

**NATIONAL
BESTSELLER**
MORE
THAN
2,000,000
How It Works
SOLD

BY RON WHITE
ILLUSTRATED BY TIM DOWNS



How Computers Work

THE EVOLUTION OF TECHNOLOGY



TENTH EDITION

QUE

FREE SAMPLE CHAPTER



SHARE WITH OTHERS

**SPECIAL
OFFER
ONLY \$4.99!**

How Computers Work

Interactive Web Version

The *How Computers Work, Interactive Web Version* is an enhanced digital copy of the *How Computers Work* print book, which can be accessed via a web browser. It includes many new and exciting features that take the one-of-a-kind *How Computers Work* print book experience to the next level. Here's a look at just a few of the new features you'll find in the *Interactive Web Version*:

Designed for the Web—This new digital edition of *How Computers Work* is built from the ground up to work in today's most popular web browsers. Whether you use Windows, Mac, or a tablet, you'll find a viewing experience that works for you.

Animations—See first-hand how morphing software morphs a photo or how video compression and resolution changes the quality of a digital video. Watch as an interactive firewall blocks dangerous data, while letting safe packets through. Have you ever wondered just how a transistor physically controls the flow of electrons to turn itself on and off? You'll see it in action here!

Interactive Elements—Directly interact with more than a dozen of the topics covered in this book. Drag data into a processor to see how it divides up a work load and computes an answer. Or use a virtual laser to photoetch a microchip. You'll even control the flow of light through an LCD monitor as you learn how it generates an image!

How to Purchase the Interactive Web Version:

Register this eBook to unlock a single-use discount, which will allow you to purchase a copy of the Interactive Web Version of *How Computers Work* (ISBN: 9780133096781) for only \$4.99, a savings of \$25!*

Follow the steps below:

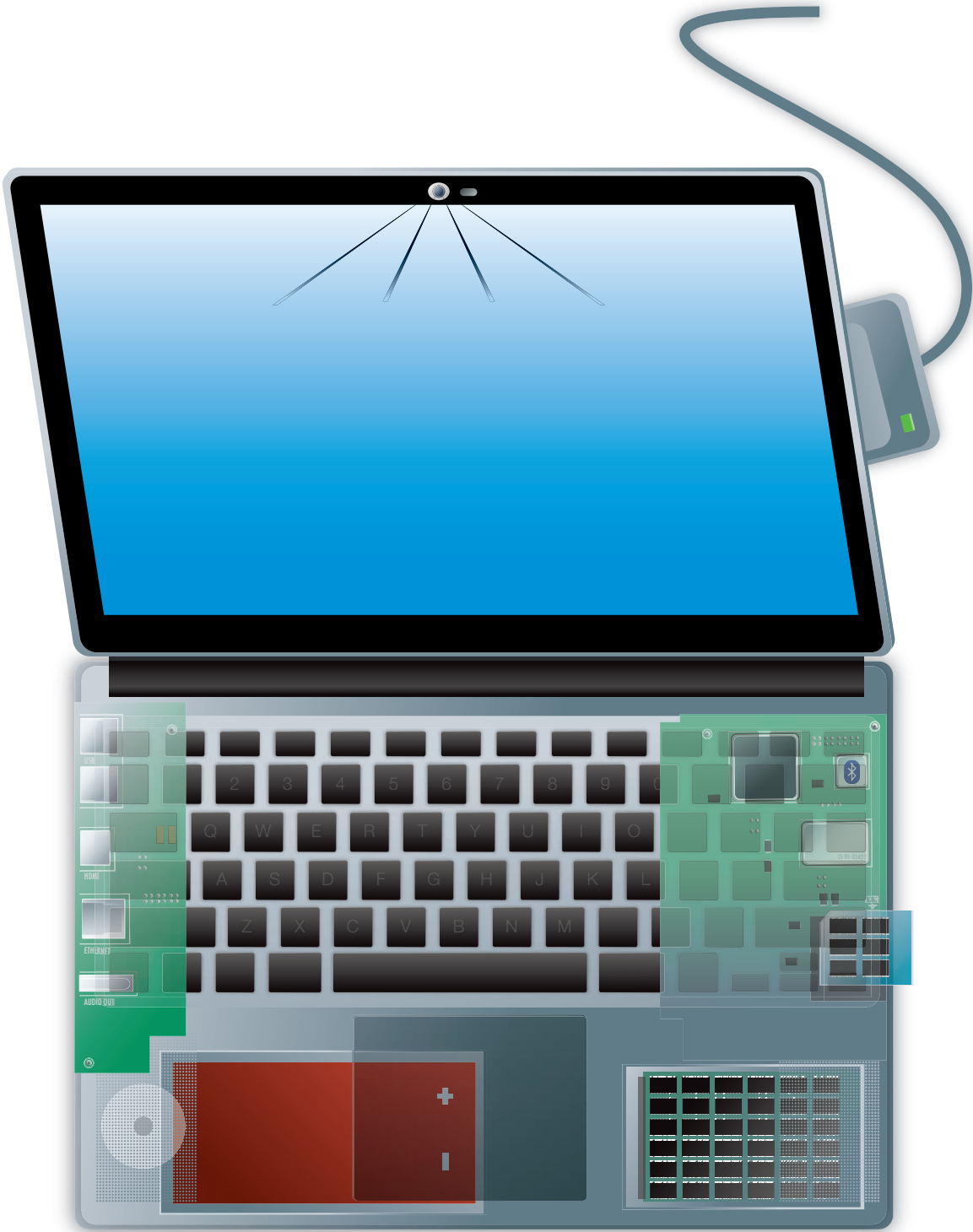
1. Go to quepublishing.com/register and log in or create a new account.
2. Enter the ISBN: 9780789749840.
(NOTE: please enter the print book ISBN provided to register the eBook you purchased)
3. Answer the challenge question as proof of purchase.
4. Click on the "Access Bonus Content" link in the Registered Products section of your account page, to be taken to the page where your discount code is available.
5. Go to www.quepublishing.com/title/9780133096781.
6. Add the product to your Cart.
7. Click on "Proceed To Checkout".
8. Enter your discount code in the "Enter a New Discount Code" box.
9. Complete payment information.

Discount may not be combined with any other offer and is not redeemable for cash. Offer subject to change.

How Computers Work

Tenth Edition





How Computers Work

The Evolution of Technology

Tenth Edition

Ron White

Illustrated by Timothy Edward Downs

que[®]

800 East 96th Street
Indianapolis, IN 46240

How Computers Work, Tenth Edition

Copyright © 2015 by Ron White

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-4984-0

ISBN-10: 0-7897-4984-X

Library of Congress Control Number: 2014951587

Printed in the United States of America

First Printing: December 2014

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Publishing 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.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

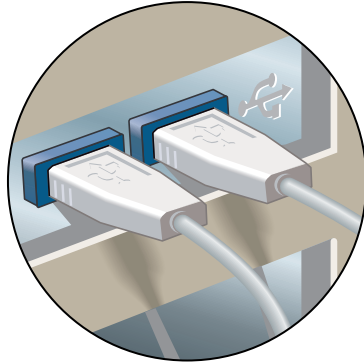
Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact international@pearsoned.com

Editor-in-Chief	Greg Wiegand
Executive Editor	Rick Kughen
Development Editor	Todd Brakke
Managing Editor	Sandra Schroeder
Project Editor	Mandie Frank
Copy Editor	Charlotte Kughen
Indexer	Lisa Stumpf
Proofreader	The Wordsmithery LLC
Editorial Assistant	Kristen Watterson
Designer	Mark Shirar
Compositor	Mary Sudul
Graphics	Timothy Edward Downs



For Shannon and Michael, who always asked, "Why?"
And Sue, who asked "Why Not?"
— Ron

For my mother, Lillian, whose tireless work ethic and
uncompromising dedication to quality has inspired me all
of my life. I owe so much of who I am to you.
— Tim

Introductionxvii

Part 1: What Makes a Computer a Computer 2

Chapter 1
The Ghostly LEGOs of Computing6

Chapter 2
How Computers Remember18

Chapter 3
How a Little Microprocessor Does Big Things28

Chapter 4
How Motherboards Conduct a Symphony of Data42

Part 2: Software—The Computer’s Own Poetry 52

Chapter 5
How Words Are Stitched into Programs64

Chapter 6
How Applications Work So You Can Play74

Chapter 7
How Games Create New Worlds96

Chapter 8
How Security Software Fights Off Invaders106

Part 3: How Computers Evolve 126

Chapter 9
The Origins of Computer DNA130

Chapter 10
How Small Mutations Pay Off Big154

Chapter 11
Honey, I’ve Shrunk the PC168

Chapter 12
Evolution of the Super Computer180

Chapter 13
How Cameras Capture Memories192

Part 4: How Computers Expand Our Senses 200

Chapter 14
How We Stay in Touch204

Chapter 15
How a Computer Creates Visions224

Chapter 16
How Computers Tickle Your Ears238

**Part 5: How the Internet
Connects Us
All 248**

Chapter 17
How Networks Tie Computers
Together254

Chapter 18
How the Internet Brings Us the World...266

Chapter 19
How the Web Puts It All at Your
Fingertips286

Chapter 20
How We Keep in Touch the Internet
Way.....298

**Part 6: How Printers
Put Data in
Our Hands 312**

Chapter 21
How Black-and-White Printing Works .316

Chapter 22
Printing Gutenberg Never Imagined ...324

**Part 7: So, What's
Next? 338**

Index.....348

Table of Contents

Introductionxvii

Part 1: What Makes a Computer a Computer 2

Chapter 1: The Ghostly LEGOs of Computing	6
How Waves Take the Universe on a Joy Ride.....	8
How Information Rides the Waves	10
How Electromagnetism Is the Ghost in the Machine	12
How We Control Electricity	14
How Computers Create Our World Using Numbers.....	16
Chapter 2: How Computers Remember	18
How a Little Transistor Does Big Jobs	20
Writing Data to RAM	22
Reading Data from RAM	24
How Flash Memory Remembers When the Switch is Off	26
Chapter 3: How a Little Microprocessor Does Big Things	28
How a Processor Tracks Numbers.....	32
How a Processor Does Math	34
How a Processor Moves Data	36
How Multi-Core Processors Work.....	38
How Desktop CPUs Keep It Complex	40
How Mobile CPUs Keep It Simple	41
Chapter 4: How Motherboards Conduct a Symphony of Data.....	42
How a Motherboard Brings It All Together	46
How the Chipset Directs Traffic.....	48
How PCI-Express Breaks the Bus Barrier.....	50

Part 2: Software—The Computer’s Own Poetry 52

Chapter 5: How Words Are Stitched into Programs	64
How a Program Is a Roadmap.....	68
How Software Interpreters Generate Action.....	70
How a Compiler Churns Out Software.....	72

Chapter 6: How Applications Work So You Can Play	74
How Text SHOUTS and <i>Whispers</i>	76
How Databases Track Everything	78
How Databases Pack it In	80
How Databases Make Connections	82
How Spreadsheets Solve Formulas	84
How Numbers Become Pictures	86
How Art Is Compressed to Save Space	88
How Imaging Software Paints by Numbers	90
How Photo Editors Revive Old Memories	92
How Apps Fuel Mobile Devices	94
Chapter 7: How Games Create New Worlds	96
How Computers Plot a 3D World	98
How 3D Gets Dressed	100
How Shaders Control the World.....	102
How Games Populate New Worlds	104
Chapter 8: How Security Software Fights Off Invaders	106
How Computer Hackers Break In.....	108
How Spyware Reports Everything You Do	110
How Viruses Invade Your Computer.....	112
How Viruses Hitch Rides in Your Email.....	114
How Antivirus Software Fights Back.....	116
How Firewalls Keep Hackers Out	118
How Spammers Find You.....	120
How Antispam Software Sniffs Out Phony Email.....	122
How Prime Numbers Protect Prime Secrets	124

Part 3: How Computers Evolve 126

Chapter 9: The Origins of Computer DNA	130
Fossils of the Premobilian Era.....	132
Persistent Relics: The Mouse	134
Persistent Relics: The Keyboard	136
How the Workaday Floppy Drive Ruled	138
How the CRT Was the Grande Dame of Displays.....	140

How the Impact Printer Was Right on the Spot.....	142
The Unlikely Progenitor to the Internet: The Dialup Modem	144
How Cabling Was Worse: Serial.....	146
How Cabling Was Worse: Parallel	148
How the Power Supply Hasn't Changed.....	150
How the iPhone Made It Slick.....	152
Chapter 10: How Small Mutations Pay Off Big	154
How USB Really Is Universal	156
How Little Bits Add Up to Big Changes.....	158
How File Compression Makes Files Smaller	160
How Solid State Drives Command the Instant	162
How Solid State Drives Take Out the Trash	163
How PCs Use Light to Remember	164
How Optical Disc Drives Write with Light.....	166
Chapter 11: Honey, I've Shrunk the PC... ..	168
How Computers Get Smaller... and Better.....	170
How iPods Dish Out Media.....	172
How eInk Puts Words on Your eReader	174
How All Those Smarts Got Packed into a Smartphone.....	176
How Google Glass Makes Us Cyborgs	178
Chapter 12: Evolution of the Super Computer	180
How Video Cards Break the Game Barrier.....	182
How Overclocking Multiplies Time.....	184
How a PC Keeps Its Cool	186
How Advanced Cooling Refrigerates Your PC.....	188
How Jailbreaking Frees Your Devices.....	190
Chapter 13: How Cameras Capture Memories	192
How Digital Cameras Capture the Moment	194
How Autofocus Clears the Picture	196
How Cameras Choose an Exposure.....	198

Part 4: How Computers Expand Our Senses 200

Chapter 14: How We Stay in Touch	204
How Your Smartphone Knows Where You Are	206
How Devices Recognize Our Touch	208
How Glass Becomes Strong and Supple.....	209
How Game Controllers Let You Play.....	210
How Game Controllers Let You Feel the Force.....	212
How Smartphones Get Good Vibrations	214
How Devices Capture Light	216
How a Flatbed Simplifies Scanning	218
How Codes Keep Track of Everything (Everything!).....	220
How Optical Character Recognition Works.....	222
Chapter 15: How a Computer Creates Visions	224
How an LCD Creates Glowing Colors	226
How Plasma Displays Glow	228
How DLP Spins Colors	230
How OLED Lights Up a New Generation of Displays.....	232
How 3D Makes You Duck	234
Chapter 16: How Computers Tickle Your Ears	238
How Your Device Listens	240
How Speakers Sound Off	242
How Digital Sound Tricks Your Ear	244
How 3D Audio Surrounds You	246

Part 5: The Little Net That Grew 248

Chapter 17: How Networks Tie Computers Together	254
How Computers Connect to Each Other.....	256
How Data Travels Between Computers	258
How Wi-Fi Makes the Internet Portable.....	260
How Bluetooth Keeps Devices Connected.....	262
How NFC Lets Smart Devices Make Close Calls.....	264
Chapter 18: How the Internet Brings Us the World.....	266
How Broadband Crams in the Data.....	268
How DSL Turbocharges a Phone Line	270
How Cable Brings the Internet to Your Neighborhood	272
How Fiber Optics Lights Up the Future	274

How Computers Make Phone Calls.....276

How Cellular Evolved277

How Networks Talk with Each Other.....278

How Information Travels the Internet.....280

How Online Services Serenade You.....282

How Movies Flow into Your Home.....284

Chapter 19: How the Web Puts It All at Your Fingertips286

 How a Browser Opens Pages.....288

 How a Browser Displays Web Pages290

 How Cookies Trade Crumbs of Data.....292

 How Google Knows Everything.....294

 How eBay Sells Everything.....296

Chapter 20: How We Keep in Touch the Internet Way.....298

 How Email Outpaces Snail Mail.....300

 How Facebook Lets Your Cat Go Viral302

 How Twitter Changes Communication.....304

 How Internet File Sharing Works306

 How BitTorrents Spread the Wealth308

 How Clouds Encompass Us All.....310

Part 6: How Printers Put Data in Our Hands 312

Chapter 21: How Black and White Printing Works316

 How Printers Make Cookie Cutter Text318

 How Outline Fonts Set the Imagination Free320

 How a Printer Writes with Light.....322

Chapter 22: Printing Gutenberg Never Imagined324

 How Printers Create Color326

 How an Ink-jet Sprays Images328

 How a Printer Spits Out Photographs.....330

 How a Laser Printer Creates in Color.....332

 How a Solid Ink Color Printer Works334

 How Printers Create in 3D.....336

Part 7: So, What's Next? 338

Index.....348

About the Author

RON WHITE has been writing *How Computers Work* for 20 years, during which time he's also been executive editor at *PC Computing* magazine, *BYTE.com*, and *groovyPost.com*. He's been a computer columnist for *Windows Sources* and *80 Micro*. *How Computers Work* was named best nonfiction computer book, and his writing has been honored by the Maggie Awards, the Robert F. Kennedy Journalism Awards, and The National Endowment for the Humanities. He and his wife, Sue, have bounced back and forth between Boston and San Francisco before finally ending up in San Antonio.

About the Illustrator

TIMOTHY EDWARD DOWNS is the national award-winning illustrator of *How Computers Work* and *How Digital Photography Works*. Tim has been involved in all facets of graphic design in his illustrious career. From illustrator to creative director, Tim has led teams of artists and designers in advertising agencies, marketing communications firms, and consumer magazines to better tell their stories through illustration, photography, typography, and design. "Our job doesn't start when the writer hits Save. In order to effectively communicate the tone or the concept of the piece, we need to know and understand the story from the original brainstorm all the way through final execution," reminds Tim.

Examples of Tim's design, illustration, and photographic work can be seen at <http://timothyedwarddowns.com>.

Acknowledgments

YOU CAN'T WRITE 10 versions of the same book over a span of 20 years without a lot of help from a lot of people. This book certainly couldn't have been done without Tim Downs. His illustrations have set a new standard for technical art. And *art* is the word. Not only is he always careful to make sure his illustrations are correct technically, he has imbued them with his talent to make them works of art that could stand alone outside of this book's technical explanations. We've worked together for 20 years, and I'm happy that unlike some other partnerships I've entered into, I continue to think of Tim as not just an illustrator, but a dear friend.

This particular edition could not have been done without the aid of some others. My wife, Sue, has served as editor, critic, coach, and personal trainer. I know that the time I've spent in front of a keyboard was not always pleasant for either of us. But when I felt like chucking it all, Sue was there, helping me in the ways I needed most. Our son, Michael White, also spent hours helping with tedious research and checking for errors, despite running his own business and returning to the classroom. Thank you, both.

It's not fashionable among many writers to heap praise on their editors and publishers. But I can't heap enough on my editor, Todd Brakke, and Que Publisher Greg Wiegand. In this and other editions, Todd has been the editor all writers want, ever ready with unerringly good suggestions, encouragement, and lending a hand with the words when time was pressing. Greg has always seemed more like a friend than someone who hired me to write still another edition. When I was ill during the course of writing this, his concern was for my health, not the deadlines I missed. Thank you, guys.

I'm indebted to other editors who gave me the foundations to write. Roddy Stinson taught me the importance of listening to my own voice. Jim Dolan showed me how every detail matters. The late Jim Seymour, who was the editor-in-chief of *PC Computing* when it was launched in 1988, and his wife Nora, who was editor, gave me the opportunity to jump from newspaper to magazines, and were exceptional, warm friends. Preston Gralla guided me in setting up a How It Works section when I joined *PCC*. This book wouldn't be here without him, and I can never thank him enough for a friendship and too many acts of kindness to list.

I owe a huge debt to Mike Edelhart, who was editor-in-chief and publisher of *PC Computing* and who convinced a then dubious book publisher to take a chance on a kind of computer book that had never been done before. Deep thanks go to Randy Ross, Margaret Ficklen, Herb Brody, Brett L. Glass, Marty Jerome, Raymond Jones, Matthew Lake, Jack Nimersheim, Stephen Sagman, Jan Smith, Dylan Tweney, Doug van Kirk, Mark L. Van Name, Bill Catchings, Christine Grech Wendin, and Kenan Woods, all of whom helped in earlier editions.

Thanks to editors now and over the course of nine editions: Rick Kughen, Stephanie McComb, Tonya Simpson, Nick Goetz, Angelina Ward, Sarah Robbins, Cindy Hudson, Melinda Levine, Lysa Lewallen,

Renee Wilmeth, and Leah Kirkpatrick for pulling everything together. Juliet Langley pushed the first edition to book sellers with such fervor I doubt it could ever have been a success without her.

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 Ray Soneira, author of Sonera Technologies' DisplayMate and the person who knows more about display technology than anyone else in the world; Griffin Lemaster of Creative Electron; Chris Thornton of ClipMate; Karen Thomas at Olympus; Joe Vanderwater, Bryan Langerdoff, Dan Francisco, Tami Casey, John Hyde, Bill Kircos, Susan Shaw, and Seth Walker at Intel; Russell Sanchez and Adam Kahn of Microsoft; Jim Bartlett, Ellen Reid Smith, Tim Kearns, and Desire Russell at 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 at Tektronix; Ben Yoder and Lisa Santa Anna at Epson; Dewey Hou at TechSmithCorporation; Kathryn Brandberry and Doyle Nicholas at Thrustmaster; Dani Johnston at Microtek Lab; Lisa Tillman of The Benjamin Group; Dvora Samansky at Iomega; Brandon Talaich at SanDisk; Chris Walker with Pioneer USA; Carrie Royce at the Cirque Corporation; Andy Marken at Marken Communications; and Tracy Laidlaw at BeyondWords. If there are any mistakes in this book, I can't lay the blame on them.

As this *may* be the last edition of the book I work on, I want to thank all the readers for their kind comments, spotting errors, and, of course, shelling out money to buy it. This odyssey would not be possible without you.

We Want to Hear From You!

AS the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: feedback@quepublishing.com

Mail: Que Publishing
ATTN: Reader Feedback
800 East 96th Street
Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at quepublishing.com/register for convenient access to any updates, downloads, or errata that might be available for this book.

About This Book

“Any sufficiently advanced technology is indistinguishable from magic.”

—Arthur C. Clarke

SORCERERS have their magic wands—powerful, potentially dangerous tools with lives 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 wave of a mouse or a spoken incantation 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 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 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.

Introduction to the Tenth Edition

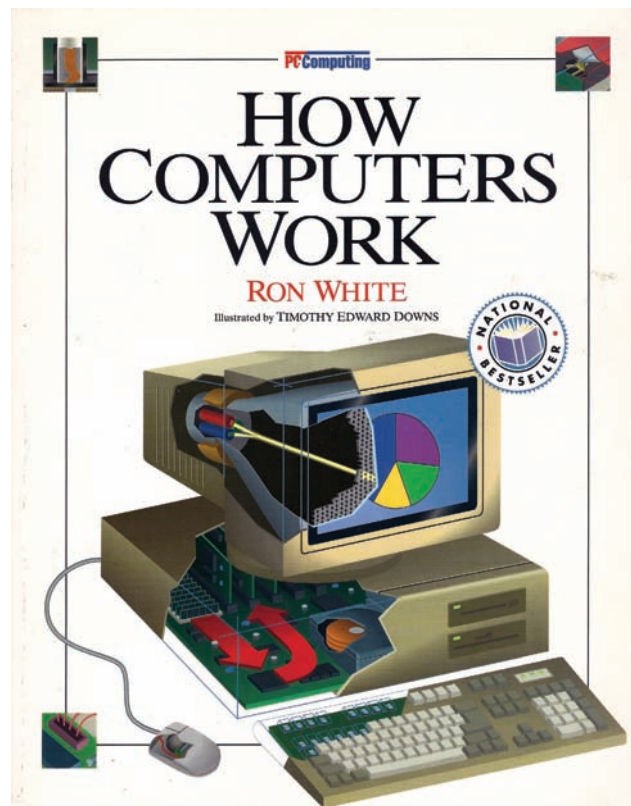
“If automobiles had followed the same development cycle as the computer, a Rolls-Royce would today cost \$100, get a million miles per gallon, and explode once a year, killing everyone inside.”

—Robert Cringely

THIS BOOK was so much easier to write 20 years ago. Computers were relatively new, and they were all pretty much the same. There were only two important differences—whether a computer was an Apple or a Wintel, that is, running Windows on an Intel processor. There were some variations in size, portability, and the customization enthusiasts could do by installing different drives and expansion boards inside PCs. Stuffing the newest-fangled components into my computer at *PC Computing*, the magazine where I worked, was a wonderful pastime and the source of many stories in the magazine. If I made a mistake and the computer groaned its last breath, that often made for an even better story.

One reason PC disasters made for good reading is that every computer user had a similar story of computer disaster. We were all in the same boat because even the editors of computer magazines—if the truth be told—didn’t know what they were doing a lot of the time. Back then if you took a class on computers in college, you waited your turn to run a program on some monster of a computer that would return a solution in the form of punch cards. There was no formal place to go to learn about personal computers. Manuals were useless beyond finding out how to plug a PC into the wall and where the switch was located. Programs that worked one way had no resemblance to how other programs worked. You learned about computers by trial and error, by going to user groups made up of equally befuddled PC owners, and by reading magazines.

Before I went to work for *PC Computing* in 1989, my teachers were *PC Magazine*, *PC World*, *Byte*, *Popular Computing*, and other magazines covering this new technical phenomenon. You found a morsel of information in one magazine, a nugget in another, and eventually you pieced together what was going on inside that steel box on your desktop.



But I was going to tell why it was easier to write this book 20 years ago. Back then, you could say all someone needed to know about a floppy disk in a couple of two-page spreads. Something as mind-busting and intricate as a monitor you could do with one illustration. The technology was, in retrospect, elementary. How computers and their components worked was actually simple when you sketched it all out and added some callouts and arrows. No wonder I figured I could, with the help of a great illustrator like Timothy Edward Downs, write a book that laid out for all to see the mysteries of what really is a fascinating technology.

But today? Lord love a duck! Today there's not even a simple explanation of what a computer is. Simply having a microchip doesn't cut it. Microchips are everywhere, in watches, cars, thermostats, ovens, refrigerators, flashlights, dog collars, coffee pots—just about anything that uses electricity. But these aren't computers. They're just electronic doo-dads that use the abilities of microprocessors to perform routine chores.

To be a computer, something must be programmable. It must be capable of doing different things based on the instructions you give it. Your yard's sprinkler system is a computer. True, it doesn't do a lot of different things, but you can program it to water different parts of your yard for different lengths of time, and on different days, or not at all if it's raining. Microwave ovens are computers you use by programming it on the fly by setting how long it will cook at different temperatures, or you can push a Popcorn button to tell the oven to use a preprogrammed sequence.

These are all computers, but the truth is they're not very interesting. How deeply do you care how your sprinkler works? But a computer that can be programmed to do so... many... different... things—now, that is a computer worthy of the name, and we see them everywhere today. Phones, cameras, tablets, music players, and TVs have all become computers that are indispensable parts of everyday life. Our lives are richer for the knowledge, problem solving, communication, entertainment, health, and income they provide.

They certainly deserve to be included in a book that calls itself *How Computers Work*. The problem—my problem anyway—is that they make organizing a book like this an exercise in controlled chaos. In previous editions, I simply had a section on storage, another section on displays, one on networking.... I admit that in this edition I've fallen back on that system in some chapters out of pure convenience. But generally the workings of today's computers are not so compartmentalized. The more I looked at computing over the last 20 years, the more it struck me that the real story of how computers work is how they are like living creatures, organisms that have evolved and continue to evolve. These metal and silicon and plastic and glass animals have followed the same patterns of mutation, natural selection and survival of the fittest that Darwin found in plants and animals.

I've tried to explain how technological developments follow the same rise and fall that carry organic species to new shores. Today's achievements in computing are possible only because they rise out of the fossil beds of earlier technology. There is no way for today's technology to have been invented from scratch. I recall that some 30 years ago, I was working for a computing service company

that relied on mainframes, an obsolete term to describe computers the size of a couple of high-end refrigerators slammed together. They used two kinds of storage: hard drives and magnetic tape. The 5 megabyte hard drives were the size of large garbage cans, and the platters where information was stored could have doubled as hub caps. You had to have seen the magnetic tape at work to understand one of computer's most common terms: RAM—random access memory. It sounds as if data is retrieved willy-nilly from all different parts of memory chips with no order system to it. Memory chips got their RAM moniker because retrieving tape memory was not random. If you wanted data stored on a reel at the innermost 10 inches of tape, the computer had to wind through the first 3,590 feet to get to that data.

Sure, it would have been better to skip tape altogether and go straight to hard disks. But to paraphrase a certain infamous quote, sometimes you don't build computers with the technology you wish you had; you build them with the technology you have.

Life in the primordial seas had to get the hang of existence as one-celled organisms before the cells could master hooking together and dividing up the jobs needed for mutual survival. Some of those bigger organisms learned to move by wiggling. They developed a way to filter oxygen from water, and later that system developed into lungs that could suck oxygen out of water. It took millions of years for these changes to evolve. You couldn't create magnetic disks until you'd learned how to record data to magnetic tape. Then it took years of engineers trying first this and then that to make the disks smaller and at the same time more capacious. Current solid state drives come from the fossils of technological forays, such as bubble memory, that didn't survive.

The most amazing aspect of this whirlwind of technical progress is that it's only taken 20 years to go from floppy to solid state, from displays that amounted to stick figures to animated graphics that are almost indistinguishable from film images. That smartphone in your pocket has more computing power than those refrigerator-sized mainframes. The wires that have been needed for communications for a century are likely to disappear completely in the next 10 years, along with the need for local data storage. It's possible the keyboard and mouse will become extinct. In fact much of the hardware we're used to today, as it gets smaller and more efficient, may migrate from our desks and our pockets to become parts of our eyes, ears, and brain.

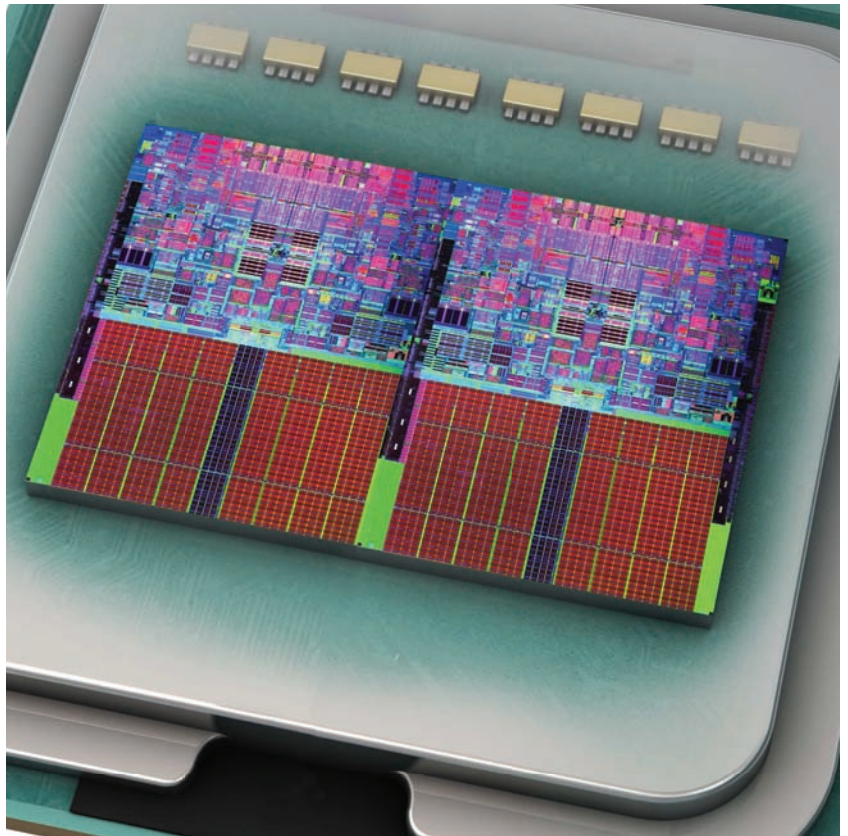
And that's just in the next 20 years. Beyond that we can't imagine what will become of technology any more than one-celled organisms could have imaged a dinosaur. Let's check back with each other after a couple of decades and then see what may be possible.

Ron White
San Antonio, TX

CHAPTER

3

How a Little Microprocessor Does Big Things



IF YOU WANT to feel what it's like to be a microprocessor, go to a toy store for tots and get one of those baby toys that has a bunch of different-sized rings the kid is supposed to put on a stick to make a kind of round pyramid thing. If you don't feel like blowing money on a kid's toy, pull some change out of your pocket. A dime, penny, nickel, quarter, half-dollar (if you can find one), and a dollar bill have a couple of things in common with the toy rings: None is the same size as any other, and they can be stacked. That's all you need for this passage into the brain-numbing world of the microprocessor.

This is called the Towers of Hanoi Puzzle. It's simple but tedious. All you have to do is move the complete stack of objects—rings, coins, plates, whatever—from their initial position, which I'll call "A," to one of two other spots, "B" and "C." There are two rules:

- 1) You can move only one object at a time.
- 2) You cannot place one object on an object smaller than the object you're moving. For example, you can't put a quarter on top of a dime.

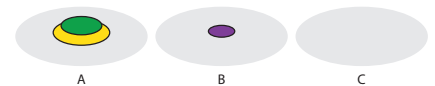
In the original, legendary version of the game, you must move 64 rings while following those rules. It doesn't sound that complicated. And it's not, really. Mathematicians have pointed out the similarities between the Towers of Hanoi and binary computations, geometry, and the exact duplication of a function called the binary carry sequence.

But never mind all that. Let's play the game using only three rings, or disks, as we've drawn them here:

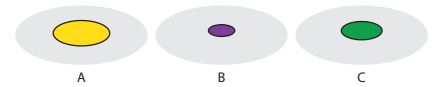
1. Take the top, purple disk and place in the empty B spot.



2. Place the green disk on C.



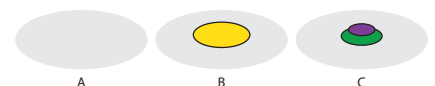
3. Put the purple disk from position B and put it on top of the green disk.



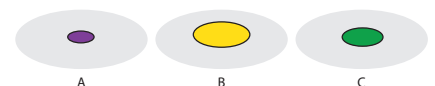
4. Move the yellow disk from A to the B position.



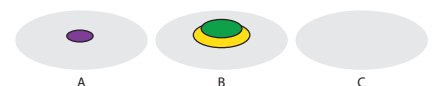
5. Put the purple disk on the now empty position A.



6. Move the green disk on top of the yellow disk at B.



7. Move the purple disk to sit on the green disk.



There. In seven moves you've relocated three disks to a different spot. As you might have guessed, it gets more intricate as you move down through the stack—not more difficult, just more, and more, of the same. Add one more disk, and it takes 15 moves. Ten disks add 1,023 moves.

The Towers of Hanoi Puzzle was discovered by French mathematician Edouard Lucas in 1883. A legend, possibly created by Lucas, has it that there is a temple with a room that contains three poles and 64 gold disks with holes in the centers. The priests at the temple are on a mission from God to move all the discs from one pole to another, Towers-of-Hanoi style. When they complete the task, the world will come to an end. Not to worry. The number of moves to complete the puzzle increases exponentially as the number of discs increases. If the priests could move one disc each second, it would take them 2^{64} [ms]—1 seconds, or roughly 585 billion years, which is 127 times the age of the sun, to complete all the 18,446,744,073,709,551,615 moves required to finish the mission.

This sort of mind-numbing task was made for computing. The **CPU**, or **central processing unit**, whether in a computer, a watch, a smartphone, hearing aid, smart TV, or even a toaster oven plays its own version of the Towers of Hanoi. Like the moves in the puzzle, the methods used by processors work by doing the same thing many times. Multiplying 100 by 3,000 is accomplishing by adding 3,000 a hundred times. Division consists of subtracting one number from another continuously until the result is 0 or the final number is not large enough to subtract the divisor from it.

In the most powerful processors for PCs, the CPU can work with 64 bits at a time, but this doesn't mean that in the monks' full 65 ring game, a 64-bit computer could grab the first 64 disks and move them all at once. It still has to move only one tower ring with each tick of its internal clock, but the computer could simultaneously move one ring on 63 other towers. (A processor such as the Intel Core i7-965 Extreme Edition, with a clock speed of 3.20 gigahertz—3,200,000,000 ticks of the clock every second—could complete the monks' original tower exercise in 13 day and four hours.)

A processor is limited in the number of places it can put the binary numbers representing the bits while still carrying out its software's instructions. After adding two numbers, the CPU might need to put the result in a part of the chip that acts as a temporary holding pen. After adding two more numbers, it retrieves the stored number to add it to the sum of the second addition. It's kind of like how we humans scribble the result of some math operation on a slip of paper, so we can use it a minute later when we solve a related problem.

Much is made of all the amazing things computing can do, but what it can't do, so far, requires intuition, the ability to find a link among seemingly dissimilar ideas. When, in 1997, IBM's Big Blue computer was the first machine to win a tournament with a reigning chess champion—Garry Kasparov—it did so by playing "what if." What if it moved this piece, and then Kasparov moved that piece, and so on through six to eight moves starting from 20 or more contingencies. Big Blue was capable of 200 million positions a second. It's much like doing the Towers of Hanoi—very fast. Human chess champions, on the other hand, say they uniformly avoid the what-if approach in favor of patterns. They will look at the board, and the years of experience buried in their brains see patterns of pieces that reveal a strategy to a win. Or, in Kasparov's case, a loss.

Humans are very good at insight and hunches. Computing is very good at doing the same thing over and over again. Remember that when your computer seems to be taking too long to do something. It's just plodding along in its own manner. It doesn't have your skills.

How a Processor Tracks Numbers

FEW OF US can do complex math in our heads. Even for something as simple as adding several rows of numbers, we need a pencil and paper to keep track of which numbers go where. Microprocessors are not all that different in this regard. Although they are capable of performing intricate math involving thousands of numbers, they, too, need notepads to keep track of their calculations. Their notepads are called **registers**, and their pencils are pulses of electricity.

1 A microprocessor's registers consist of reserved sections of transistors in the faster memory inside the microprocessor. There the processor's **arithmetic logic unit (ALU)**, in charge of carrying out math instructions, and the **control unit**, which herds instructions and data through the processor, have quick access to the registers. The size of the registers determines how much data the processor can work with at one time. Most PCs have registers with 32 or 64 bits for data.



2 The processor's **control unit** directs the fetching and execution of program instructions. (See "How a Microprocessor Moves Data", on pages 36-37.) It uses an electrical signal to fetch each instruction, decodes it, and sends another control signal to the arithmetic logic unit telling the ALU what operation to carry out.

3 With each clock cycle—the thin unit of time during which the different components of a computer can do one thing—the processor writes or reads the values of the bits by sending or withholding a pulse of electricity to each. Each chunk of binary numbers is only that. They have no labels to identify them as instructions, data, values going into a computation, or the product of executing instructions. What the values represent depends on which registers the control unit uses to store them.

4 **Address registers** collect the contents of different addresses in RAM or in the processor's on-board **cache**, where they have been **prefetched** in anticipation that they would be needed.

5 When the processor reads the contents of a location in memory, it tells the data bus to place those values into a **memory data register**. When the processor wants to write values to memory, it places the values in the memory data register, where the bus retrieves them to transfer to RAM.

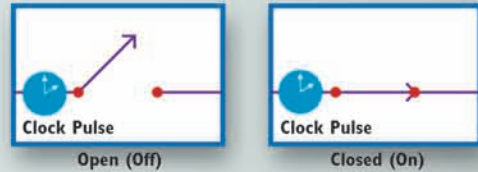
6 A **program counter register** holds the memory address of the next value the processor will fetch. As soon as a value is retrieved, the processor increments the program counter's contents by 1 so it points to the next program location. (A computer launches a program by putting the program's first value into the counter register.)

7 The processor puts the results of executing an operation into several **accumulation registers**, where they await the results of other executing operations, similar to those shown in the illustration on the next spread, "How a Processor Does Math." Some of the instructions call for adding or subtracting the numbers in two accumulators to yield a third value that is stored in still another accumulator.

How a Processor Does Math

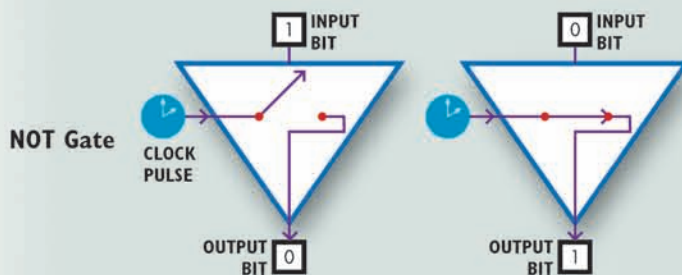
- 1** All information—words and graphics as well as numbers—is stored in and manipulated by a PC in the form of binary numbers. In the binary numerical system, there are only two digits—0 and 1. All numbers, words, and graphics are formed from different combinations of those digits.

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



- 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.

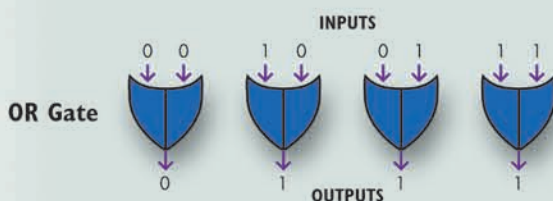
- 3** The simplest operation that can be performed with a transistor is called a **NOT logic gate**, made up of only a single transistor. This NOT gate is designed to take one *input* from the clock and one from another transistor. The NOT gate produces a single *output*—one that's always the opposite of the input from the transistor. When current from another transistor representing a 1 is sent to a NOT gate, the gate's own transistor switches open 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.



NOT Gate Operations

INPUT FROM CLOCK	INPUT FROM OTHER TRANSISTOR	OUTPUT
1	1	0
1	0	1

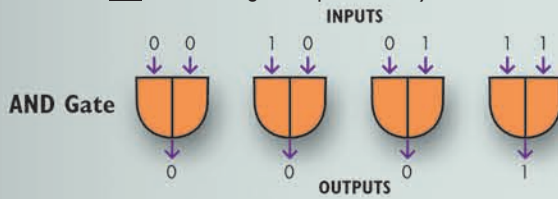
- 4** NOT gates strung together in different combinations create other logic gates, all of which have a line to receive pulses from the clock and two other input lines for pulses from other logic gates. The **OR gates** create a 1 if either the first or second input is a 1, and put out a 0 only if both inputs are 0.



OR Gate Operations

1ST INPUT	2ND INPUT	OUTPUT
0	0	0
1	0	1
0	1	1
1	1	1

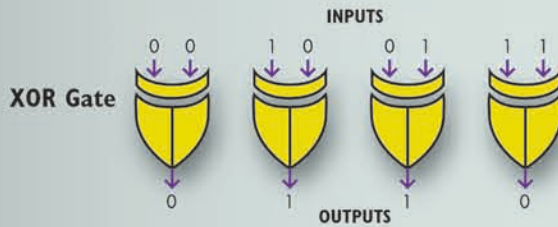
5 An AND gate outputs a 1 only if both the first and the second inputs are 1s.



AND Gate Operations

1ST INPUT	2ND INPUT	OUTPUT
0	0	0
1	0	0
0	1	0
1	1	1

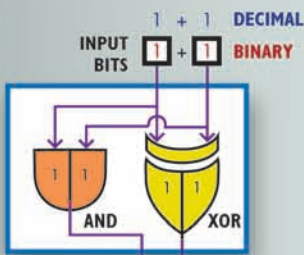
6 An XOR gate puts out a 0 if both the inputs are 0 or if both are 1. It generates a 1 only if one of the inputs is 1 and the other is 0.



XOR Gate Operations

1ST INPUT	2ND INPUT	OUTPUT
0	0	0
1	0	1
0	1	1
1	1	0

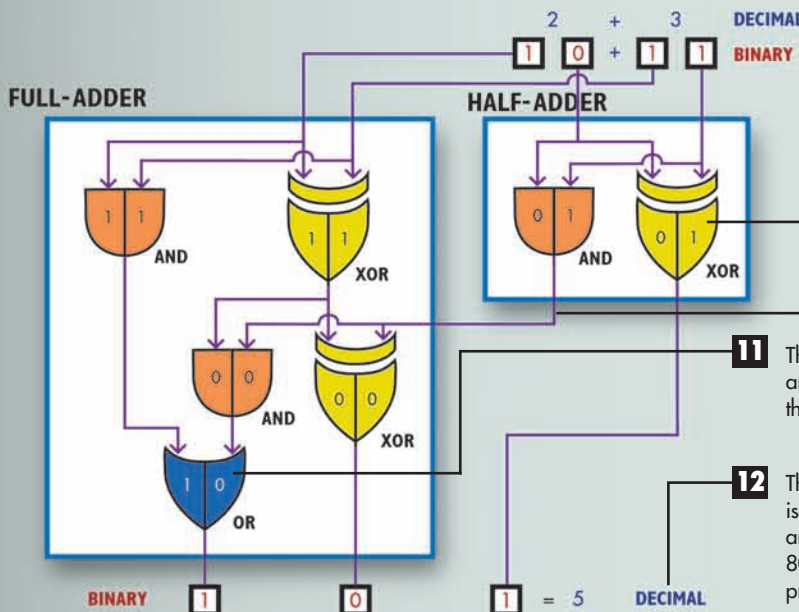
7 With different combinations of logic gates, a computer performs the math that is the foundation of all its operations. This is accomplished with gate designs called **half-adders** and **full-adders**. A half-adder consists of an XOR gate and an AND gate, both of which receive the same input representing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



HALF-ADDER 1 0 BINARY = 2 DECIMAL

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, including today's Pentium class processors, uses 32 full-adders.

How a Processor Moves Data

TODAY'S MICROPROCESSORS have more than 100 billion transistors. Taking a walk through one of them could get a person hopelessly lost. Old or new, however, how a processor performs its most basic functions hasn't changed. They may have as many as eight execution cores and multiple caches—you can look at those on pages **XXX-xxx**—but, like the old single-core Pentium III processor illustrated here, they all face the same problem of how to move data quickly and with nary a hitch.

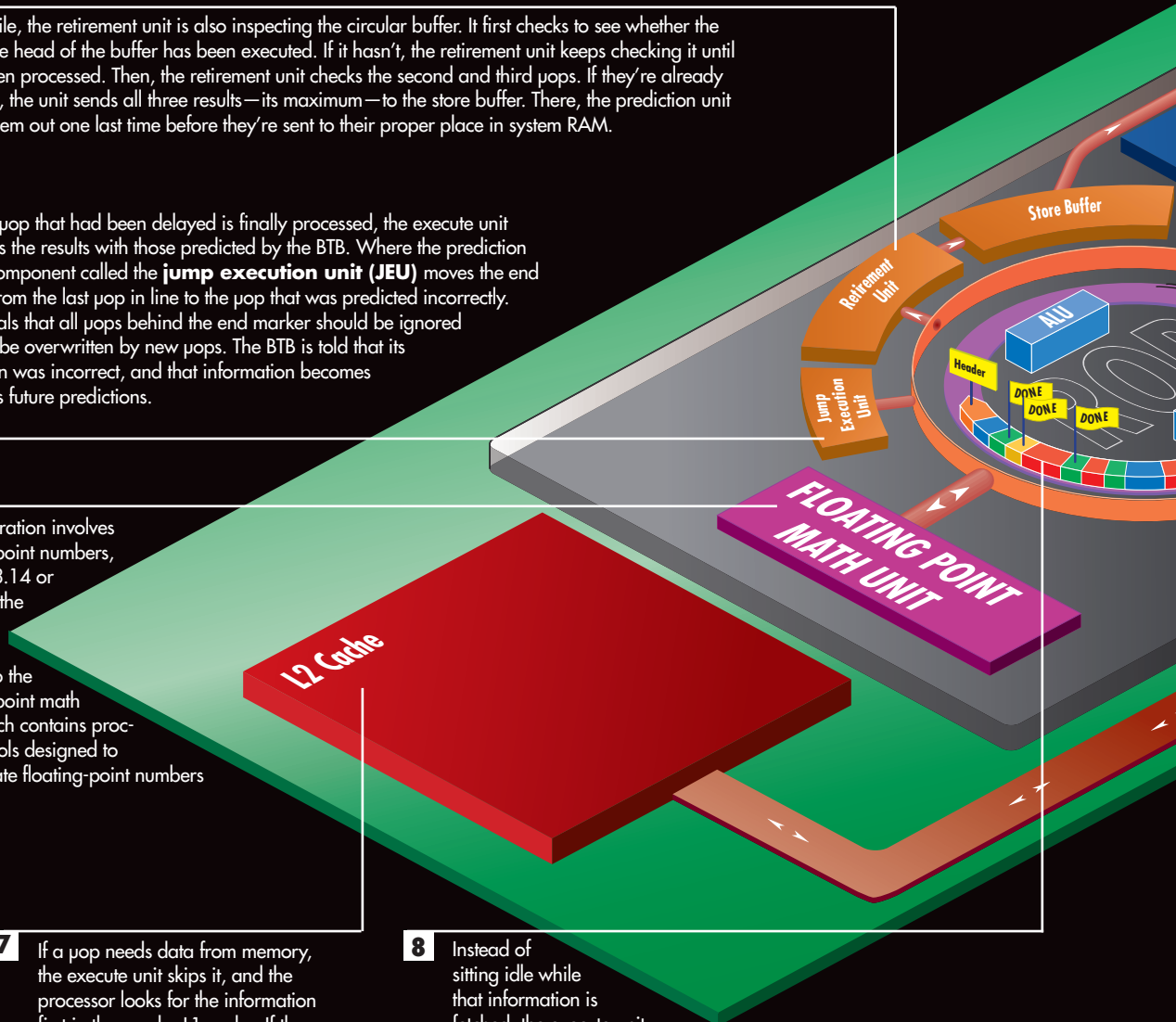
11 Meanwhile, the retirement unit is also inspecting the circular buffer. It first checks to see whether the μ op at the head of the buffer has been executed. If it hasn't, the retirement unit keeps checking it until it has been processed. Then, the retirement unit checks the second and third μ ops. If they're already executed, the unit sends all three results—its 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.

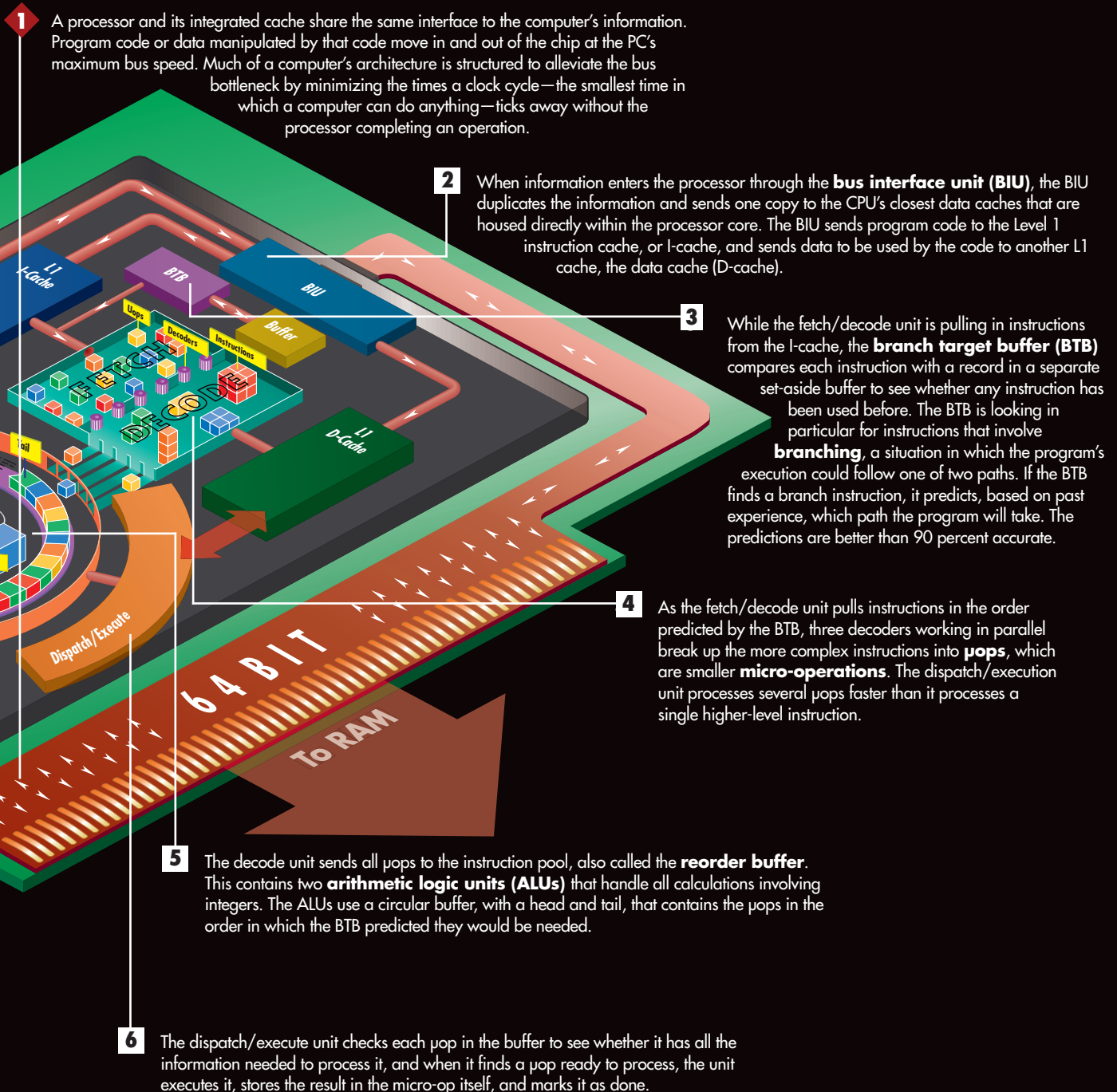
10 When a μ op that had been delayed is finally processed, the execute unit compares the results with those predicted by the BTB. Where the prediction fails, a component called the **jump execution unit (JEU)** moves the end marker from the last μ op in line to the μ op that was predicted incorrectly.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

9 If an operation involves floating-point numbers, such as 3.14 or .33333, the ALUs hand off the job to the floating-point math unit, which contains processing tools designed to manipulate floating-point numbers quickly.

7 If a μ op needs data from memory, the execute unit skips it, and the processor looks for the information first in the nearby L1 cache. If the data isn't there, the processor checks the next cache level, L2 in this case. Cache size and organization vary based on the specific processor design, but each level of cache increases in both capacity and time needed to fetch data from it.

8 Instead of sitting idle while that information is fetched, the execute unit continues inspecting each μ op in the buffer for those it can execute. This is called **speculative execution** because the order of μ ops in the circular buffer is based on the BTB's branch predictions. The unit executes up to five μ ops simultaneously. When the execution unit reaches the end of the buffer, it starts at the head again, rechecking all the μ ops to see whether any have finally received the data they need to be executed.





How Multi-Core Processors Work

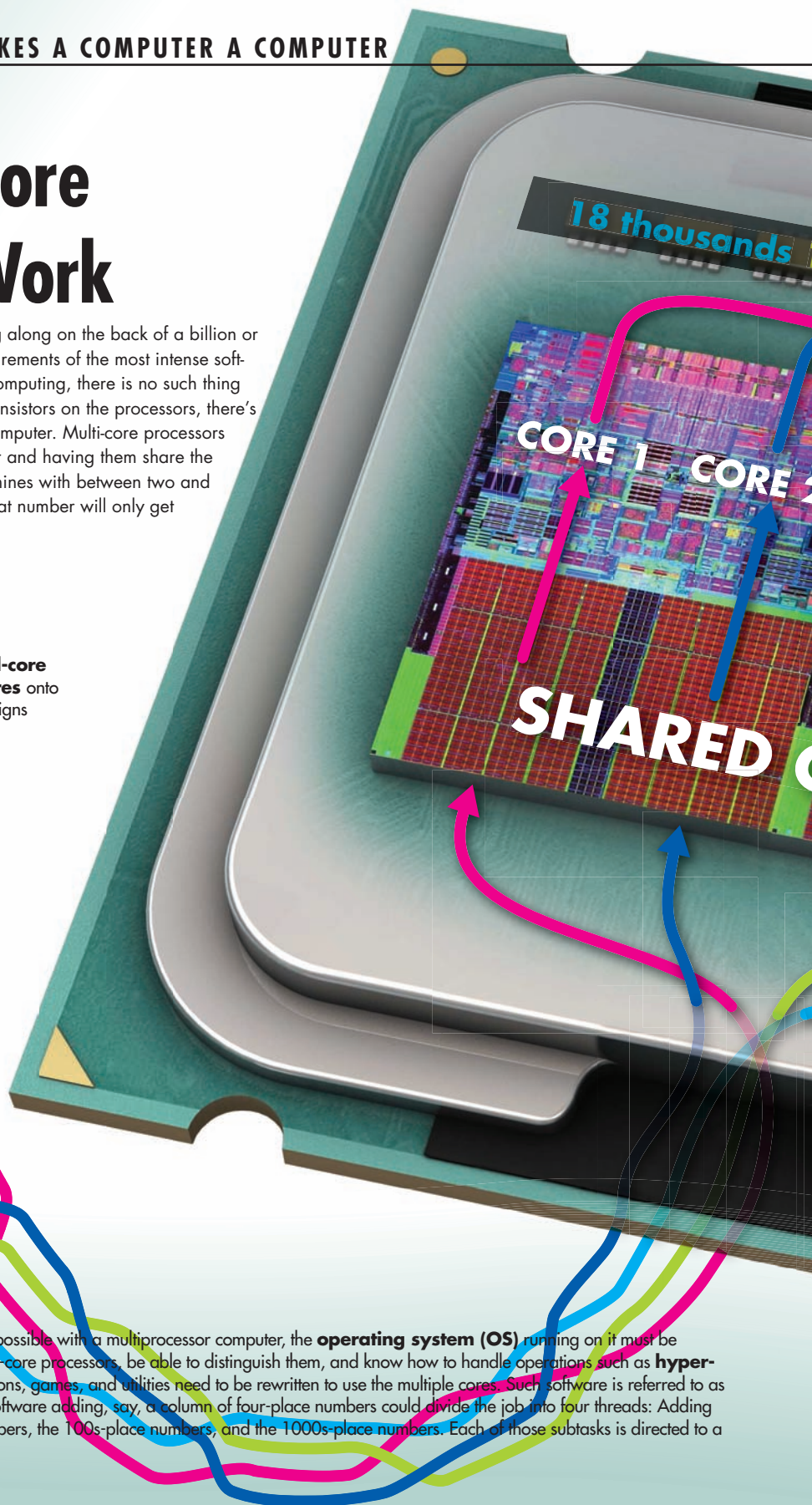
YOU'D THINK the microprocessors humming along on the back of a billion or so transistors would more than satisfy the requirements of the most intense software you can push through the chips. But in computing, there is no such thing as "enough." So if it's too hard to put more transistors on the processors, there's another solution: Put more processors in the computer. Multi-core processors are like bolting a couple of computers together and having them share the same memory, power, and input/output. Machines with between two and ten processor cores are standard issue, and that number will only get bigger.

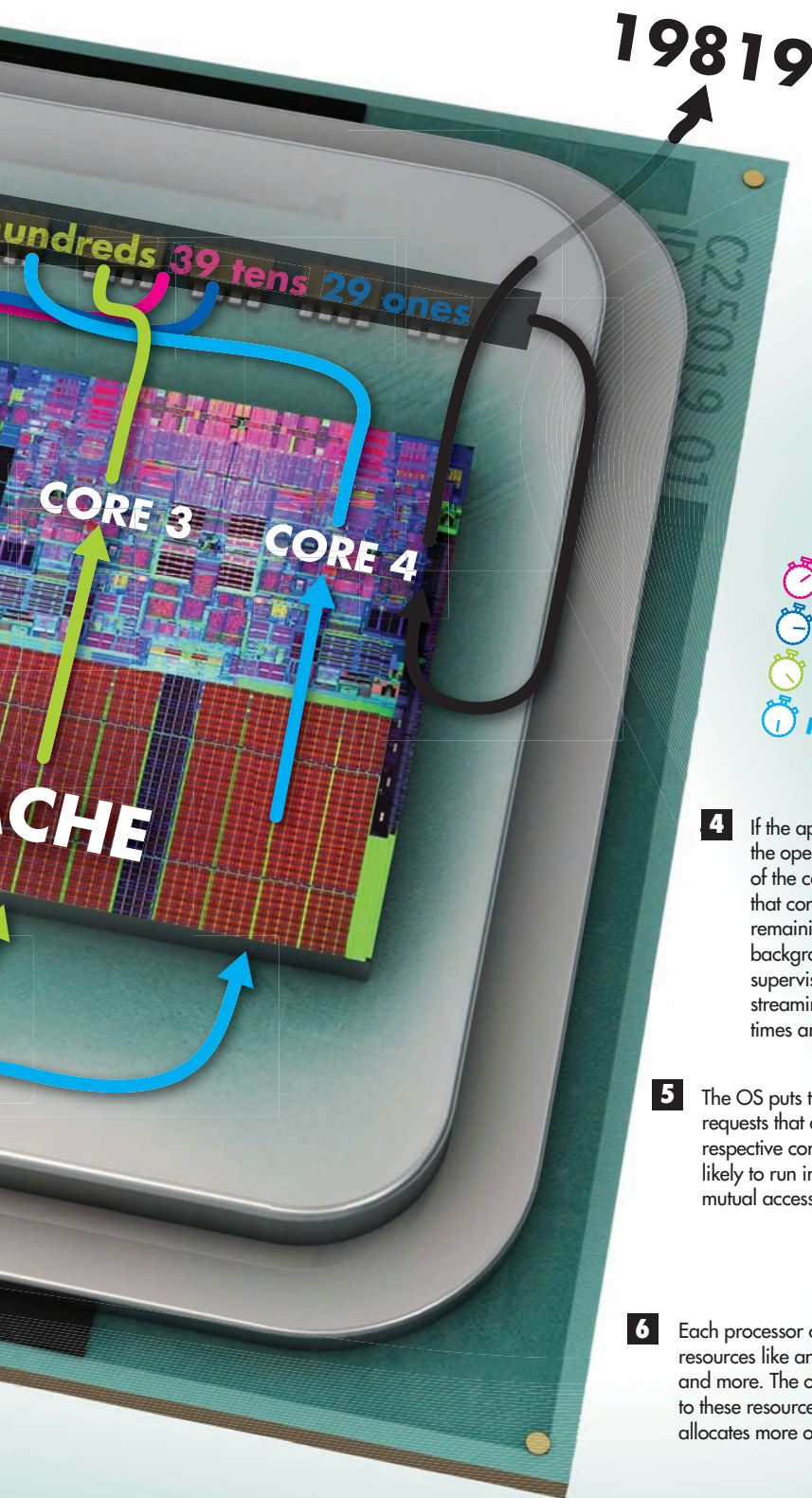
1 Specific designs vary, but a typical **quad-core** processor combines four **execution cores** onto a single **die**, or **silicon chip**. Other designs spread their

cores across two dies. Regardless, these identical cores are the heart of any microprocessor and the part that does the heavy work of executing instructions from software. The wildly colored areas above the cores in the photo are supporting the circuitry.





1480
0379
4077
8294
+5589

2 To gain the speed and other advantages possible with a multiprocessor computer, the **operating system (OS)** running on it must be designed to recognize that a PC has multi-core processors, be able to distinguish them, and know how to handle operations such as **hyper-threading**. Similarly, software applications, games, and utilities need to be rewritten to use the multiple cores. Such software is referred to as **threaded**, or **multi-threaded**. The software adding, say, a column of four-place numbers could divide the job into four threads: Adding the 1s-place numbers, the 10s-place numbers, the 100s-place numbers, and the 1000s-place numbers. Each of those subtasks is directed to a different core.





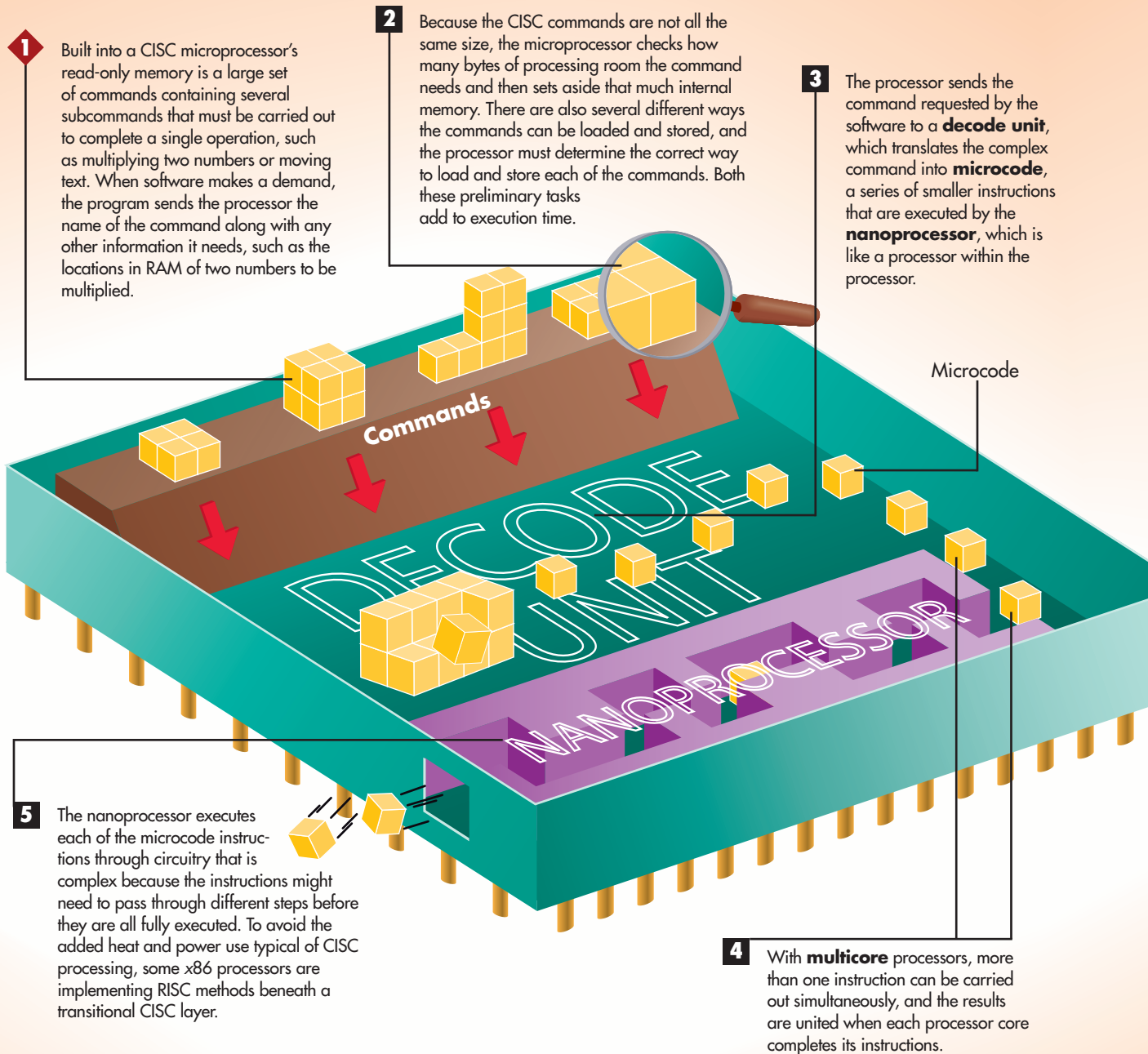
- 3** When the subtasks exit the cores, the operating system combines the threads into a single number, and sends that operation to one of the cores for execution.

-  Word to CORE 1
-  Optimize Disk to CORE 2
-  Download File to CORE 3
-  Render Video to CORE 4

- 4** If the application software isn't equipped to work in multiple cores, the operating system can still take advantage of them. It picks one of the cores to run the software and creates an **affinity** between that core and the program. It then creates affinities between the remaining cores and various tasks. A second core may handle background operations, such as disk optimizing; a third core might supervise a download; and the fourth could render a video that's streaming from the Internet. Neither the operations nor their finish times are affected by the processing going on in the other cores.
- 5** The OS puts that operation into a **time-staggered queue** along with requests that are going to other cores. Each of the operations enters its respective core on different clicks of the computer's clock so they are less likely to run into each other or cause a traffic jam in the areas they have mutual access to.
- 6** Each processor core isn't completely distinct. They might share access to resources like an on-die graphics processor, memory caches (as shown here), and more. The operating system can determine how each core shares access to these resources. If only one core is active, for example, the OS dynamically allocates more of the shared cache to that core.

How Desktop CPUs Keep It Complex

THE CHORES that desktop computers face—putting out a company payroll, rendering blueprints for a new factory—require solving a lot of complex computations. That's why the everyday desktop processors, modeled after Intel's **x86** line, use **complex instruction set computing (CISC)**. It uses complicated, intertwined instructions to bite off manageable problems that it solves and returns to a still-churning solution.



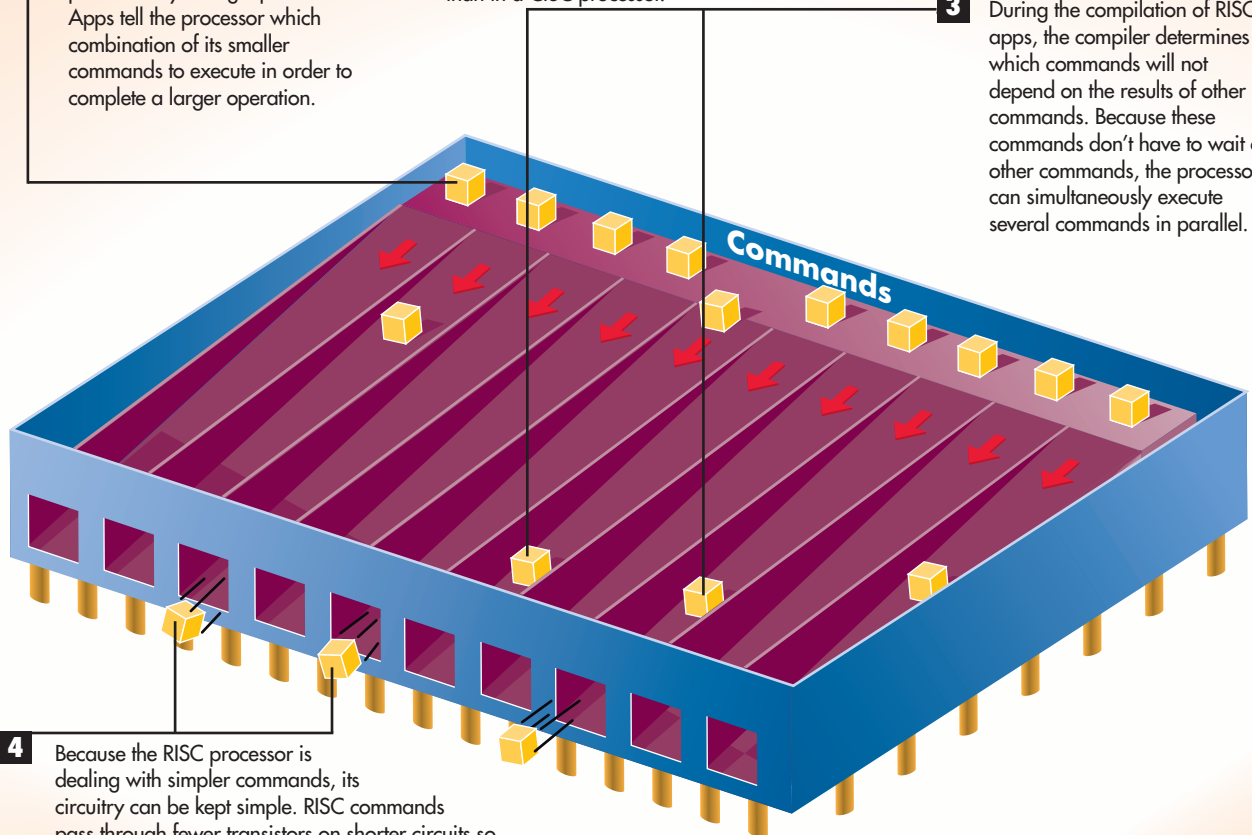
How Mobile CPUs Keep It Simple

THE JOBS that mobile devices take on—dialing a phone number, displaying web pages, playing Angry Birds—are simple as far as computing goes. So they use a simpler system to execute code: **reduced instruction set computing (RISC)**. It deals with instructions that are already broken into chewable pieces, which it quickly gnashes before it spits out the result, using less power and generating less heat in the process. RISC is the basis for the many variants of **ARM architecture** prevalent in 75 percent of mobile products, including everything from tablets to smartphones to digital cameras.

1 Command functions built into an RISC processor consist of several small, discrete instructions that perform only a single job. Apps tell 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 they can be loaded and stored. Because each command is already a form of microcode, RISC processors don't require the extra step of passing instructions through a decode unit to translate them into simpler microcode. This lets the processor load commands for execution faster than in a CISC processor.

3 During the compilation of RISC apps, 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 several commands in parallel.



4 Because the RISC processor is dealing with simpler commands, its circuitry can be kept simple. RISC commands pass through fewer transistors on shorter circuits so the commands execute faster. The result is that often only one CPU **clock cycle** is needed for each instruction. The number of cycles needed to complete a full operation depends on the number of small commands that make up that operation, but generally it will be faster than a CISC processor.

Index

Symbols

0 bit, 22

1 bit, 22

3D, 234

- active shutter glasses, 235
- anaglyph stereoscopy, 234
- autostereoscopic display, 237
- creating with printers, 336-337
- film pattern retarder, 236
- glasses, 235

3D audio, 246-247

3D objects, dressing, 100

3D printers, 314

3D printing, 345

3D worlds, plotting, 98

3G (Generation 3), 277

4G (Generation 4), 277

24-bit graphics, 87

μ -law codec, 145

A

AC (alternating current), PC power supplies, 150

AC3, 244

accelerated graphics port (AGP), 50

accelerometers

- iPhones, 153

- Wiimote, 211

access latency, 162

access points (APs), 260

accumulation registers, 33

action buttons, 210

active content, 114

active matrix technology, 174

active shutter glasses, 3D, 235

adaptive dictionary-based algorithm, 160

ADC (analog-to-digital converters), 16, 217-219

- dial-up modems, 145

additive color (color printing), 326

additive manufacturing, 336

address buses, 46

address lines, 22

address lines (pixels), 229

address registers, 33

addresses, 26

ADSL (asynchronous digital subscriber lines), 271

adventure games, 97

affinities, 39

AGP (accelerated graphics port), 50

ahead of time compilation (AOT), 71

Airbrush, 93

alerts, 119

algorithms, prime numbers, 124

alpha blending, 100

alternating current, 13-14

ALUs (arithmetic logic units), 32, 37

AM (amplitude modulation), 10

American National Standards Institute (ANSI), 75

amperage (amps), 13

amplitude modulation (AM), 10

amps (amperage), 13

anaglyph stereoscopy, 234

analog, 16

analog data communication, 258

analog local loops, dial-up modems, 144-145

analog sticks, 210

analog-to-decimal converter (ADC), 16, 217-219

anchor, 155

AND gates, 35

Android, 63

anodes, 15, 233

ANSI (American National Standards Institute), 75

antispam software, 122-123

antivirus software, 116-117

AOL Time-Warner, 252

AOT (Ahead of Time compilation), 71

AP (Access Points)

- NAP, data transfers, 281

- Wi-Fi networks, 261

Apple, 62

- iPhones, 153

- iPods, 173

application layer (networks), data transfers, 278

application overload, 267

applications, 61

apps, 75

- killer apps, 57

- running mobile devices, 94-95

APs (access points), 260

architecture, ARM architecture, 41

arithmetic logic units (ALUs), 32, 37

ARM architecture, 41

ARM processors, 129

ARPA (Advanced Research Projects Agency), 250

ARPAnet, 250

art, compressing, 88-89

artifacts, 89

assembly language, 66

- atip (Absolute Timing In Pregroove), CD-R drives, 166
 - attachment viruses, 114
 - auctions (online), 296
 - audio
 - 3D audio, 246-247
 - digital audio compression, 245
 - Dolby Digital 5.1, 244
 - Dolby noise reduction (NR), 244
 - JND, 245
 - MP3 files, 245
 - multichannel sound, 244
 - audio ports, 133
 - aura, 208
 - auto exposure (digital cameras), 198-199
 - autofocus lenses (digital cameras), 196-197
 - automatically recalculating spreadsheets, 85
 - autostereoscopic display, 3D, 237
- B**
- back plates, 241
 - backchannel, 111
 - backdoors, 109
 - banding, 89
 - bandwidth
 - boosting, 268-269
 - capacity, 268
 - carriers, 268
 - channel bonding, 51
 - banners, 108
 - Baran, Paul, 250
 - bar codes, 220
 - batteries, 170
 - Google Glass, 179
 - motherboards, 47
 - Bayesian filters, 123
 - bayonet connectors, coaxial cable, 256
 - beeps, 239
 - Bell Labs, 20
 - bias/reset buses, DLP projectors, 230
 - big iron, 56
 - Big Pipe, 303
 - bilinear filtering, 102
 - binary notation, 19
 - binary trees, 81
 - BIOS (basic input/output) system, 133
 - keyboard scan codes, 136-137
 - motherboards, 47
 - bit addresses, 26
 - bit patterned media, 159
 - bit sensors, 26
 - bitmaps, 100
 - bitmap tables, 142
 - cookie-cutter text, 318-319
 - printing, 317
 - bitmap tables, printers, 142
 - bits, 19
 - BitTorrents, 308-309
 - BIU (bus interface unit), 37
 - black-and-white printing, 317
 - outline fonts, 317, 320-321
 - printing, 317
 - vector fonts, 317
 - write-black method, 322
 - write-white method, 322
 - blacklists, 122
 - blocks, 27, 163
 - Bluetooth, 135
 - development of, 262
 - hold mode, 263
 - inquiry mode, 263
 - LM, 263
 - page mode, 263
 - park mode, 263
 - radio modules, 262
 - sniff mode, 263
 - standby mode, 263
 - Bluetooth devices, Wiimote, 211
 - Blu-Ray, 165
 - BNC (Bayonet Neill-Concelman) coaxial cable, 256
 - Boolean logic, 19, 67
 - brackets (session layer), 278
 - branch target buffer (BTB), 37
 - branches, 67
 - branching, 37
 - Bricklen, Dan, 57
 - bridges, data transfers, 281
 - broadband, 269
 - browsers
 - displaying web pages, 290-291
 - domain names, 288
 - ftp, 288
 - http, 288
 - hyperlinks, 288
 - mirror sites, 289
 - page files, 288
 - URL, 288
 - web pages,
 - displaying, 290-291
 - opening, 288-289
 - brute force, how hackers get in, 109
 - BTB (branch target buffer), 37
 - buffers, 114, 283
 - buffers (printers), 142
 - built-in displays, 170
 - Burn, 92
 - bus interface unit (BIU), 37
 - buses
 - address buses, 46
 - data buses, 46
 - expansion buses, 46
 - external buses, 46
 - frontside buses, overclocking, 185

- FSB, 48
 - internal buses, 46
 - motherboards, 46
 - PCI-E, 50-51
 - bushings, 14
 - busy signals, 149
 - bytecode, 71
 - bytes, 24
- C**
- cable
 - coaxial cable, 256
 - fiber-optic cable, 256
 - HFC, 272
 - twisted-pair cable, 256
 - cable modems
 - HFC cable, 272
 - Internet connections, 272-273
 - IP packets, 272
 - cache, 291
 - cache data, boosting
 - bandwidth, 269
 - CAD (computer-aided design), 336
 - calculating spreadsheets,
 - automatically, 85
 - cameras, 170
 - Google Glass, 179
 - cameras (digital), 193
 - auto exposure, 198-199
 - autofocus lenses, 196-197
 - full-frame pictures, 198
 - image sensors, 193
 - CCD, 195
 - CMOS, 195
 - light, capturing, 194-195
 - POS cameras, 198
 - pressure plates, 193
 - shutter locks, 199
 - spot metering, 199
 - white balance, 193
 - camouflage, viruses, 113
 - camps, 105
 - capacitance, 240
 - Capacitive keys (keyboards), 137
 - capacitive touchscreen, 208
 - capacitors, 15, 23, 44, 240
 - PC power supplies, 151
 - Capture button, Google Glass, 179
 - capturing light, 215-216
 - carrier waves, 11
 - analog data communication, 258
 - carriers, bandwidth, 268
 - catch hook, 155
 - cathodes, 15, 233
 - CCD (charged coupled devices)
 - image sensors, digital cameras, 195
 - CCD (coupled-charge device), 217
 - CDMA, 277
 - CD-R (CD-Recordable) drives,
 - 166-167
 - CD-ROM drives, 165
 - CDs, 165
 - dye layer, 167
 - stripes, 167
 - cellphones, 277
 - cellular, 277
 - ceramic-coated porous paper, photo
 - printing, 330
 - chain of trust, 190
 - Chameleon, 169
 - channel bonding, 51
 - charge, 13
 - chat window, Facebook, 303
 - check digit, 220
 - chipsets, 47
 - circuit boards, floppy drives, 139
 - circuit traces, 45
 - circuits, multiplier circuits, 185
 - circular polarizer, 236
 - CISC (complex instruction set
 - computing), 40
 - C.K., Louis, 7
 - cladding fiber-optic cable, 256
 - Clarke, Arthur C., 12
 - classic, 173
 - clepsydra, 155
 - clock cycles, 32, 41
 - clock speeds, 34
 - clocks,
 - overclocking, 184-185
 - real-time, 133
 - Cloning tool, 93
 - clouds, 63, 310-311
 - CMOS chips, 133
 - CMOS (complementary metal oxide
 - semiconductor) image sensors, digital cameras, 195
 - CMOS photodiode array, 217
 - CMTS (cable modem termination
 - system), 284
 - coaxial cable, 256
 - code generators, 73
 - code tokens, 72
 - codec (coder-decoder), 282, 284
 - codes, 220
 - bar codes, 220
 - matrix codes, 221
 - colloidal, 343
 - color printing
 - additive color, 326
 - CYMK color system, 326
 - dithering, 327
 - dye-sublimation method, 331
 - inkjet printers, 328
 - laser printers, 332
 - solid-ink printers, 334
 - subtractive color, 326
 - colors
 - DLP (digital light processing), 230-231
 - true color, 87
 - communications, 170

Compact Flash, 27
 compilers, 65
 how they turnout software, 72-73
 just-in-time compilers, 71
 complex instruction set computing (CISC), 40
 compressing art to save space, 88-89
 compression, digital audio
 compression, 245
 compression (data), bandwidth,
 boosting, 268
 compression (files)
 lossless compression, 160-161
 lossy compression, 161
 CompuServe, 251
 computer-aided design (CAD), 336
 computers
 clocks, overclocking, 184-185
 cooling systems
 forced convection, 186-187
 heat pipes, 187
 heat sinks, 186
 latent heat, 187
 Peltier cooling, 189
 water-cooled PCs, 188
 Google Glass, 179
 making phone calls, 276
 overclocking, 184-185
 power supplies, 150-151
 condenser microphone, 240
 conduction, 186
 conductive, 12
 conductive layers, 233
 cones, 240
 control channel, 304
 control gates, 26, 162
 control signals, 46
 control units, 32
 controlling electricity, 14-15

convection (forced), cooling systems,
 186-187
 cookie-cutter text, printing, 318-319
 cookies, 292-293
 floppy disk data storage, 139
 cooling systems
 forced convection, 186-187
 heat pipes, 187
 heat sinks, 186
 latent heat, 187
 Peltier cooling, 189
 water-cooled PCs, 188
 Corning, 209
 corona, 208
 corona wires (laser printers), 322
 couple-charge device (CCD), 217
 cover image, 303
 CP/M, 129
 CPUs (central processing units),
 30, 43
 desktop CPUs, 40
 mobile CPUs, 41
 CPU sockets (motherboards), 46
 crackers, 107
 CRC (cyclic redundancy checks), 50
 packets, 259
 crest, 240
 crossbar switches, South Bridge, 51
 CRT (cathode ray tube), 15
 CRT (Cathode-Ray Tube)
 screens, 140
 dot pitch, 141
 crystal oscillators, 184
 CSNET (Computer Science
 Network), 250
 cycles, 162
 CYMK (Cyan Yellow Magenta
 Black) color system, color
 printing, 326

D

DAC (digital-to-analog converter),
 17, 140
 dial-up connections, 145
 data, 75
 moving, 36-37
 reading from RAM, 24-25
 writing to RAM, 22-23
 data buses, 46
 data compression, boosting
 bandwidth, 268
 data lines, 22
 data-link layer (networks), data
 transfers, 279
 data storage, floppy drives,
 138-139
 data transfers
 bandwidth, 268
 boosting, 268-269
 carriers, 268
 Internet, 280-281
 Networks, 278-279
 packets, 258-259
 data transmission
 DSL, 271
 modems, 272-273
 POTS, 271
 database manager, 95
 databases
 binary trees, 81
 fixed-length field records, 78
 indexes, 78
 making connections, 82-83
 packing in the data, 80-81
 record structure, 78
 records, indexing, 80
 relational databases, 82
 tables, 82
 dBASE II, 59

- DC (direct current), PC power supplies, 150
 - DDR (double data rate), 25
 - decode units, 40
 - deep learning, 343–344
 - delivery, viruses, 113
 - demodulators, 275
 - depletion area, 241
 - depth cueing, 100
 - desktop CPUs, 40
 - detectors (CD-ROM drives), 165
 - devices, how they listen, 240–241
 - dielectric, 162
 - dial-up Internet connections, 144–145
 - dial-up modems, 144–145
 - diaphragms, 240
 - dictionary attacks, 121
 - die, 38
 - digital, 16
 - digital audio compression, 245
 - digital cameras, 193
 - auto exposure, 198–199
 - autofocus lenses, 196–197
 - full-frame pictures, 198
 - image sensors, 193
 - CCD, 195
 - CMOS, 195
 - light, capturing, 194–195
 - POS cameras, 198
 - pressure plates, 193
 - shutter locks, 199
 - spot metering, 199
 - white balance, 193
 - digital computers, 55
 - digital data, packets, 259
 - digital information, 274
 - digital light processors, 337
 - digital signal processor (DSP), 135
 - digital signatures, 125
 - digital subscriber lines, 270
 - digital-to-analog converter (DAC), 17
 - direct current, 14
 - directional pad, 210
 - directory harvest attacks, 121
 - dirty power, 150
 - discharges, 15
 - displacement, shaders, 103
 - display drivers, 77
 - display lines (pixels), 229
 - distributed Internet structures, 307
 - distributed structure, 307
 - DLP (digital light processing), colors, 230–231
 - DLP (digital light processing) projectors, bias/reset buses, 230
 - DNS (domain name servers), 289
 - doc servers (Google), 295
 - Dodge, 92
 - Dolby Digital 5.1, 244
 - Dolby noise reduction (NR), 244
 - domain name servers, SMTP messages, 300
 - domain names, 288
 - DOS, 61–63
 - dot-matrix printers, 142
 - dot pitch (CRT screens), 141
 - double anti-aliased graphics, 183
 - double data rate (DDR), 25
 - downstream data, DSL, 271
 - Doyen, Chris, 7
 - D-pad, 210
 - drains, 26, 162
 - DRAM (Dynamic Random Access Memory), 162
 - drums (laser printers), 323
 - DSL (digital subscriber lines), 270
 - DSP (digital signal processor), 135
 - 3D audio, 247
 - Dual Shock (Playstation), 210
 - Dungeons and Dragons, 104
 - DVDs, 165
 - dye layer, CDs, 167
 - dye-sublimation method, color laser printers, 331
 - dynamic microphone, 240
- ## E
- eBay, 296
 - eccentric rotating mass actuator (ERM), 215
 - Edison, Thomas A., 55
 - elink, 174
 - Eisenhower, President, 250
 - electret microphones, 241
 - electric dynamo, 14
 - electricity, 4, 12
 - alternating current, 13–14
 - controlling, 14–15
 - direct current, 14
 - ohms, 15
 - static electricity, 12
 - electromagnetic spectrum, 5, 9, 12
 - electromagnetic waves, 9
 - electromagnetism, 12–13
 - electron clouds, 12
 - electron holes, 233
 - electronic paper display (EPD), 174
 - electronic spreadsheets, 58
 - electrophoretic frontplate, 174
 - email
 - POP, 301
 - SMTP, 300
 - viruses, 114–115
 - EMI (electromagnetic interference), PC power supplies, 150
 - emissive layers, 233
 - encoder chips, 210
 - encoding, 10
 - encryption software, 124
 - entanglement, 341

e-paper versus laptops, 174
 EPD (electronic paper display), 174
 eReaders, 174
 how black and white produce shades of gray, 175
 ERM (eccentric rotating mass actuator), 215
 escapement, 155
 escape wheel, 155
 Ethernet networks, 257
 ethernet ports, 170
 evaporative cooling, 187
 event watching, viruses, 112
 evolution, 128
 of Lara Croft, 101
 executable files, 72
 execution cores, 38
 expansion buses, 46
 expansion slots, 133
 experience, MMOs, 105
 exploits, 191
 extension points (Wi-Fi), 261
 external buses, 46
 extinction, 128

F

Facebook, 302–303
 false positives, 123
 fans, 133
 fiber-optic cable, 256
 fiber optics, 274
 fields (screens), 141
 file compression, 160-161
 file servers, LAN, 255
 file sharing, 306
 film pattern retarder, 3D, 236
 filters, polarizing filters (LCD screens), 226
 finder patterns, 221

firewalls, 118–119
 firmware, 190, 320
 five-mega pixel lens, Google Glass, 179
 fixed-length field records, 78
 flash drives, 27
 flash memory, 26-27
 how it remembers when the switch is off, 26
 flat heat pipes, 187
 flatbed scanners, 218
 floating gates, 26, 162
 floppy drives, data storage, 138-139
 flowcharts, 67
 FM (frequency modulation), 10
 focusing coils (CD-ROM drives), 164
 fogging, 100
 fonts
 bitmap tables, 142
 outline fonts, printing, 317, 320-321
 vector fonts, printing, 317
 footers (packets), 259
 footprint analysis, 108
 force feedback, 210
 force-feed joysticks, 212–213
 form factors (motherboards), 46
 formatting
 in-line formatting, 76
 Microsoft Word, 77
 formulas, 58
 how spreadsheets solve them, 84–85
 Fowler-Nordheim tunneling, 27
 frame buffers (video cards), 183
 freeware, 59
 frequency, waves, 9
 frequency modulation (FM), 10
 fresh crawls (Google), 294
 front panel connectors, motherboards, 47

frontside buses, overclocking, 185
 FSB (front side buses), 48
 ftp (file transfer protocol), 288
 full adders, 19, 35
 full duplex messages (session layer), 278
 full-frame pictures (digital cameras), 198
 fundamental resonance, 185
 fusing system (laser printers), 323
 fusion draw process, 209

G

G2 (Generation 2), 277
 game controllers
 force-feed joysticks, 212–213
 game pad, 210
 Wiimote, 211
 game pad, 210
 games
 populating new worlds, 104–105
 video cards, 182-183
 garbage pages, 163
 gas plasma versus LCD, 228
 gates, 34
 AND gates, 35
 NOT gates, 34
 OR gates, 34
 XOR gates, 35
 gateways
 data transfers, 281
 SMTP messages, 300
 generating action, software interpreters, 70–71
 Generation 2 (G2), 277
 Generation 3 (3G), 277
 Generation 4 (4G), 277
 generators, 13
 geometry, turning into things, 99
 geometry engine, 98, 182

gestures, 152
 Gizmo Corp., 56
 glasses, 3D, 235
 G-Lite (DSL), 271
 Global Positioning System (GPS), 206
 Google, 203, 287, 294
 doc servers, 295
 fresh crawls, 294
 PageRank, 295
 Google Glass, 63, 178–179
 Gorilla Glass, 209
 Gouraud shading, 102
 GPS (Global Positioning System), 206
 GPU (graphics processing unit), 99
 granular fusing, 337
 Gradient tools, 93
 graphene, 344–345
 graphic user interface (GUI), 62
 graphics, double anti-aliased graphics, 183
 graphics processing unit (GPU), 99
 gray, how black and white produce shades of, 175
 GSM (Global System for Mobiles), 277
 GUI (graphic user interface), 62
 Gutenberg, Johannes, 325

H

H.264, 284
 hackers, 107
 how they break in, 108–109
 half adders, 19, 35
 half duplex messages (session layer), 278
 haptics, 214
 Hard-Contact keys (keyboards), 137
 hard disk drives, 133

harvesting from email directories, spammers, 121
 hash values, 125
 HDMI ports, 170
 headbands, Google Glass, 178
 headend, 284
 headers, viruses, 112
 headers (packets), 259
 headphone jacks, 170
 healing brushes, 92
 heat pipes, 187
 heat sinks, 151
 heuristic detectors, 116
 hexadecimal, 87
 HFC (hybrid fiber coaxial) cable, 272
 hidden views, 99
 high-definition audio technology, 241
 high-end languages, 66
 high-level languages, 65, 70
 Hive, 303
 hold mode (Bluetooth), 263
 hot spots (Wi-Fi), 261
 HTML viruses, 114
 http (hypertext transfer protocol), 288
 hub site, 284
 hubs
 data transfers, 281
 Ethernet networks, 257
 network connections, 257
 twisted-pair cable, 256
 Huffman threading, 71
 hyperlinks, 288
 hyper-threading, 38

I

I-frame, 284
 IBM PCs, 58–59, 128–129

icons, 62
 identifier tables, 72
 if...then statements, 67
 IM (instant messaging), program compatibility, 329
 image sensors, 16
 image sensors (digital cameras), 193
 CCD, 195
 CMOS, 195
 images
 compressing, 88–89
 photo editors, 92–93
 imaging software, how it uses numbers, 90–91
 impact printers, 142–143
 IMP (Interface message processor), 250
 in-circuit wiring, 27
 index tables, 71
 indexes, databases, 78
 indexing, 294. *See also* Google records, 80
 information, creating our world with numbers, 16–17
 information waves, analog data communication, 258
 ink, pigment inks (photo printing), 330
 ink printers, solid-ink printers (color printing), 334
 inkjet 3D printing, 337
 inkjet printers, color printing, 328
 in-line formatting, text, 76
 input capacitors, PC power supplies, 151
 input signals, waves, 11
 inquiry mode (Bluetooth), 263
 instruction pointers, 71
 insulators, 13
 integer tokens, 72
 interference, 267
 interframe compression, 284

internal buses, 46
 Internet, 251-252
 data transfers, 280-281
 dial-up connections, 144-145
 distributed structures, 307
 evolution of, 287
 file sharing, 306
 ISP, 289
 P2P structures, 307
 traditional structure of, 307
 Internet connections
 Bandwidth, 268-269
 carriers, 268
 DSL, 270
 parallel ports, 148
 serial ports, 146-147
 wireless networks, Bluetooth, 262-263
 interpreted software, 70
 interpreters, 63, 65
 ion implanters, 21
 IP addresses, packet headers, 259-261, 289
 iPhones, 153
 IP (Internet protocol) packets, 272
 iPods, 172-173
 isopipe, 209
 ISP (Internet service providers), 289
 dial-up connections, 145
 item numbers, 220

J

Jacquard, Joseph-Marie, 55
 jaggies (printing), 327
 jailbreaking, 190-191
 Java, 66
 JEU (jump execution unit), 36
 JFET (junction gate-effect transistor), 241

JND (just noticeable difference), sound, 245
 Jobs, Steve, 209
 joysticks, force-feed joysticks, 212
 JPEG 2000, 89
 jump execution unit (JEU), 36
 jumps, 71
 junction gate field-effect transistor (JFET), 241
 just-in-time compilers, 71

K

kernels, 66
 key bytes, 88
 key frames, 284
 keyboard ports, 133
 keyboards, 136-137
 Kilburn, Tom, 57
 Kildall, Gary, 129
 killer apps, 57
 Kindles, 174
 Krupa, Gene, 10

L

LAN (Local Area Networks), 255
 data transfers, 280
 file servers, 255
 nodes, 255
 WLAN, 261
 lands (CD-ROM drives), 164
 languages, 65
 assembly language, 66
 compilers. *See* compilers
 high-end languages, 66
 high-level language, 70
 Java, 66
 low-level machine language, 70
 programming languages, 65
 scripting languages, 70
 software construction, 66-67
 software interpreters, 70-71
 laptops, 170-171
 versus e-paper, 174
 Lara Croft, evolution of, 101
 laser printers, 322-323
 latency, 267
 latency, reducing (bandwidth), 269
 latent heat, cooling systems, 187
 layers, restoring photos, 93
 LCD (Liquid Crystal Display) screens, 227, 230
 plasma screens versus, 228
 polarizing filters, 226
 versus gas plasma, 228
 versus LEDs, 226
 LEDs versus LCDs, 226
 leeches (torrenting), 309
 levels, histograms, 92
 lexer, 72
 lexical analysis, 72
 LFE (low frequency effect) channel, multichannel sound, 244
 light, capturing, 215-216
 light sources, 98
 line noise (phone), dial-up modems, 145
 linear resonate actuator (LRA), 215
 LM (link managers), Bluetooth, 263
 location, smartphones, 206-207
 logic gates, 19
 longitudinal waves, 8
 lossless compression, 160-161
 lossy compression, 161
 Lotus 1-2-3, 58-60
 low-level languages, 65
 low-level machine language, 70
 LRA (linear resonate actuator), 215
 LTE (Long Term Evolution), 277
 Luddite, 55

LZ adaptive dictionary-based algorithm, lossless compression, 160

M

machine languages, 65
 magnetic deflection yokes, 140
 magnetic fields, 13
 magnetic induction, 265
 magnetism, 12
 stator magnets, 14
 mainframes, 56
 malware, 110, 267
 manufacturer identification number, 220
 map-reduction, 303
 markers, text mapping, 76
 matrix codes, 221
 mechanical waves, 9
 megahertz, 34
 memory, flash memory, 26-27
 memory data registers, 33
 memory-resident antivirus programs, 117
 memory-resident viruses, 113
 Memory Sticks, 27
 MEMS (Micro Electro Mechanical Systems) microphone, 241
 message digest, 125
 metafiles, 282
 metal oxide, 26
 microcapsules, 174
 microchips, 20, 45
 moving data, 25
 microcode, 40
 micro-operations, 37
 microphones, 170
 condenser microphones, 240
 dynamic microphone, 240
 electret microphones, 241

MEMS (Micro Electro Mechanical Systems) microphone, 241
 speakers, 243
 micropolarizer, 236
 microprocessors, 32, 133
 building, 21
 Microsoft
 DOS, 61-63
 Windows, 63
 Microsoft Word, formatting, 77
 microtransactions, 60
 microwaves, 11
 MIME (Multi-Purpose Internet Mail Extension) viruses, 114
 MIMO (multiple in, multiple out), 277
 MIP mapping, 100
 mirror sites, 289
 MMORPG (massively multiplayer online role playing game).
 See MMOs
 MMOs (massively multiplayer online), 104-105
 mobile CPUs, 41
 mobile devices, running with apps, 94-95
 mobile switching center, 305
 modems, 133
 cable modems, 272-273
 dial-up Internet connections, 144-145
 dial-up modems, 144-145
 modifying records, 78
 modulation, 11
 modulator waves, 268
 molten polymer deposition, 337
 monitors
 CRT, 140
 dot pitch, 141
 DLP projectors, 230
 fields, 141
 LCD, 227

 plasma screens versus, 228
 polarizing filters, 226
 LCD projectors, 230
 phosphors, 141
 plasma, 228-229
 raster scanning, 141
 refreshing, 141
 refresh rates, 140
 resolution, 140
 shadow masks, 141
 Morse code, 10
 motherboards, 43
 batteries, 47
 BIOS, 47
 buses, 46
 chipsets, 47
 CPU sockets, 46
 form factors, 46
 front panel connectors, 47
 North Bridge, 48
 Port 80 displays, 47
 ports, 46
 SATA connectors, 47
 South Bridge, 49
 crossbar switches, 51
 Motorola, 277
 mouse, 134-135
 mouse ports, 133
 movies, streaming, 284-285
 moving data, 36-37
 MP3 files, 245
 multi-bit sampling, 17
 multichannel sound, 244
 multicore processors, 38-40
 multimode fibers, 275
 multiplexes, boosting bandwidth, 269
 multiplier circuits, 185
 multistrand fibers, 275
 multitasking, iPhones, 152

multi-tank cartridge, 337
 multi-threaded, 38
 multi-touch capability, 208
 music, online services, 282
 mutual capacitance screens,
 iPhones, 153

N

NAND flash memory, 162
 Nano (iPod), 173
 nanoprocessors, 40
 NAP (Network Access Points), data
 transfers, 281
 neodymium, 240
 Net, 252
 Netscape, 252
 network interface, 270
 network layer (networks), data
 transfers, 279
 network ports, 133
 networks
 coaxial cable connections, 256
 data transfers
 application layer, 278
 data-link layer, 279
 LAN, 280
 network layer, 279
 physical layer, 279
 presentation layer, 278
 session layer, 278
 transport layer, 278
 WAN, 280
 Ethernet networks, 257
 fiber-optic cable connections, 256
 hubs, 257
 LAN, 255
 node connections, 256-257
 P2P, 255
 packets, 258-259

routers, 257
 switches, 257
 twisted-pair cable
 connections, 256
 Wi-Fi, 260-261
 wireless networks, 256
 Bluetooth, 262-263
 WLAN, 261
 NFC (near field communications),
 264-265
 n-layer, 216
 nodes, 73
 LAN, 255
 nonconductive materials, 13
 non-player characters (NPCs), 105
 North Bridge (motherboards), 48
 Norton, Peter, 59
 NOT gates, 34
 NOT logic gate, 34
 NPCs (non-player characters), 105
 NSF, 251
 n-type silicon, 216
 nucleus, 12
 numbers
 analog, 16
 creating our world with, 16-17
 digital, 16
 how imaging software uses
 numbers, 90-91
 how they become pictures, 86-87

O

OCR (optical character recognition),
 scanners, 222-223
 OFDMA (orthogonal frequency
 division multiple access), 277
 offsets, 70, 78
 ohms, 15
 oil, cooling PCs, 189

OLED (ORGANIC light-emitting
 diode display), 232-233
 online auctions, 296
 online services, music, 282
 OPC (Organic Photo conducting
 Cartridges), laser printers, 323
 operating systems (OS), 38, 56, 61
 evolution of, 62-63
 optical disc drives, 166-167
 optical drives, 132
 optical mouse, 134
 optimizers, 73
 ORGANIC light-emitting diode
 display (OLED), 232
 OR gates, 34
 OS (operating system), 38
 outline fonts, printing, 317,
 320-321
 output capacitors, PC power
 supplies, 151
 overclocking, 184-185
 overflows, 114
 overtone frequencies, 185

P

P2P (peer-to-peer), torrenting, 308-
 309
 P2P (peer-to-peer) Internet structures,
 307
 P2P (peer-to-peer) networks, 255
 packet filtering, 118
 packets, 118, 258-259
 Ethernet networks, 257
 page files, 288
 page mode (Bluetooth), 263
 pagelets, 303
 PageRank (Google), 295
 paper, photo printing, 330
 paper movers, scanners, 219
 paper trains (laser printers), 323

- parallel communications, 148-149
- parallel ports, 148
- park mode (Bluetooth), 263
- parsers, 73
- parsing, 67
- particle shaders, 103
- pattern recognition, iPhones, 152
- Patterson, Tim, 129
- payloads (packets), 259
- PC (player character), 104
- PCI (peripheral components interconnect), 50
- PCI-E (peripheral components interconnect-express) buses, 50-51
- PCM (pulse code modulation), dial-up modems, 145
- PCs
 - cooling systems
 - forced convection, 186-187
 - heat pipes, 187
 - heat sinks, 186
 - latent heat, 187
 - Peltier cooling, 189
 - water-cooled PCs, 188
 - power supplies, 150-151
- Peltier cooling, PC cooling systems, 189
- peripheral components interconnect (PCI), 50
- peripherals
 - keyboards, 136-137
 - monitors
 - CRT, 140-141
 - DLP projectors, 230
 - fields, 141
 - LCD, 226-228
 - LCD projectors, 230
 - phosphors, 141
 - plasma, 228-229
 - raster scanning, 141
 - refreshing, 141
 - refresh rates, 140
 - resolution, 140
 - shadow masks, 141
 - paper movers, scanners, 219
 - scanners
 - ADC, 219
 - flatbed scanners, 218
 - OCR, 222-223
 - paper movers, 219
 - perpendicular magnetic recording (PMR), 158
 - persistence (phosphors), 141
 - perspective correction, 100
 - phones
 - how computers make phone calls, 276
 - line noise, dial-up modems, 145
 - phosphors (monitors), 141
 - photo editors, 92-93
 - photo printing, 330
 - photo sites, 216
 - photocells, 216
 - photodiodes, 16
 - photoetching, 21
 - photography, 193
 - digital cameras
 - auto exposure, 198-199
 - autofocus lenses, 196-197
 - capturing light, 194-195
 - CCD image sensors, 195
 - CMOS image sensors, 195
 - full-frame pictures, 198
 - POS cameras, 198
 - shutter locks, 199
 - spot metering, 199
 - photons, 217, 232
 - photons (plasma screens), 229
 - photoresist, 21
 - photos, restoring, 92-93
 - photovoltaic cells, 216
 - physical layer (networks), data transfers, 279
 - physics engine, 182
 - pictures, how they are created from numbers, 86-87
 - piezo controllers (solid-ink printers), 334
 - piezo-electric nozzles (photo printers), 330
 - pigment inks, photo printing, 330
 - pits (CD-ROM drives), 164
 - pixelation, 100
 - pixels
 - address lines, 229
 - display lines, 229
 - reference pixels, 89
 - virtual pixels, 87
 - PKs (player killers), 105
 - plasma displays, 228-229
 - plasma screens, 228-229
 - plastic-coated swellable paper, photo printing, 330
 - p-layer, 216
 - player character (PC), 104
 - Playstation, Dual Shock, 210
 - plotting 3D worlds, 98
 - PMR (perpendicular magnetic recording), 158
 - point-to-point connections (PCI-E buses), 51
 - polarizing filters, 227
 - LCD screens, 226
 - polygons, 98
 - POP (Post Office Protocol), 301
 - populating new worlds, games, 104-105
 - Port 80 displays, motherboards, 47
 - ports, 118
 - motherboards, 46
 - parallel, 148
 - serial ports, 146-147

- POS (point-and-shoot) digital cameras, 198
 - potentiometers, 210
 - POTS (plain old telephone systems), data transmission, 271
 - power adapters, 170
 - power supplies, 150-151
 - batteries, motherboards, 47
 - power supply, 14, 132
 - prefetch data, boosting bandwidth, 269
 - presentation layer (networks), data transfers, 278
 - pressure plates (digital cameras), 193
 - primary coil, 14
 - primary keys, 82
 - prime numbers, 124-125
 - print cartridges, 329
 - printers
 - 3D printers, 314, 336-337
 - bitmap tables, 142
 - buffers, 142
 - dot-matrix printers, 142
 - impact printers, 142-143
 - inkjet printers, color printing, 328
 - laser printers
 - color printing, 332
 - corona wires, 322
 - drums, 323
 - fusing system, 323
 - OPC, 323
 - paper trains, 323
 - toner, 323
 - printing pins, 143
 - solid-ink printers, color printing, 334
 - printing, 314
 - black-and-white printing, 317
 - bitmaps, 317
 - outline fonts, 317, 320-321
 - vector fonts, 317
 - write-black method, 322
 - write-white method, 322
 - color printing
 - additive color, 326
 - CYMK color system, 326
 - dithering, 327
 - dye-sublimation method, 331
 - inkjet printers, 328
 - laser printers, 332
 - solid-ink printers, 334
 - subtractive color, 326
 - cookie-cutter text, 318-319
 - jaggies, 327
 - photo printing, 330
 - printing pins (printers), 143
 - printing press, 325
 - privacy policies, 110
 - private keys, 124
 - probability, 341
 - probe requests (Wi-Fi), 260
 - processors, 19
 - how they do math, 34-35
 - how they track numbers, 32-33
 - moving data, 36-37
 - multi-core processors, 38-39
 - multicore processors, 40
 - nanoprocessors, 40
 - quad-core processors, 38
 - program counter registers, 33
 - programming languages, 65-66
 - programs as roadmaps, 68-69
 - projectors, Google Glass, 178
 - PSTN (public switched telephone networks), dial-up modems, 145
 - PSU (power supply unit), 150
 - p-type, 216
 - public keys, 124
 - public sources, spammers, 120
 - pulse modulation, 11
 - punched cards, 55
- Q**
- QR (quick response) codes, 221
 - quad-core processors, 38
 - quantum, 340
 - quantum computers, 341
 - quantum dots, 342-343
 - qubits, 341
- R**
- radiators, 188
 - radio modules (Bluetooth), 262
 - radio waves, 11
 - RADSL (rate adaptive digital subscriber lines), 271
 - RAM, 133
 - reading data, 24-25
 - writing data to, 22-23
 - RAM (random access memory), 131
 - RAND Corporation, 250
 - raster scanning (monitors), 141
 - rasterization, 182
 - ray tracing, 102
 - reading data from RAM, 24-25
 - read/write heads (floppy drives), 139
 - real-time clocks, 133
 - record structure, databases, 78
 - records
 - indexing, 80
 - modifying, 78
 - reduced instruction set computing (RISC), 41
 - reference pixels, 89
 - refraction, 267
 - refresh rates (monitors), 140
 - refreshing screens (monitors), 141
 - registers, 32-33, 66, 71
 - relational databases, 82

- rendering, 182
 - rendering engines, 99
 - repeaters, data transfers, 281
 - replication, viruses, 112
 - reserved words, 72
 - reservoirs, 188
 - resistance, 13
 - resistors, 15, 44
 - resolution, 17
 - resolution (monitors), 140
 - resonance, 184
 - fundamental resonance, 185
 - restoring old photos, 92–93
 - rheostat, 15
 - ribs (plasma screens), 228
 - RISC (reduced instruction set computing), 41
 - RJ-45 connectors, twisted-pair cable, 256
 - RLE (run-length encoding), 88
 - roadmaps, 68–69
 - ROM manager, 191
 - roofing, 191
 - rotors, 14
 - routers
 - data transfers, 280
 - Ethernet networks, 257
 - network connections, 257
 - twisted-pair cable, 256
 - rules, firewalls, 118
 - run-length encoding (RLE), 88
- S**
- sampling rate, 17
 - sapphire, 209
 - SATA connectors, motherboards, 47
 - satellites, 206–207
 - scalability, PCI-E buses, 51
 - scan codes, keyboards, 136–137
 - scanners
 - ADC, 219
 - flatbed scanners, 218
 - OCR, 222–223
 - paper movers, 219
 - scanning with flatbeds, 218–219
 - scissors rendering, gaming, 182
 - screens
 - CRT, 140
 - dot pitch, 141
 - fields, 141
 - LCD, 227
 - plasma screens versus, 228
 - polarizing filters, 226
 - plasma, 228–229
 - refreshing, 141
 - script kiddies, 107
 - scripting languages, 70
 - scroll wheel, 135
 - SD slots, 170
 - secondary coil, 14
 - security
 - antispam software, 122–123
 - antivirus software, 116–117
 - digital signatures, 125
 - firewalls, 118–119
 - prime numbers, 124–125
 - security logs, 119
 - seeds (torrenting), 308
 - Selection tool, 92
 - self-capacitance screens, iPhones, 153
 - semiconductors, 13
 - sensor bars, 211
 - sensors, bit sensors, 26
 - serial communications, 146–147
 - serial ports, 146–147
 - session layer (networks), 278
 - shaders, 102–103
 - shaders (video cards), 183
 - shading, 102
 - shadow masks (monitors), 141
 - shareware, 59
 - sharing files, 306
 - Sharpen tool, 93
 - shingled memory recording, 159
 - Shuffle (iPod), 173
 - shutter locks (digital cameras), 199
 - signal transfer point (STP), 305
 - signatures, 86
 - digital signatures, 125
 - signature scanners, 116
 - silicon, 13, 216
 - silicon chips, 38
 - simple machines, 4
 - single mode fibers, 275
 - single strand fibers, 275
 - size of computers, 170
 - Smart Media, 27
 - smartphones, 63, 176–177
 - location, 206–207
 - vibrations, 214–215
 - SMR (shingled memory recording), 159
 - SMSC (small message system center), 305
 - SMS (small message system), 305
 - SMTP (Simple Mail Transfer Protocol), 300
 - sniff mode (Bluetooth), 263
 - social media
 - Facebook, 302–303
 - Twitter, 304–305
 - software, 54–57
 - freeware, 59
 - shareware, 59
 - software construction, 66–67
 - software interpreters, languages, 70–71
 - solid-ink printers, color printing, 334
 - solid state disk drives, 170

- solid state drive (SSD), 162
 - sound
 - 3D audio, 246-247
 - beeps, 239
 - digital audio compression, 245
 - Dolby Digital 5.1, 244
 - Dolby noise reduction (NR), 244
 - JND, 245
 - MP3 files, 245
 - multichannel sound, 244
 - sound cards, 133
 - sources, 26
 - South Bridge (motherboards), 49
 - crossbar switches, 51
 - spam, speed of spreading, 121
 - spammers
 - harvesting from email directories, 121
 - public sources, 120
 - spammerwocky, 123
 - spawning points, 105
 - speakers, 170, 242-243
 - Google Glass, 179
 - speculative execution, 36
 - spiders, 120
 - spot metering (digital photography), 199
 - spreadsheets
 - automatically recalculating, 85
 - how they solve formulas, 84-85
 - spread spectrum, 277
 - Sputnik, 250
 - sputtering, 159
 - Spybot Search and Destroy, 111
 - spyware, 110-111
 - SSD (solid state drive), 162-163
 - stack memory, 114
 - standby mode (Bluetooth), 263
 - standing objects, 95
 - stateful inspections, 119
 - static electricity, 12
 - stations (Wi-Fi), 260-261
 - stator magnets, 14
 - stealth viruses, 113
 - step-down transformers, 14
 - stepper motors (floppy drives), 138
 - stereolithography, 337
 - stippling, 100
 - STL formatted files, 336
 - storing data, floppy drives, 138-139
 - STP (signal transfer point), 305
 - streaming movies, 284-285
 - streaming nodes, 284
 - stripes (CDs), 167
 - strobe signals, 148
 - styles, text, 76
 - sublimate, 331
 - subroutine calls, 71
 - substrate, 162
 - subtractive color (color printing), 326
 - subtractive processes, 336
 - superparamagnetic limit, 159
 - supertiling, 182
 - survival of the fittest, 128
 - SVGA (Super Video Graphics Array) adapters, 140
 - switches, 62
 - crossbar switches, South Bridge, 51
 - Ethernet networks, 257
 - network connections, 257
 - twisted-pair cable, 256
 - synapsids, 128
 - syntactic analysis, 73
 - syntax trees, 73
- T**
- tables, 82
 - tandem color laser printers, 333
 - tape drives, 132
 - targets, 159
 - TCP/IP (Transmission Control Protocol/Internet Protocol), 250
 - packets, 259
 - TDMA (time division multiple access), 277
 - terminals, 15
 - tessellation, 98
 - texels, 100-102
 - text
 - cookie-cutter text, printing, 318-319
 - fonts, bitmap tables, 142
 - in-line formatting, 76
 - Microsoft Word, formatting, 77
 - outline fonts, printing, 317, 320-321
 - styles, 76
 - text mapping, 76
 - vector fonts, printing, 317
 - text mapping, 76
 - texture map pixels, 102
 - TFT (thin-film transistor), 174
 - thermal paste, 186
 - thin-film transistor (TFT), 174
 - threaded, 38
 - Timeline, Facebook, 303
 - time-staggered queue, 39
 - tokens, 71-72
 - toner (laser printers), 323
 - toroid coil, 15
 - torrenting, 308-309
 - touch, how devices recognize touch, 208
 - touch pads, 170
 - Google Glass, 179
 - touchscreens
 - how devices recognize touch, 208
 - iPhones, 153
 - trackers (torrenting), 308
 - traditional structure, 307

traffic load, 267
 trailers (packets), 259
 transferring data
 bandwidth, 268-269
 carriers, 268
 Internet, 280-281
 networks, 278-279
 packets, 258-259
 transformers, 14
 PC power supplies, 151
 step-down transformers, 14
 transistors, 19
 how they work, 20–21
 NOT logic gate, 34
 PC power supplies, 151
 transistor switches, 20, 34
 transmitters, 274
 transmitting data
 DSL, 271
 modems, cable modems, 272-273
 POTS, 271
 transport layer, 278
 transverse waves, 8
 triangulation, 98
 tricklers, 111
 trigger buttons, 210
 trilinear filtering, 102
 trough, 240
 true color, 87
 tunnel barrier, 162
 twisted-pair cable, 256
 Twitter, 304–305

U

ultraviolet photons, 229
 uncertainty principle, 340
 Universal DSL (digital subscriber lines), 271
 universal product code (UPC), 220

UPC (universal product code), 220
 upstream data
 cable modems, 272
 DSL, 271
 URL (Universal Resource Locators), 288
 cookies, 292-293
 USB, 156
 USB connections, 135
 USB debugging mode, 191
 USB ports, 133, 170
 USB3 (Universal Serial Bus 3), 156-157
 User Database Protocol, 283

V

variable-length field records, databases, 79
 variable resistors, 15
 variable tokens, 72
 variables, 66, 70
 variations, 93
 VBS.Hard.A@mm virus, 115
 VDSL (very high-speed digital subscriber lines), 271
 vector fonts, printing, 317
 vertex shaders, 103
 vertices, 98
 vibrations, smartphones, 214–215
 video cards, 133
 double anti-aliased graphics, 183
 gaming, 182-183
 video on demand (VOD), 285
 video ports, 133
 views, hidden views, 99
 virtual pixels, 87
 viruses, 112–113
 email, 114–115
 VisiCalc, 58
 VOD (video on demand), 285

voice coils, 215
 VoIP (voice over internet protocol), 277
 voltage, 13, 240
 von Neumann, John, 55

W

WAN (Wide-Area Networks), data transfers, 280
 water-cooled PCs, 188
 waveforms, 268
 wavelength, 9
 wavelet, 89
 waves, 8
 carrier waves, 11
 electromagnetic waves, 9
 frequency, 9
 how they carry information, 10–11
 input signals, 11
 longitudinal waves, 8
 mechanical waves, 9
 microwaves, 11
 radio waves, 11
 transverse waves, 8
 wavelength, 9
 web browsers, 288-289
 web pages, displaying, 290-291
 Web denizens, 252
 web pages
 displaying, 290-291
 how browsers display, 290–291
 Web. See Internet
 websites, 287
 cookies, 292-293
 domain names, 288
 eBay, 296
 Google, 294-295
 mirror sites, 289
 page files, 288
 URL, 288

While command, 68
 white balance (digital cameras), 193
 whitelists, 123
 wide, flat drive cables, 133
 wideband adaptive multi-rate coding, 241
 Wi-Fi, 267
 AP, 261
 Bluetooth, 262-263
 extension points, 261
 hot spots, 261
 IP addresses, 261
 probe requests, 260
 stations, 260-261
 WLAN, 261
 Wi-Fi networks, 256
 Wiimote, 211
 Willow Glass, 209
 Windows, 63
 wipers, 15
 wireframe, 101
 wireless networks, 256
 Bluetooth, 262-263
 WLAN (Wireless Local Area Networks), 261
 word addresses, 26
 WordPerfect, 60
 World of Warcraft, 104
 World Wide Web, 251
 wrappings, 14
 write-black printing method, 322
 write-white printing method, 322
 writing data to RAM, 22-23
 WWW (World Wide Web).
 See Internet

Z

z-buffering, 99
 zipping files
 lossless compression, 160-161
 lossy compression, 161
 zoned bit recording, 158
 z-sorting, 99

X-Y

xDSL (digital subscriber lines), 270-271
 XOR gates, 35