, e-mail, 257  
 local area network, see LAN  
 local bus, 120  
   PCI (Peripheral Component Interconnect),  
     120-121, 124  
 Video Electronics Standards Association (VESA), 120

local control loop, force-feedback joysticks, 169  
 logic board, hard drives, 88  
 logic gates  
   AND gates, 53  
   half adders, 39  
   NOT gates, 52  
   OR gates, 52  
   transistors, 39  
   XOR gates, 53  
 lossless compression, 98  
 lossy compression, videoconferencing, 226  
 LZ (Lempe and Ziv) adaptive dictionary-based algorithm, file compression, 98-99

## M

machine language, ASCII, 43  
 magnetic deflection yokes, super-VGA monitors, 150  
 magnetic disks, 73  
   constant angular velocity, 207  
   formatting, 73, 76  
   magnetic film, 74  
   see also disks  
 magnetic drives, read/write heads, 73-74  
   see also drives  
 magnetic film, magnetic disks, 74  
 magneto-optical disks, 106  
 magneto-optical drives, 106  
 mail reflector, e-mail, 257  
 mailing lists, e-mail, 257  
 master connections, 134  
 math operations, 51-53  
 matrix matching, OCR (optical character-recognition), 188  
 MCA (Microchannel Architecture), 120  
 megahertz (MHz), 52

memory  
   cooperative multitasking, 33  
   dual-ported, 237  
   EPROM, replacing with flash memory, 48-49  
   flat memory, Windows 95 and 98, 31  
   processors, 31  
   RAM (random access memory)  
     address lines, 44  
     capacitors, 45-47  
     data lines, 44  
     DIMMs (Dual In-line Memory Modules), 47  
     DRAM (dynamic random access memory), 47  
     ECCs (Error Correcting Codes), 47  
     EDORAM (Extended Data Out random access memory), 47  
     flash memory, 43, 48-49  
     reading data from, 46-47  
     SDRAM (Synchronous DRAM), 47  
     segmented memory (DOS), 31  
     sharing in Windows, 31  
     SIMMs (Single In-line Memory Modules), 47  
     SRAM (Static random access memory), 47  
     switches, 45-47  
     transistors, 44  
     virtual memory, 31, 33  
     VRAM (video random access memory), 47  
     windows, 31-33  
     writing to, 44  
 memory addresses, PC Cards, 180  
 memory registers, 6  
   PC Cards, 180  
 memory windows, PC Cards, 180





- metafiles, streaming audio, 262
    - decoding packets, 263
    - launching players, 263
    - transferring packets, 263
  - metatext, search engines, 273
  - micro-operations, uop, 63
  - Microchannel Architecture (MCA), 120
  - microchips
    - Pentium, 55
    - transistors, 39
  - microcode, 56
  - microprocessor, 6
  - Microsoft Baseball 3-D, 236
  - MIDI (musical instrument digital interface), 220-221
    - MIP mapping (multim in parvum), 234
  - mirror sites, Web browsers, 269
  - mirrored drive arrays, 90-91
  - mirrors, as pixels (digital light processing), 157
  - MMX units, Pentium II processors, 67
  - MO, see magneto-optical
  - modem lights, 175
  - modems, 171-174
    - demodulating, 171
    - ISDN (integrated services digital network), 171
    - lights, 175
    - modulating, 171
    - see also 56K modems
  - monitors
    - additive colors, 143
    - Ben Day dots, 143
    - bitmaps, 143-144
      - colors, 145
    - file extensions, 144
  - pixel locations, 144
  - pixels, 143, 147
    - RLE (run-length encoding), 146
  - phosphor current streams, 147
  - super-VGA, 150-151
    - see also gas plasma displays; LCDs (Liquid Crystal Displays)
  - motherboards, 23
  - motion capture, 236
  - Motion Pictures Expert Group (MPEG), video, 226
  - mouse, 159
    - encoders, 160
    - pointing devices, 160-161
  - MR (Modem Ready), modems, 175
  - mu-law codec, 56K modems, 177
  - multimedia sound, see sound
  - multimedia video, 226
  - multipurpose Internet mail extensions, see MIME
  - multim in parvum (MIP) mapping, 234
  - musical instrument digital interface, (MIDI), 220-221
- N**
- N-type silicon, 40
  - nanoprocessors, 56
  - NAP (network access point), 255
  - network communications, 252-253
  - network layers, network communications, 252
  - network topologies, 243
  - networks
    - clients, 243
    - Ethernet, 244-245
    - nodes, 243
    - peer-to-peer, 243
    - print servers, 243
    - stars, 248-249
    - token rings, 246-247

nodes  
 Ethernet networks, 245  
 networks, 243  
 repeaters, token ring networks, 247  
 star networks, 248-249  
 transceivers, Ethernet networks, 244  
 nonvolatile flash memory, 48-49  
 NOT gates, logic gates, 52  
 notebooks, see laptops

**O**

OCR (optical character-recognition), 183, 188-189  
 OH (Off-Hook), modems, 175  
 OPCs (organic photoconducting cartridges), laser printers, 291  
 operating systems, 9, 23  
 BIOS (basic input/output systems), 19-21, 25  
 enumerators, 28  
 functions of, 19  
 Windows CE, 192  
 optical character-recognition (OCR), 183, 188-189  
 optical gray-scale position sensor, joysticks, 167  
 optimization, see defragmenting  
 OR gates, logic gates, 52  
 organic photoconducting cartridge, see OPC  
 outline fonts, 279, 282-283

**P**

P-type silicon, 40  
 packets  
 data transfers, Internet, 254  
 streaming audio, 263  
 video compression, 265

page description language (PDL), 279  
 palm PCs, 192  
 LCDs (Liquid Crystal Displays), 192-193  
 PalmPilot (3Com), 192  
 Windows CE, 192  
 PalmPilot (3Com), 192  
 Graffiti character set, 193  
 Paper train, laser printers, 291  
 parallel ports, 127-129  
 parity bit  
 error correction, modems, 173  
 XOR operation, 92  
 passive-matrix color LCDs (Liquid Crystal Displays), 152-153  
 PC Cards, 179-181  
 nonvolatile flash memory, 48-49  
 PC Card slots, 179  
 PCI (Peripheral Component Interconnect)  
 local bus, 120-121, 124  
 PCMCIA (Personal Computer Memory Card Interface Association) Cards, see PC Cards  
 PCs, digital devices, 171  
 PDAs (personal digital assistants), see palm PCs  
 PDL (page description language), 279  
 peer-to-peer networks, 243  
 Pentium II processors, 59, 66  
 caches, 66  
 Celeron and Xeon processors, 67  
 MMX units, 67  
 Pentium microchips, 55  
 Pentium MMX processors, 64-65  
 Pentium Pro processors, 62-63  
 Pentium processors, 60-62  
 Peripheral Component Interconnect, see PCI  
 personal digital assistants, see palm PCs  
 perspective correction (virtual reality), 234





- Phase change technology, rewriting to disc, 212
- phonemes, voice recognition, 223
- phosphor current streams, color monitors, 147
- phosphors, CRTs (cathode-ray tubes), 151
- photo-quality printers, dye-sublimation, 300-301
- photo-sensitive diode, 3.5-inch floppy drives, 85
- physical layers, network communications, 253
- piezo controller, solid-ink color printers, 302
- piezo-electric sensor, joysticks, 167
- pits
- CD-ROMs, 206
  - DVDs, 210
- pixelation, 234
- pixels, 143
- bitmaps, turning on and off, 144
  - color bitmaps, bit requirements, 145
  - location data, 144
  - mirrors, as digital light processing, 157
  - values, 147
  - virtual, 145
- platters, hard drives, 87
- players, audio (launching with metafiles), 263
- Plug and Play, 27-29
- hotswapping, 27
- Plug-ins, audio players (launching with metafiles), 263
- point locations, drawing vector graphics, 148
- pointing devices
- digitizing tablets, 159
  - eraserheads, 159
  - joysticks, 159
  - mouse, 160-161
  - pointing stick, 163
  - see also mouse
- touch screens, 159
- touchpads, 159, 162
- trackballs, 159, 161
- polarization, magneto-optical drives, 106
- polarizing filters, color LCDs, 152-153
- polygons
- 3-D graphic acceleration, 232-234
  - full animation, 236
- POP (post office protocol), e-mail, 257, 259
- portable computers
- gas plasma displays, 154-155
  - LCDs (Liquid Crystal Displays), 152-153
  - palm PCs, 192-193
  - PC Card slots, 179
  - PC Cards, nonvolatile flash memory, 48-49
- ports, 127
- IDE (integrated drive electronics), 134
  - EIDE (Enhanced Integrated Device Electronics), 127, 134
  - parallel ports, 127-129
  - SCSI (Small Computer System Interface), 127, 135-137
  - serial ports, 127, 130-131
  - universal serial bus (USB), 132-133
- possible colors, super-VGA monitors, 150
- POST (power-on self-test), boot-up, 5-7
- post office protocol, see POP (post office protocol)
- Pot, see potentiometer
- potentiometer, joysticks, 167
- power-on self-test, see POST (power-on self-test)
- presentation layers, network communications, 252
- print cartridge, color ink-jet printers, 296

print commands  
     algorithms, 282  
     bitmapped fonts, 280-281  
 print engines, laser printers, 289-290  
 print head, printing pins (dot matrix printers), 287  
 print servers, networks, 243  
 printers  
     bitmap tables, 286  
     color ink-jet, 296-297  
     color laser, 293  
     color thermal, 293, 298-299  
     commands, fonts, 283  
     dot-matrix printers, 285-287  
     dye diffusion thermal transfer (D2T2), 293  
     dye-sublimation, 300-301  
     fonts, firmware, 282  
     ink-jet, 293  
     laser printers, 289-291  
     RAM (random access memory), 286  
     solid-ink, 293  
     solid-ink color, 302-303  
     write-black, 291  
     write-white, 291  
 printing  
     colors, dots, 295  
     in color, 294  
 printing pins, 287  
 priorities, data transfers, 133  
 processors, 20-21  
     CPUs (central processing units), 59  
     floating point numbers, 59  
     memory, 31  
     Pentium II, 59  
 program counter, 6  
 programmable registers, enumerators, 29  
 protocols, data transfers (over Internet), 254

proxy servers, Web browsers, 269  
 public switched telephone network, 56K  
     modems, 177  
 pulse code modulation, 56K modems, 177

## Q-R

quarter-inch cartridge (QIC) tape backup drive, 110-111  
 R-axis, joysticks, 167  
 radio receivers, GPSs, 195  
 RAID (redundant array of inexpensive drives), 90  
 RAM (random access memory)  
     address lines, 44  
     capacitors, closed, 45-47  
     data, writing, 44-45  
     data lines, 44  
     DIMMs (Dual In-line Memory Modules), 47  
     DRAM (dynamic random access memory), 47  
     ECCs (Error Correcting Codes), 47  
     EDO RAM (Extended Data Out random access memory), 47  
     flash memory, 43, 48-49  
     hardware tree, 29  
     printers, 286  
     reading data from, 46-47  
     SDRAM (Synchronous DRAM), 47  
     SIMMs (Single In-line Memory Modules), 47  
     SRAM (Static random access memory), 47  
     switches, 45-47  
     transistors, 44  
     VRAM (video random access memory), 47  
 random access, video, 225





raster scanning, CRTs (cathode-ray tubes), 151  
**RD** (Receive Data), modems, 175  
 read-only memory (ROM), 6  
 read/write heads  
   3.5-inch floppy drives, 85  
   function of, 74-75  
   magnetic drives, 73-74  
 reading  
   bits, from disks, 74  
   data, RAM (random access memory), 46-47  
   files, from disks, 80-81  
 Receive Data line, modems, 172  
 receivers, GPSs, 195  
 red lasers, DVD-RAM, 213  
 reduced instruction set computing, see RISC  
 redundant array of inexpensive drives, see RAID  
 refresh rates, super-VGA monitors, 150  
 registers  
   FPU, 80-bit, 65  
   Pentium processors, 61  
 removable storage, 103  
 rendering engines, 233  
 repeaters  
   Internet, data transfers, 255  
   nodes, token ring networks, 247  
 Request to Send signal, see RTS (Request to Send) signal  
 Request line, 137  
 resource arbitration, hardware tree, 29  
 restoring  
   backed up files, 111  
   files, from DAT (digital audio tape), 113  
 retirement unit, Pentium Pro, 62  
 ribbon cables, SCSI (Small Computer System Interface), 135

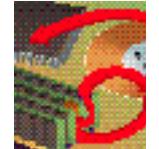
RISC (reduced instruction set computing)  
 processors, 55  
   commands, 57  
 RLE (run-length encoding), key bytes, 146  
 (ROM), read-only memory, 6  
 routers  
   e-mail, 258  
   Internet access, 254  
   Web browsers, 269  
 RS-232 ports, see serial ports  
 RTS (Request to Send) signal, modems, 174  
 Run-length encoding, see RLE

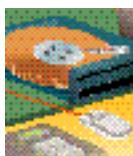
## S

satellites, GPSs, 194-195  
 scan codes, keyboards, 140-141  
 scanners, 183  
   degradation, OCR (optical character-recognition), 188  
   fingerprint recognition  
     IVS (Identity Verification System), 197  
     software, 196  
   flatbeds, 183-185  
   hand-held, 183, 186-187  
   sheet-fed, 183  
 SCSI (Small Computer System Interface), 127, 135-137  
 SCSI bus, daisy chains, 135  
 SCSI controller card, 7  
 SCSI controllers, initiators, 136  
 SD (Send Data), modems, 175  
 SDRAM (synchronous dynamic random access memory), 47, 125  
 search engines, 272-273  
 searching, World Wide Web (WWW), 267  
 sectors, magnetic disks, 73, 76

segmented memory, DOS, 31  
 select line, parallel ports, 128  
 sending e-mail, 258-259  
 sensing positions, piezo-electric sensor (joysticks), 167  
 serial ports, 127, 130-131  
     common ground connection, 130  
 server software, e-mail, 257  
 session layers, 253  
 Shadow masks, super-VGA monitors, 151  
 sheet-fed scanners, 183  
 showing ink, palm PC LCD displays, 193  
 shutter, 3.5-inch floppy drives, 84  
 signature IDs, bitmap headers, 144  
 silicon, 40  
 silicon transistors, 40  
 SIMD (single instruction/multiple data),  
     Pentium MMX processors, 65  
 SIMMs (Single In-line Memory Modules), 47  
 simple mail transfer protocol, (SMTP),  
     257-258  
 single/instruction/multiple data (SIMD), 65  
 sites, World Wide Web (WWW), 268  
 six degrees of freedom (6DOF), 231  
 slack, clusters, 96  
 slave, connections, 134  
 Small Computer System Interface, see SCSI  
 SMTP (simple mail transfer protocol), e-mail,  
     257-258  
 software, fingerprint recognition, 196  
     IVS (Identity Verification System), 197  
 solenoids, printing pins, 287  
 solid-ink color printers, 302-303  
     piezo controller, 302  
 solid-ink printers, 293  
 sound, 217  
     MIDI (musical instrument digital interface), 220-221  
     sound cards, 218-219  
     voice recognition, 217

sound cards, 218-219  
     ADC (analog-to-digital converter) chip,  
         218-219  
     DAC (digital-to-analog converter) chip,  
         218  
     DSP (digital signal processor), 219  
     FM synthesis, 221  
     wave-table synthesis, 220  
 source, silicon, 40  
 speculative execution, uops, 63  
 speech recognition, see voice recognition  
 speed, 95  
 spelling checker, OCR (optical character-recognition), 189  
 spiders, see Web crawlers  
 spindles, hard drives, 88  
 sprites, 236  
 SRAM (static random access memory), 47  
 star networks, 243, 248-249  
 start/stop bits, modems, 173  
 static random access memory (SRAM), 47  
 stepper motor, 3.5-inch floppy drives, 84  
 storage  
     compression, 95  
     devices, 205  
     lasers, 103  
     on hard drives, 95  
 store buffers, Pentium Pro, 62  
 storing, images (digital photography), 199  
 streaming  
     Internet audio, 261  
         decoding packets, 263  
         metafiles, 262-263  
         transferring packets, 263  
         UDP (user database protocol), 261  
     Internet video, 261, 264-265  
         capture cards, 264  
         corrupted packets, 265





IP multicast packets, 265  
 UDP (user database protocol), 261  
 UDP packets, 265  
 striped drive arrays, 92-93  
 strobe, parallel ports, 128  
 stylus, palm PC displays, 192  
 sublimation, 300  
     photographic quality, 301  
 subtractive color, 294  
 super-VGA monitors  
     adapters, DAC (digital-to-analog converters), 150  
     CRTs (cathode-ray tubes), 150-151  
 switches, RAM (random access memory), 45-47  
 synchronous dynamic random access memory, see SDRAM  
 system files, 9  
 system resources, 27

## T

T-connection, Internet access, 254  
 tape backups, 109  
 tape drives, 109  
     digital audio-tape (DAT) backup drive, 112-113  
     Error Correction (EC), 111  
     quarter-inch cartridge (QIC) tape backup drive, 110-111  
 TCP (transmission control protocol), 261  
 telephone systems, analog devices, 171  
 terminator, daisy chains, 135  
 tessellation, 232-237  
 texture maps, 234  
     alpha blending, 235  
     caching, 237  
 TI connection, Web browsers, 269

TLB (translation lookaside buffers), 64  
 token ring adapter cards, networks, 246  
 token ring networks, 243, 246-247  
     nodes, repeaters, 247  
     token ring adapter cards, 246  
     tokens, 246  
 tokens  
     force-feedback joysticks, 168  
     token ring networks, 246  
 toner, laser printers, 291  
 top hat, joysticks, 167  
 top-level domain name, URLs, 268  
 topologies, networks, 243  
 torsion hinges, 156  
 touch pads, pointing devices, 159, 162  
 touch screens, pointing devices, 159  
 TR (Terminal Ready), modems, 175  
 traces, buses, 119  
 trackballs, pointing devices, 159, 161  
 tracks  
     clusters, 76  
     magnetic disks, formatting, 73, 76  
 transceivers, nodes (Ethernet networks), 244  
 transistors, 40, 44  
     binary numbers, 39  
     flash memory, floating gates, 48-49  
     gates, 52  
     logic gates, 39  
     microchips, 39  
     silicon, 40  
     switches, 45-47  
 translation lookaside buffers (TLB), 64  
 transmission control protocol (TCP), 261  
 transmission speeds, modems, 173  
 Transmit Data line, modems, 172  
 transport layers, network communications, 253

triangulation, 232-237  
 trilinear filtering, 235  
 tunneling, see channel hot electron injection  
 two frame buffers, 237

**U**

UDP (user database protocol), 261  
 UDP (user database protocol) packets, 265  
 universal resource locators, see URLs  
 universal serial bus (USB), 127, 132-133  
 universal serial bus connections, joysticks, 167  
 uops  
   dispatch/execution unit, Pentium Pro, 63  
   micro-operations, 63  
   speculative execution, 63  
 URLs (universal resource locators), 268  
   search engines, 272  
 USB, see universal serial bus  
 user database protocol (UDP), 261  
 user immersion (virtual reality), 229-230

**V**

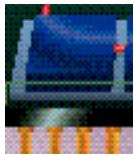
V.90 modems, see 56K modems  
 vector fonts, see outline fonts  
 vector graphics, 148-149  
 vertical positions, vector graphics, 148  
 VESA local bus, see VL-Bus  
 VESA, see Video Electronics Standards Association  
 VFAT (virtual file allocation table),  
   Windows 95, 76, 89  
 video, 225  
   ADC (analog-to-digital converter), 227  
   compression, 226-227  
   delta, 226

Internet, 261, 264-265  
 bandwidth, 261  
 capture cards, 264  
 corrupted packets, 265  
 IP multicast packets, 265  
 streaming, 261  
 UDP packets, 265  
 MPEG (Motion Pictures Expert Group), 226  
 multimedia, 226  
 random access, 225  
 video adapters  
   DAC (digital-to-analog converters),  
     converting pixel values, 147  
   frame buffers, storing pixel values, 147  
   super-VGA color storage, 150  
 Video Electronics Standards Association (VESA), local bus, 120  
 video random access memory (VRAM), 47  
 videoconferencing, lossy compression, 226  
 viewing graphics, 149  
 virtual drives, 97  
 virtual file allocation table, see VFAT  
 Virtual Machine Manager (VMM),  
   windows, 32  
 virtual memory, 31, 33  
 virtual pixels, 145  
 virtual reality, 229-230  
   accelerated 3-D graphics, 232-237  
   accelerometers (controllers), 230  
   data gloves, 231  
   fogging, 235  
   helmets, 231  
 VL-Bus, 120-121  
 VMM, see Virtual Machine Manager  
 voice recognition, 191, 217, 222-223  
 VRAM (video random access memory),  
   47, 237



## W

WAN (wide area network), 254  
 wave forms, force-feedback joysticks, 168  
 wave-table synthesis, sound cards, 220  
 Web browsers, 268-271  
 Web crawlers, search engines, 272  
 Web search engines, 272-273  
 wide area networks (WAN), 254  
 wide-SCSI, 135  
 windows  
     memory, 31-33  
     sharing, 31  
 Virtual Machine Manager (VMM), 32  
 Windows 95, VFAT (virtual file allocation table), 76  
 Windows CE, Jot character set, 193  
 Windows CE (palm PCs), 192  
 Wire-frame views, 233  
 World Wide Web (WWW)  
     browsing, 267  
     hyperlinks, 268  
     searching, 267  
     sites, 268  
     URLs, 268  
 WORM (Write Once, Read Many), 103  
     CD-R (CD-Recordable), 209  
 write-black printers, 291  
 write-protected, 3.5-inch floppy drives, 85  
 write-white printers, 291  
 writing  
     bits, to disks, 74  
     data, to RAM (random access memory),  
         44-45  
     to disks, 78-79



## X-Z

X-Y axes coordinates, joysticks, 166  
 Xeon processors (Pentium II), 67  
 XOR gates, logic gates, 53  
 XOR operations, 92  
 Y-adapters, joysticks, 166  
 yokes, joysticks, 166  
 Z-buffering, 233-234  
 zip disks, cookies, 104  
 zip drives, 104-105  
 zipping files, 98  
 zone recording, zip drives, 105