How Computers Work

Ninth Edition

Ron White

Illustrated by Timothy Edward Downs

QUE®

800 East 96th Street
Indianapolis, IN 46240

How Computers Work, Ninth Edition

Copyright © 2008 by Ron White

ISBN-13: 978-0-789-73613-0

ISBN-10: 0-789-73613-6

Library of Congress Cataloging-in-Publication data is on file.

Printed in the United States of America

First Printing: November 2008

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(s) 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 or from the use of the CD or programs accompanying it.

Bulk Sales

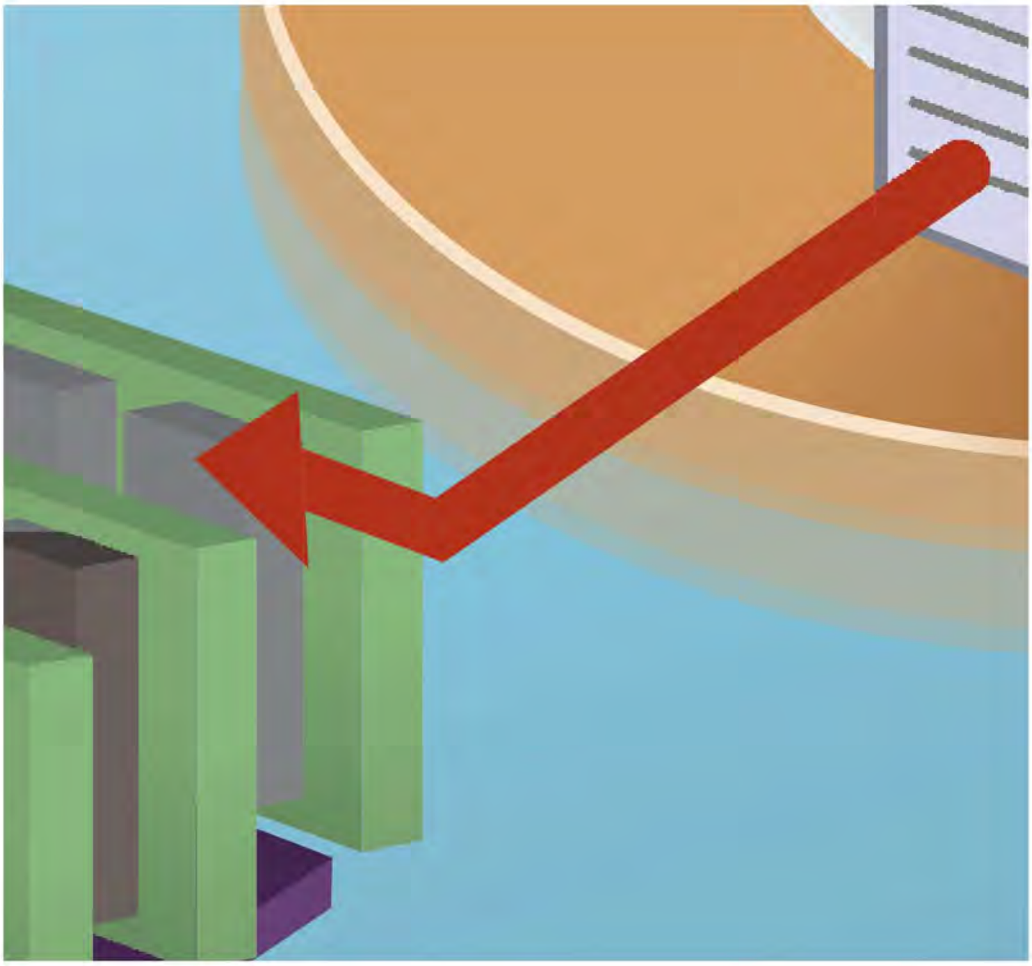
Que Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact

International Sales
international@pearsoned.com

Associate Publisher	Greg Wiegand
Acquisitions Editor	Todd Brakke
Development Editor	Todd Brakke
Managing Editor	Patrick Kanouse
Project Editor	Seth Kerney
Copy Editor	Megan Shaw
Indexer	Ken Johnson
Proofreader	Heather Waye Arle
Technical Editor	Mark Reddin
Publishing Coordinator	Cindy Teeters
Book Designer	Anne Jones



30,000 B.C.
Paleolithic peoples in central Europe record numbers by notching tallies on animal bones, ivory, and stone.

3400 B.C.
Egyptians develop a symbol for the number 10, simplifying the representation of large numbers.

2600 B.C.
The Chinese introduce the abacus. It was used in China for calculating the census as recently as A.D. 1982.

300 B.C.
Euclid's Elements summarizes all the mathematical knowledge of the Greeks. It is used for the next 2,000 years.

260 B.C.
The Maya develop a sophisticated base-20 system of mathematics that includes zero.

1500
Mechanical calculator invented by Leonardo da Vinci.

1614
John Napier describes the nature of logarithms. He also builds Napier's Bones, the forerunner to the slide rule.

1621
William Oughtred invents the slide rule, which does not become obsolete for nearly 350 years.

1642
Blaise Pascal invents Pascaline, the first mechanical calculator. It was hand turned and could only add and subtract.

1670
Gottfried Leibniz improves upon Pascaline by adding multiplication, division, and square root capabilities.

1679
Leibniz introduces binary arithmetic.

PART 1

Boot-Up Process

CHAPTERS

1822
Charles Babbage invents the Difference Engine, a large mechanical calculator capable of addition and subtraction.

1830
Charles Babbage conceives of the Analytical Engine but dies before its completion.

1890
Herman Hollerith creates an electric tabulating system for U.S. Census Bureau.

1896
Hollerith forms the Tabulating Machine Company, which later becomes International Business Machines.

1902–1905
Albert Einstein discovers Theory of Relativity. He publishes it in dissertation at University of Zurich.

1904
John Ambrose Fleming develops vacuum tubes.

1926
Patent for semiconductor transistor that allowed electrical currents to flow through computer, passing data.

1936
Konrad Zuse creates a programmable, digital computing machine that introduces use of binary system and valves.

1943
British build Colossus, a machine to break German codes.

1943–45
U.S. Army builds ENIAC computer to calculate weapons' trajectories.

"I think there is a world market for maybe five computers."

—Thomas Watson, chairman of IBM, 1943

BEFORE your personal computer is turned on, it is a dead collection of sheet metal, plastic, metallic tracings, and tiny flakes of silicon. When you push the On switch, one little burst of electricity—only about 3–5 volts—starts a string of events that magically brings to life what would otherwise remain an oversized paperweight.

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

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

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

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

1944
Harvard University and IBM develop the Mark 1, which uses IBM punched cards.

1945
John von Neumann describes a general purpose electronic digital computer with a stored program.

1948
ENIAC scientists create Electronic Control, the first computer firm, and begin to build UNIVAC for Census Bureau.

1949
Popular Mechanics predicts: "Computers in the future may weigh no more than 1.5 tons."

1951
UNIVAC delivered to U.S. Census Bureau three years late. It uses magnetic tape for input instead of punched paper.

1952
A complaint is filed against IBM, alleging monopolistic practices in its computer business.

1952
UNIVAC predicts landslide victory for Eisenhower on CBS. Human forecasts predict tight race. UNIVAC wins.

1954
IBM brings out 650, the first mass-produced computer. It's a great success, with 120 installations in first year.

1954
Texas Instruments announces the start of commercial production of silicon transistors.

1956
Massachusetts Institute of Technology builds the first transistorized computer.

1958
Control Data Corporation introduces Seymour Cray's 1604. At $1.5 million, it's half the cost of the IBM computer.

1958
Jack Kilby completes first integrated circuit, containing five components on a single piece of silicon.

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

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

How Computers Used to Work

At the beginning of the 21st century, computers are such complex beasts—despite their relative youth—that it's difficult to imagine how such elaborate contraptions could have sprung fully grown from the brows of their creators. The explanation is, of course, that they didn't. The development of computers has been an evolutionary process, and often it's well nigh impossible to figure out which came first, the chicken of software or the egg of hardware.

This portion of the Difference Engine #1, a forerunner to Charles Babbage's Analytical Engine--the first true computer--was completed in 1821. It contained 2,000 handmade brass parts. The entire machine would have used 25,000 parts and would have weighed 3 tons. The Analytical Engine was never completed, although part of it was built by Babbage's son, Henry, in 1910, and was found to be "buggy."

Courtesy of IBM

Human attempts to create tools to manipulate data date back at least as far as 2600 B.C. when the Chinese came up with the abacus. Later on, Leonardo da Vinci created a mechanical calculator. When the slide rule was invented in 1621, it remained the mathematician's tool of choice until the electronic calculator took over in the early 1970s.

All the early efforts to juggle numbers had two things in common: They were mechanical and they were on a human scale. They were machines made of parts big enough to assemble by hand. Blaise Pascal's Arithmetic Machine used a system of gears turned by hand to do subtraction and addition. It also used punch cards to store data, a method that's survived well into the 20th century.

1960
2,000 computers are in use in the United States.

1970
Xerox creates the Palo Alto Research Center (PARC), which gave birth to many essential computer technologies.

1973
Architecture using CP/M operating system becomes the standard for the next eight years until MS-DOS is introduced.

1975
The first known use of the word Micro-soft appears in a letter from Bill Gates to his future partner, Paul Allen.

1977
Radio Shack introduces the TRS-80 Model 1, lovingly referred to by its hobbyist fans as the Trash 80.

1982
Compaq introduces the first IBM PC clone computer. Personal Computer is Time's "Man of the Year."

1986
Microsoft goes public at $21 a share, raises $61 million.

1965
Digital Equipment Company's first successful minicomputer, the PDP-8. At $18,000, soon 50,000 are sold.

1971
Intel's Ted Hoff designs 4004 chip, the first microprocessor. Price $200, with 2,300 transistors and 60,000 OPS.

1975
Popular Electronics announces the Altair 8800, the first personal computer.

1976
Stephen Jobs and Steve Wozniak show first Apple computer at Home Brew Computer Club, later known as Silicon Valley.

1981
IBM introduces its personal computer, which uses Intel's 16-bit 8086 processor.

1984
Apple introduces the Macintosh, a computer using a mouse and graphic interface.

In 1830, Charles Babbage invented—on paper—the Analytical Engine, which was different from its predecessors because, based on the results of its own computations, it could make decisions such as sequential control, branching, and looping. But Babbage's machine was so complex that he died in 1871 without finishing it. It was built between 1989 and 1991 by dedicated members of the Science Museum in London. The physical size and complex mechanics of these mechanisms limited their usefulness; they were good for only a few tasks, and they were not something that could be mass produced.

Mechanical devices of all types flourished modestly during the first half of the 20th century. Herman Hollerith invented a mechanized system of paper cards with holes in them to tabulate the U.S. Census. Later, in 1924, Hollerith's Computing-Tabulating-Recording Company changed its name to International Business Machines.

In 1888, Herman Hollerith, the founder of what was to become IBM, created a machine that used punched cards to tabulate the 1890 U.S. Census. The device tabulated the results in six weeks instead of the seven years it had taken to compile the census by hand.

Courtesy of Smithsonian Institute

Although no one could have known it at the time, the first breakthrough to the modern computer occurred in 1904 when John Ambrose Fleming created the first commercial diode vacuum tube, something Thomas Edison had discovered and discarded as worthless. The significance of the vacuum tube is that it was the first step beyond the human scale of machines. Until it came along, computations were made first by gears and then by switches. The vacuum tube could act as a switch turning on and off thousands of times faster than mechanical contraptions.

Vacuum tubes were at the heart of Colossus, a computer created by the British during World War II to break the codes produced by the Germans' Enigma encrypting machine. And the Germans reportedly came up with a general-purpose computer—one not limited to a specific task as Colossus was. But the German invention was lost or destroyed in the war.

The war also gave birth to ENIAC (Electronic Numerical Integrator Analyzer and Computer), built between 1943 and 1945 by the U.S. Army to produce missile trajectory tables. ENIAC performed 5,000 additions a second, although a problem that took two seconds to solve required two days to set up. ENIAC cost $500,000, weighed 30 tons, and was 100 feet long and 8 feet high. It contained 1,500 relays and 17,468 vacuum tubes.

Those same tubes that made ENIAC possible in the first place were also its Achilles' heel. Consuming 200 kilowatts of electricity each hour, the tubes turned the computer into an oven, constantly cooking its own components. Breakdowns were frequent. What was needed was something that did the job of the tubes without the heat, bulk, and fragility. And that something had been around since 1926.

The ENIAC, built between 1943 and 1945, was the first all-electronic computer. It used so much power that legend says the lights of surrounding Philadelphia dimmed when the ENIAC was switched on.

Courtesy of Smithsonian Institute

In 1926, the first semiconductor transistor was invented, but it wasn't until 1947, when Bell Labs' William Shockley patented the modern solid-state, reliable transistor, that a new era in computing dawned. The transistor did essentially the same thing a vacuum tube did—control the flow of electricity—but it was the size of a pea and generated little heat. Even with the transistor, the few computers built then still used tubes. It wasn't until 1954, when Texas Instruments created a way to produce silicon transistors commercially, that the modern computer took off. That same year IBM introduced the 650, the first mass-produced computer. Businesses and the government bought 120 of them the first year.

Four years later, Texas Instruments built the first integrated circuit by combining five separate components and the circuitry connecting them on a piece of germanium half an inch long. The integrated circuit led to the modern processor and has made a never-ending contribution to smaller and smaller computers.

The computer grew increasingly smaller and more powerful, but its cost, complexity, and unswerving unfriendliness kept it the tool of the technological elite. It wasn't until 1975 that something resembling a personal computer appeared. The January issue of Popular Electronics featured on its cover something called the Altair 8800, made by Micro Instrumentation and Telemetry Systems (MITS). For $397, customers got a kit that included an Intel 8080 microprocessor and 256 bytes of memory. There was no keyboard; programs and data were both entered by clicking switches on the front of the Altair. There was no monitor. Results were read by interpreting a pattern of small red lights. But it was a real computer cheap enough for anyone to afford. MITS received orders for 4,000 Altair systems within a few weeks.

The first computer cheap enough for individuals to afford was the Altair 8800, created by a small New Mexico firm, MITS. It cost $397 without a keyboard or screen.

Courtesy of The Computer Museum

The new computer was at first a toy for hobbyists and hackers. They devised clever ways to expand the Altair and similar microcomputers with keyboards, video displays, magnetic tape, and then diskette storage. Then two hackers—Stephen Jobs and Steve Wozniak—created a personal computer that came complete with display, built-in keyboard, and disk storage, and began hawking it at computer clubs in California. They called it the Apple, and it was the first personal computer that was powerful enough, and friendly enough, to be more than a toy. The Apple, along with computers from Radio Shack and Commodore, began appearing in businesses, sometimes brought in behind the backs of the people in white lab coats who ran the "real" mainframe computers in a sealed room down the hall. The information services—or IS, as the professional computer tenders came to be called–disparaged the new computers as toys, and at the same time they saw microcomputers as a threat to their turf.

The Apple, introduced in 1976, was an immediate hit partially because a program called VisiCalc, which did the math of an electronic ledger sheet, justified the computer as a business cost.

Courtesy of Apple Corp.

The development that finally broke the dam, unleashing microcomputers on a society that would forever after be different, was not a technical invention. It was a marketing decision IBM made when creating its first personal computer, the IBM PC. IBM wanted to keep the price down, and so it decided to build the computer from components that were already available off the shelf from several suppliers. IBM also made the overall design of the PC freely available to competitors. The only part of the machine IBM copyrighted was the BIOS, the crucial basic input/output system, computer code residing in a single chip that defined how software was to interact with the PC's hardware. The competition could create their own PCs as long as they duplicated the operations of IBM's BIOS without directly copying it.

While Apple continued to keep its design proprietary, IBM's openness encouraged the creation of IBM clones that could use the same software and hardware add-ons the PC used. And the clones, while competing with IBM, at the same time helped establish the IBM architecture as the machine for which software and add-on hardware developers would design.

KEY CONCEPTS

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

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

circuit board Originally, wires ran from and to any component in any electrical device, not just computers. A circuit board replaces the need for separate wiring with the metallic traces printed on the board—sometimes also on the bottom of the board and in a hidden middle layer. The traces lead to connections for processors, resistors, capacitors, and other electrical components. The importance of the circuit board is that its entire creation can be automated, and the board packs more components into an ever-smaller space.

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

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

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

expansion slot Most PCs have unused slots into which the owner can plug circuit boards and hardware to add to the computer's capabilities. Most slots today are personal computer interface (PCI) or it's next-generation sibling PCI-Express (PCI-E). One other slot, the accelerated graphics port (AGP), accepts a video card designed to move images out of memory quickly, although it is fast being replaced by PCI-E–you might also see shorter slots on older computers. These are industry standard architecture (ISA), the only type of slots on the first PC.

motherboard A sheet of plastic onto which metallic circuits have been printed and to the rest of the PC's components are connected. These components could be connected via a socket, such as with the CPU, a slot, as with graphics cards and memory modules or they may be built directly onto the motherboard, as with external ports, such as USB.

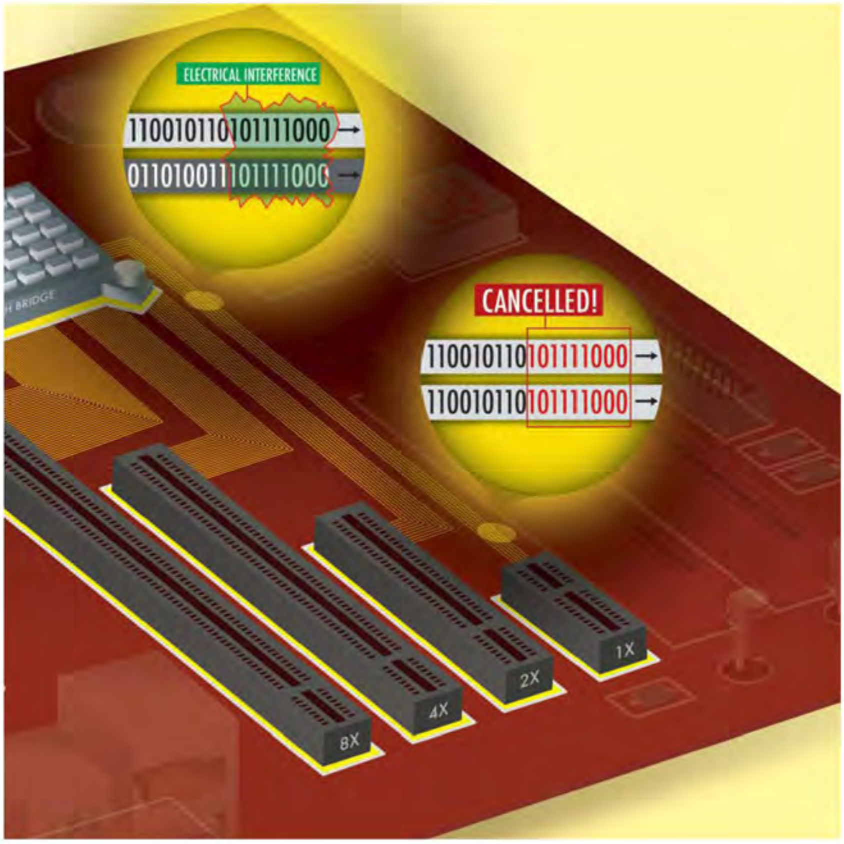
operating system Software that exists to control the operations of hardware. Essentially, the operating system directs any operation, such as writing data to memory or to disk, and regulates the use of hardware among several application programs that are running at the same time. This frees program developers from having to write their own code for these most basic operations.

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

system files Small disk files that contain software code that are the first files a computer reads from disk when it is booted. The system files contain the information needed, following the initial hardware boot, to load the rest of an operating system.

write and read Writing is the process by which a computer stores data in either RAM chips or on a disk drive. Reading is the process by which a computer transfers data or software code from a drive to RAM or from RAM to the microprocessor.

CHAPTER 2 How Circuits Juggle Data



UNDER the big top of your computer, the microprocessor—the central processing unit—is always the headliner. You don't see ads or reviews raving that a new PC has "revolutionary 100-ohm resistors!" Hard drives and graphics cards have the top supporting roles, but when it comes to the components on the motherboard—the mother of all boards—the CPU steals the spotlight.

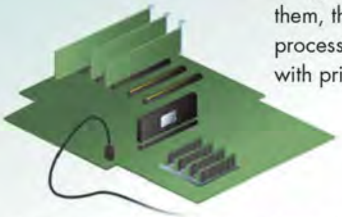
There are many good reasons for the CPU's fame, but like all stars, it owes a lot to the little components—the circuit board supporting parts without which the central microprocessor would be only a cold slab of silicon. Without them, electronic messages meant for the CPU would crash into the chips and each other, moving so fast there would be no time to read their license plates. Contrarily, other messages would arrive like dying murder victims at the ER, so weak they can only whisper their crucial clues in pulses so faint the microprocessor can't understand them.

These supporting casts were much smaller in the early days of PCs because the starring role of the motherboard was much smaller. It was basically a platform for the microprocessor, which is a transportation grid for conveying signals back and forth between the CPU and the parts the CPU controlled—disc controller cards, video cards, sound cards, input/output cards. Back then, nearly everything that made a PC a PC was handled by expansion cards, which was handy because you could easily update a single component as innovation and budget allowed. Today, almost any computer comes with sound, video, disk controllers, and an assortment of input/output options all on the motherboard. Your computer's character is largely determined by the motherboard's capability, and those capabilities are largely defined by the parts that populate it.

So here, ladies and gentlemen, are the little parts that make it all possible.

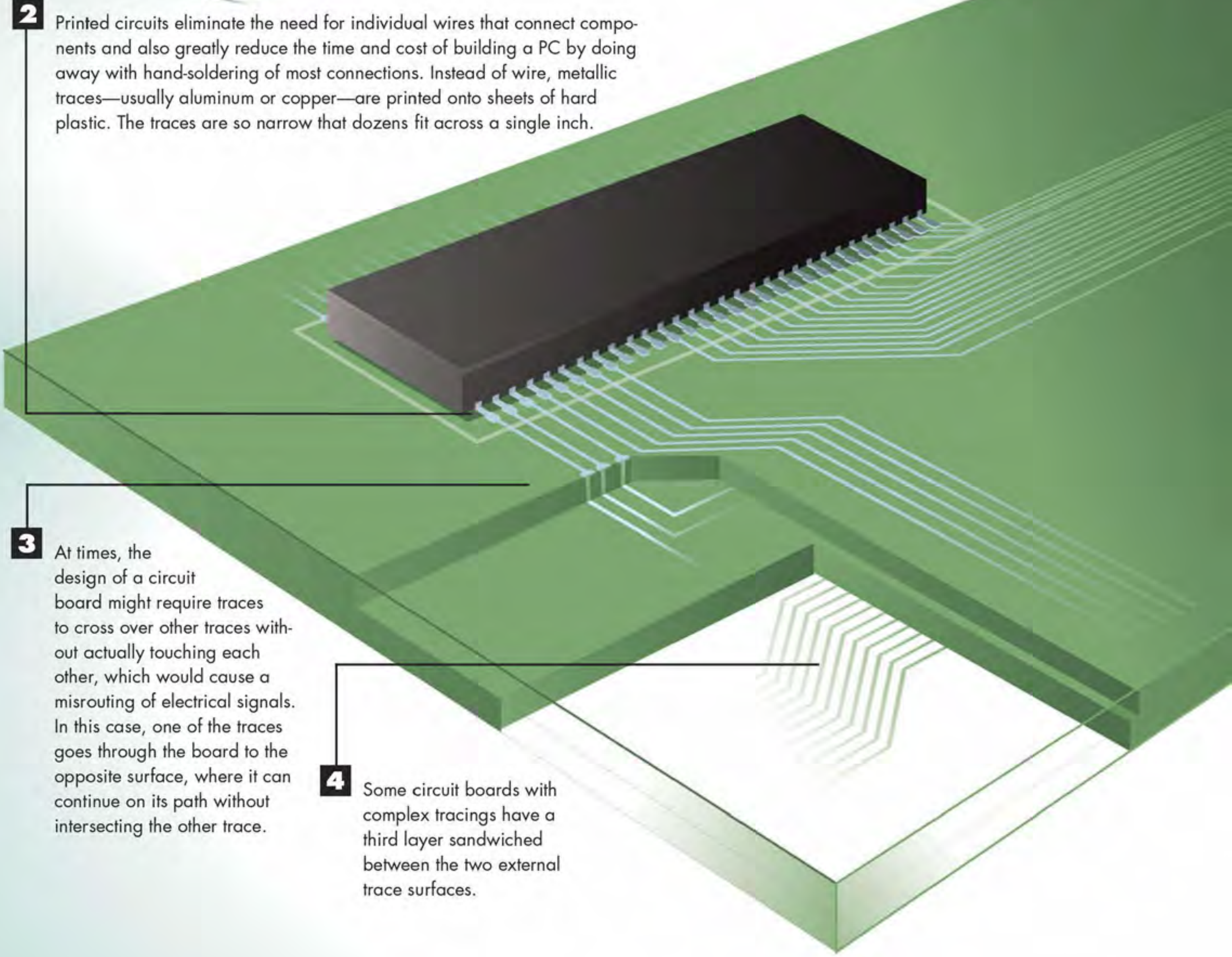
- Tiny metallic cans house the circuit board's strong men—the **resistors**! They clamp down on the wild, untamed electricity before it has the chance to burn up the rest of the components.
- Wrapped in ceramic casing and coats of plastic are the voracious, singing **capacitors**! They hum as they consume great quantities of electrical charge, holding it in so other components can have a steady supply or a sudden surge of electricity when they need it.
- Scattered everywhere on the motherboard are those mysterious, miniature monoliths, the **microchips**! What the millions of transistors do inside them is known to only a few.
- And connecting them all are stripes of copper and aluminum, **circuit traces,** that tie it all together so the individual players are a coordinated whole.

How Circuit Boards Work



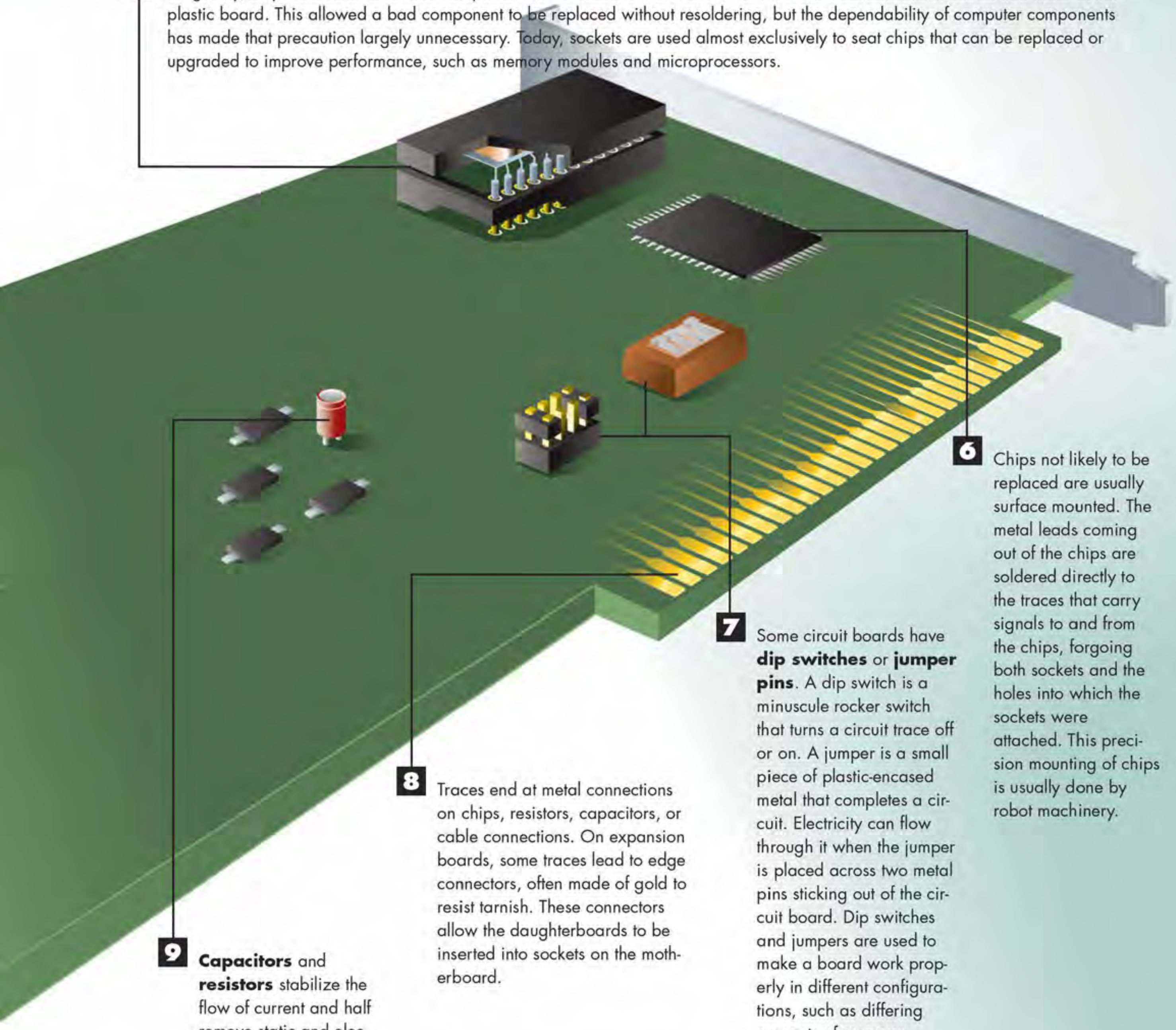
1 Most of the components in a PC are mounted on printed circuit boards. The motherboard is the largest printed circuit. Expansion cards and memory chips plug into the motherboard. The memory chips are grouped together on small circuit boards to create **dual in-line memory modules**, or **DIMMs**. Components that at first glance don't appear to have circuit boards often have them, they're just hidden inside their housings. Disk drives and some microprocessors, such as the Athlon 64 and Core 2, tie their internal parts together with printed circuits.

2 Printed circuits eliminate the need for individual wires that connect components and also greatly reduce the time and cost of building a PC by doing away with hand-soldering of most connections. Instead of wire, metallic traces—usually aluminum or copper—are printed onto sheets of hard plastic. The traces are so narrow that dozens fit across a single inch.



3 At times, the design of a circuit board might require traces to cross over other traces without actually touching each other, which would cause a misrouting of electrical signals. In this case, one of the traces goes through the board to the opposite surface, where it can continue on its path without intersecting the other trace.

4 Some circuit boards with complex tracings have a third layer sandwiched between the two external trace surfaces.



5 Originally, chips and other electrical components were inserted into sockets that had metal connections soldered into holes in the plastic board. This allowed a bad component to be replaced without resoldering, but the dependability of computer components has made that precaution largely unnecessary. Today, sockets are used almost exclusively to seat chips that can be replaced or upgraded to improve performance, such as memory modules and microprocessors.

6 Chips not likely to be replaced are usually surface mounted. The metal leads coming out of the chips are soldered directly to the traces that carry signals to and from the chips, forgoing both sockets and the holes into which the sockets were attached. This precision mounting of chips is usually done by robot machinery.

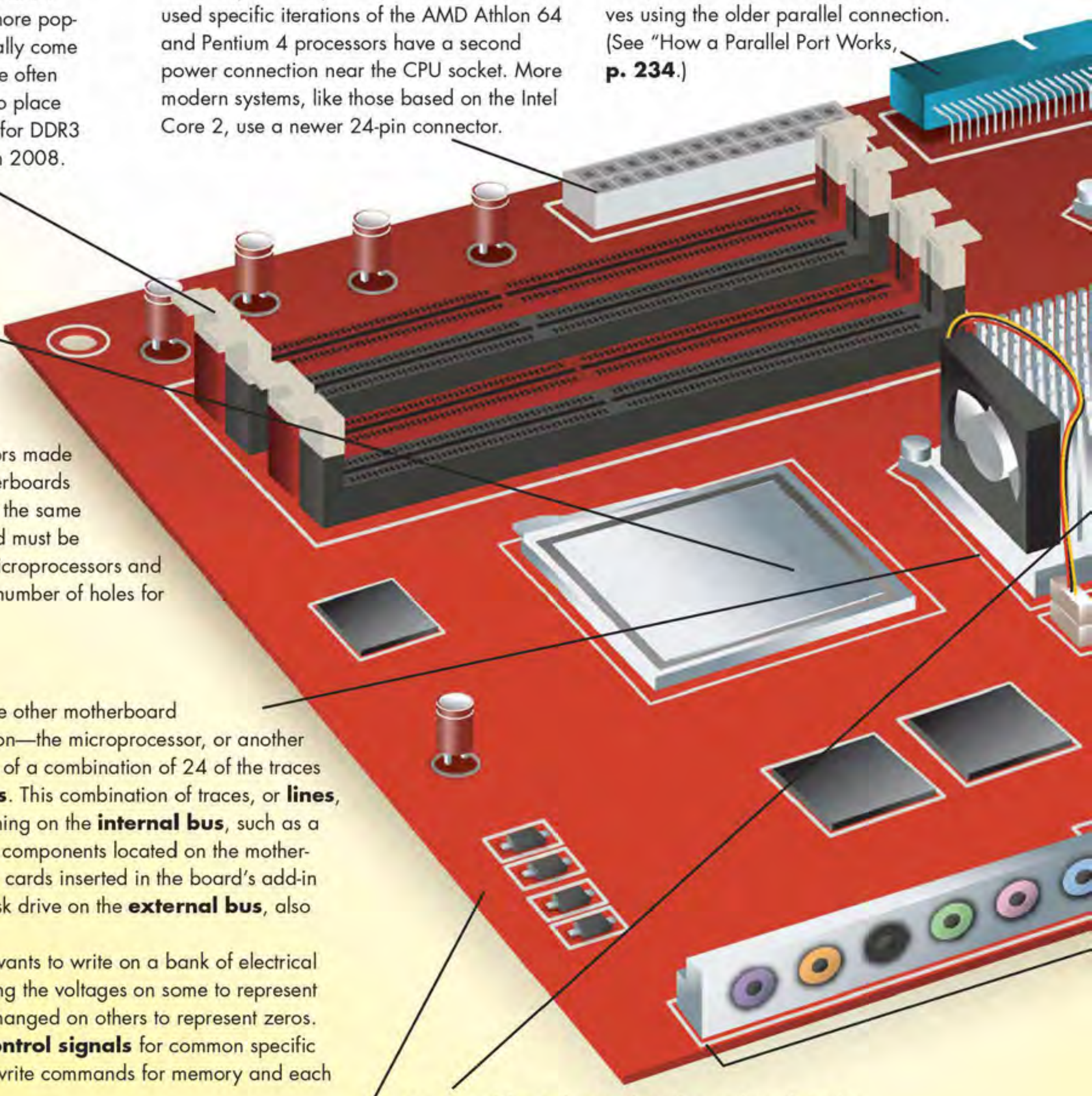
7 Some circuit boards have **dip switches** or **jumper pins**. A dip switch is a minuscule rocker switch that turns a circuit trace off or on. A jumper is a small piece of plastic-encased metal that completes a circuit. Electricity can flow through it when the jumper is placed across two metal pins sticking out of the circuit board. Dip switches and jumpers are used to make a board work properly in different configurations, such as differing amounts of memory.

8 Traces end at metal connections on chips, resistors, capacitors, or cable connections. On expansion boards, some traces lead to edge connectors, often made of gold to resist tarnish. These connectors allow the daughterboards to be inserted into sockets on the motherboard.

9 **Capacitors** and **resistors** stabilize the flow of current and half remove static and electrical surges or drops.

10 **Pin connectors** are used by **ribbon cables**—wide, flat collections of wires joined together—for internal connections between circuit boards and disk drives.

How the Motherboard Brings It All Together



Memory Slots: Current slots support either DDR (184 pins) or DDR2 RAM (240 pins), which is now the more popular type of memory. Slots usually come two or four to a board, and are often color-coded to tell you where to place matching memory cards. Look for DDR3 to start appearing in systems in 2008.

Power Supply Connections: Older boards have only one 20-pin connector. Boards that used specific iterations of the AMD Athlon 64 and Pentium 4 processors have a second power connection near the CPU socket. More modern systems, like those based on the Intel Core 2, use a newer 24-pin connector.

IDE Connector: Connects to two EIDE/ATA hard drives and optical drives using the older parallel connection. (See "How a Parallel Port Works," **p. 234**.)

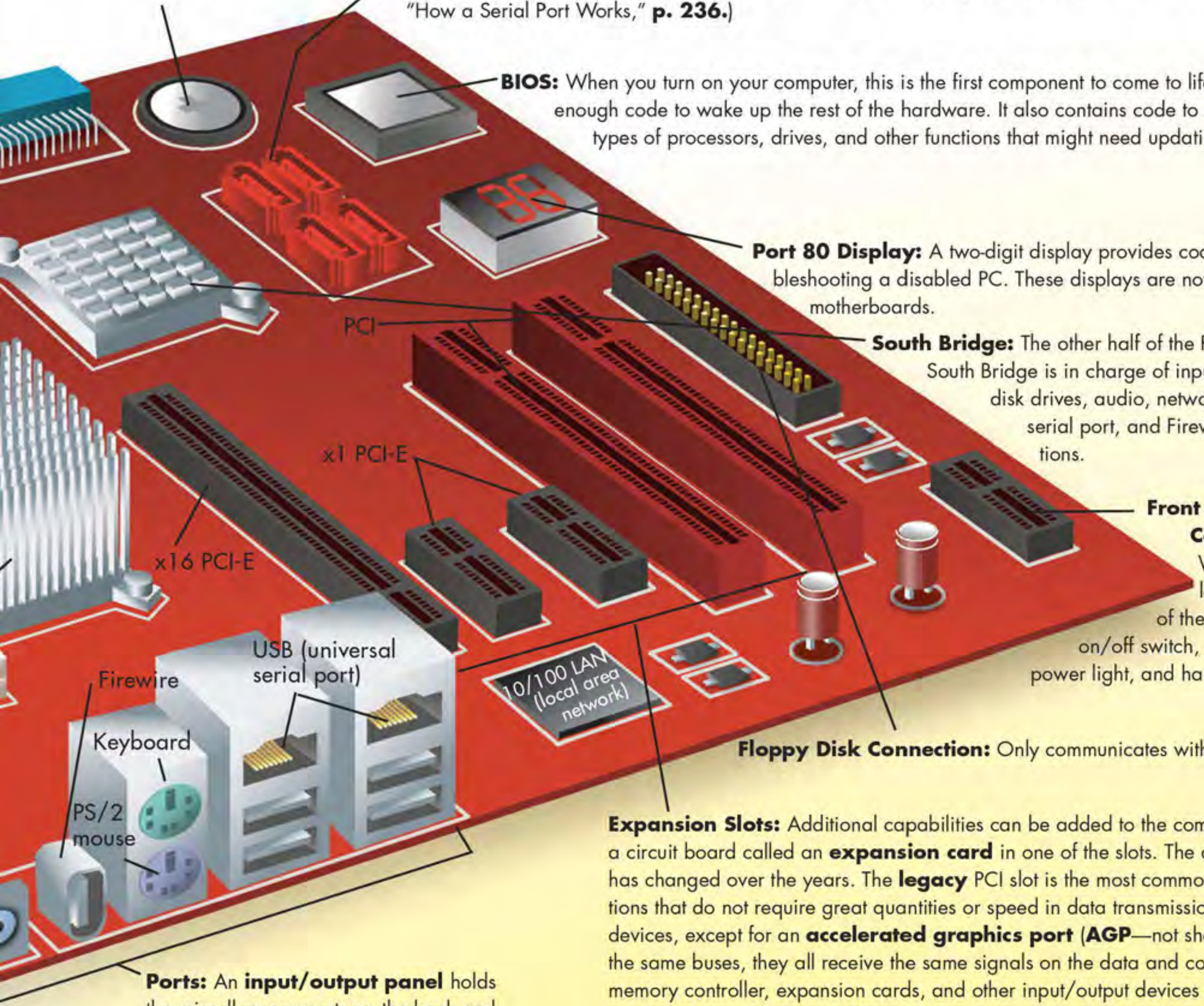
CPU Socket: This determines what kind of **microprocessor**, or **central processing unit (CPU)**, the motherboard uses. Boards are designed to work with processors made by either **Intel** or **AMD**. Motherboards do not work with all CPUs from the same company. The socket and board must be designed for specific lines of microprocessors and must have the right shape and number of holes for the chip's pins to fit.

Bus: To send data to any of the other motherboard components—a **write** operation—the microprocessor, or another component, raises the voltages of a combination of 24 of the traces that make up the **address bus**. This combination of traces, or **lines**, is the unique address of something on the **internal bus**, such as a location in memory; one of the components located on the motherboard itself, such as expansion cards inserted in the board's add-in slots; or a device, such as a disk drive on the **external bus**, also called the **expansion bus**.

The processor puts the data it wants to write on a bank of electrical traces, the **data bus**, by raising the voltages on some to represent ones and leaving voltages unchanged on others to represent zeros. Other lines are used to pass **control signals** for common specific commands, such as read and write commands for memory and each input/output device.

The Motherboard: As its name implies, the motherboard is the uniting element among all the chips and circuitry that make up a computer. Devices communicate with each other through the motherboard's circuits, from which they also draw their power. Motherboards come in different **form factors** that align the board with different sizes and styles of computer cases. They also come with different sockets that determine what types of chips and **expansion boards** they can accept.

North Bridge: The North Bridge and South Bridge together form the computer's **chip set**, secondary only to the processor in determining the performance and capabilities of a PC. The North Bridge chip either provides or controls the computer's graphics, RAM, and the **front side bus**, the main highway for data connecting graphics and memory to the CPU.



Battery: It keeps the BIOS chip alive.

SATA Connectors: Each connector, or **header**, is designed for the newer serial-ATA hard drives, providing faster delivery of drive data. These will eventually entirely replace IDE connections on most boards. (See "How a Serial Port Works," **p. 236.**)

BIOS: When you turn on your computer, this is the first component to come to life, providing enough code to wake up the rest of the hardware. It also contains code to support specific types of processors, drives, and other functions that might need updating occasionally.

Port 80 Display: A two-digit display provides codes used in troubleshooting a disabled PC. These displays are not found on all motherboards.

South Bridge: The other half of the PC **chip set**, the South Bridge is in charge of input/output with the disk drives, audio, networking, universal serial port, and Firewire communications.

Front Panel Connectors: Wires from these lead to the front of the PC for the on/off switch, reset switch, power light, and hard drive light.

Floppy Disk Connection: Only communicates with a floppy drive.

Ports: An **input/output panel** holds the miscellaneous ports on the back and front of the PC that are used for communicating with external devices. (Notice the lack of serial or parallel ports, which used to be standard. If they are needed for your computer's peripherals, they can be added with an expansion card.) Audio input/output ports are often part of the panel, although this board has a separate panel for them.

Expansion Slots: Additional capabilities can be added to the computer by plugging a circuit board called an **expansion card** in one of the slots. The design of the slots has changed over the years. The **legacy** PCI slot is the most common, used for functions that do not require great quantities or speed in data transmissions. Because all devices, except for an **accelerated graphics port** (**AGP**—not shown here) are on the same buses, they all receive the same signals on the data and control buses. The memory controller, expansion cards, and other input/output devices along the bus constantly monitor the command lines. When a signal appears on the write command line, for example, all the input/output devices recognize the command. The devices, alerted by the write command, turn their attention to the address lines. If the address specified on those lines is not the address used by a device, it ignores the signals sent on the data lines.

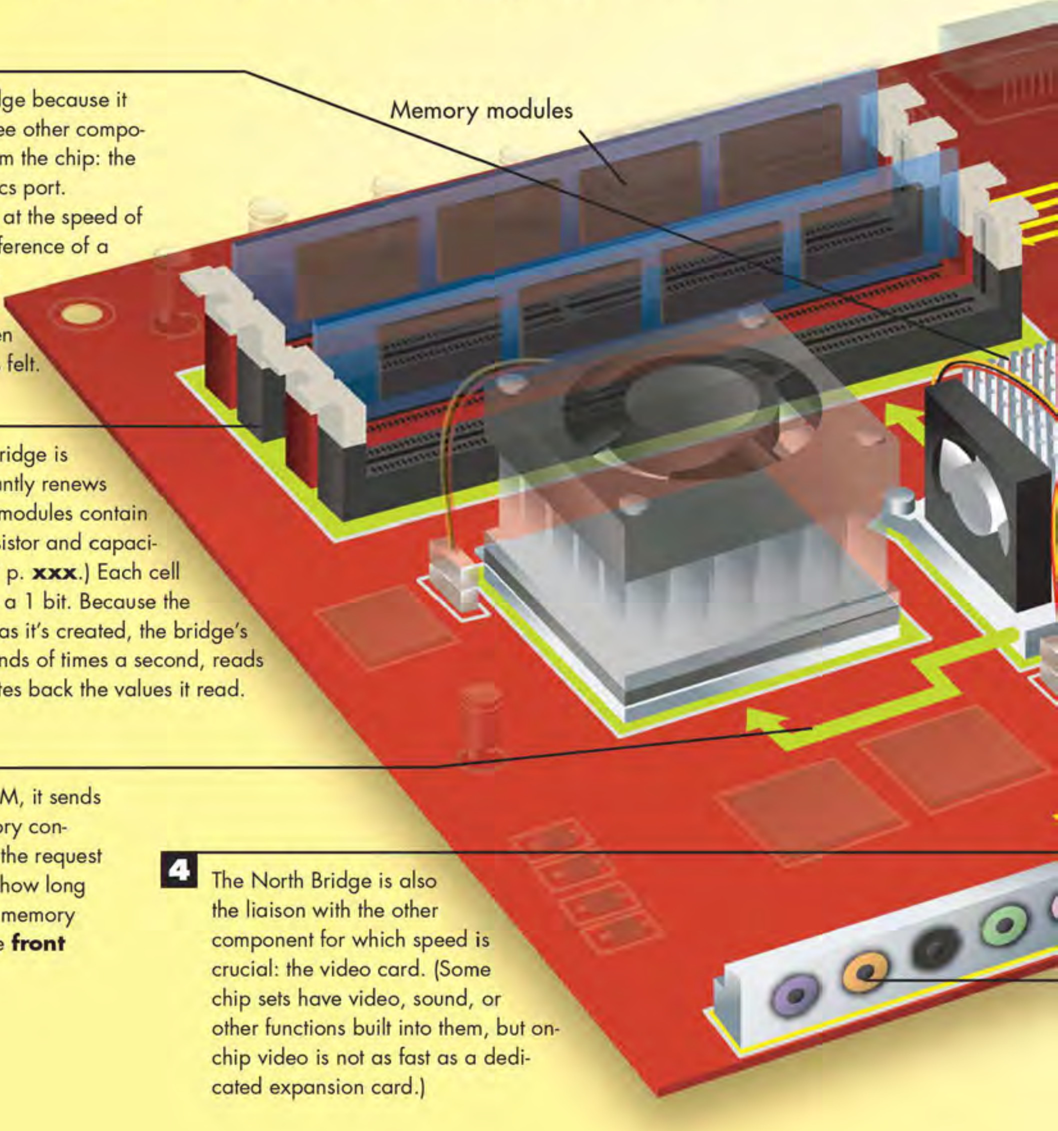
If the signals on the address lines match the address used by the adapter, the adapter accepts the data sent on the address lines and uses that data to complete the write command.

The **accelerated graphics port** is being pushed aside by the newer **PCI-Express** slots, which come in several denominations to make them the do-all, fit-all slot for every expansion board, not just graphics. The shorter ones here are **x1 PCI-E** shots and are common to all PCI Express slots. To handle graphics and sound data faster, the PCI-E slot can be expanded to **x4**, **x8**, or, shown here, **x16** slots, where the numbers represent multiples of the speed of an x1 PCI-E slot. Their ability to move data is indicated by the multiplier factor in their designations.

How the North and South Bridge Move Traffic

The personal computer has become so complex that even the most recent, powerful processors can't do the entire job of managing the flow of data by themselves. The CPU has been given help in the form of the **chip set**, located nearby on the motherboard. The chip set traditionally consists of two microchips, often referred to as the **North Bridge** and the **South Bridge**, that act as the administrators to the CPU, or chief executive. The chip set bridges logical and physical gaps between the CPU and other chips, all the time watching and controlling the input and output of specific components. The exact function of the chip set is constantly changing. The bridges have been put into one chip and in some designs, the CPU reclaims some functions. But in all cases, the bridges determine what kinds of memory, processors, and other components can work with that particular motherboard. There is an unfortunate trend to replace the names North Bridge and South Bridge with less elegant terms such as **Graphics Memory Controller Hub (GMCH)** and the **I/O Controller Hub (ICH)**, even though their basic purpose is the same. For our purposes here, we'll stick to the more seemly bridge nomenclature.

The North Bridge



1 You can distinguish the North Bridge because it resides as close as possible to three other components that get special attention from the chip: the CPU, the memory, and the graphics port. Although for something operating at the speed of light, you wouldn't think that a difference of a couple of inches could matter. But when you're counting in nanoseconds—billionths of a second—even small differences make themselves felt.

2 A crucial mechanism in the North Bridge is the memory controller, which constantly renews the memory modules (RAM). These modules contain memory cells, each made of a transistor and capacitor. (See "How a Transistor Works," p. **xxx**.) Each cell with an electrical charge represents a 1 bit. Because the charge begins to dissipate as soon as it's created, the bridge's memory controller endlessly, thousands of times a second, reads each of the millions of cells and writes back the values it read.

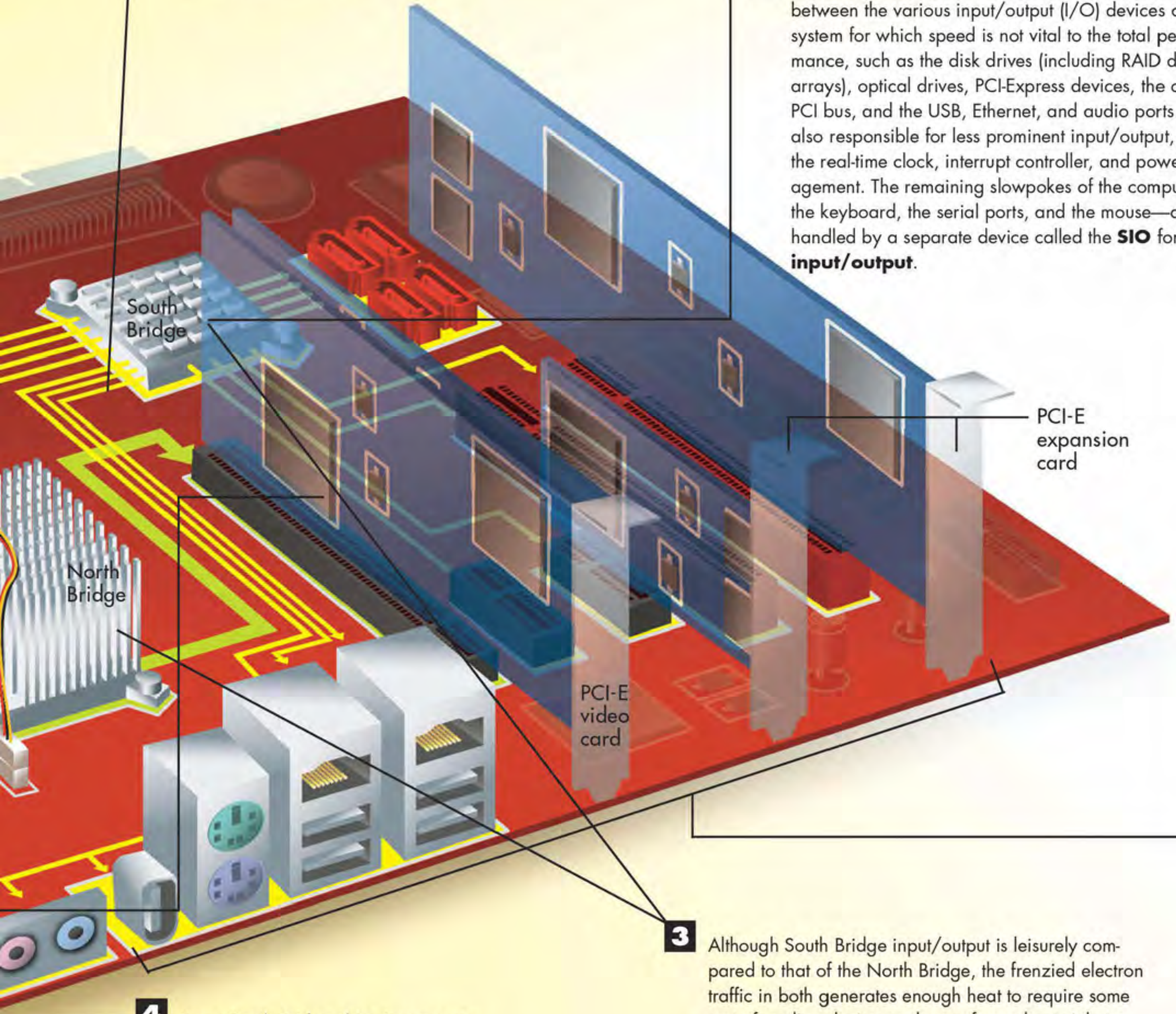
3 When the CPU needs data from RAM, it sends a request to the North Bridge memory controller. The controller, in turn, sends the request along to memory and tells the CPU how long the processor must wait to read the memory over a speedy connection called the **front side bus (FSB)**.

4 The North Bridge is also the liaison with the other component for which speed is crucial: the video card. (Some chip sets have video, sound, or other functions built into them, but on-chip video is not as fast as a dedicated expansion card.)

The South Bridge

1 The remaining connection of the North Bridge is to the South Bridge, ICH, or Input/Output Bridge, as the case may be.

2 The South Bridge primarily handles the routing of traffic between the various input/output (I/O) devices on the system for which speed is not vital to the total performance, such as the disk drives (including RAID drive arrays), optical drives, PCI-Express devices, the older PCI bus, and the USB, Ethernet, and audio ports. It is also responsible for less prominent input/output, such as the real-time clock, interrupt controller, and power management. The remaining slowpokes of the computer—the keyboard, the serial ports, and the mouse—are handled by a separate device called the **SIO** for **super input/output**.



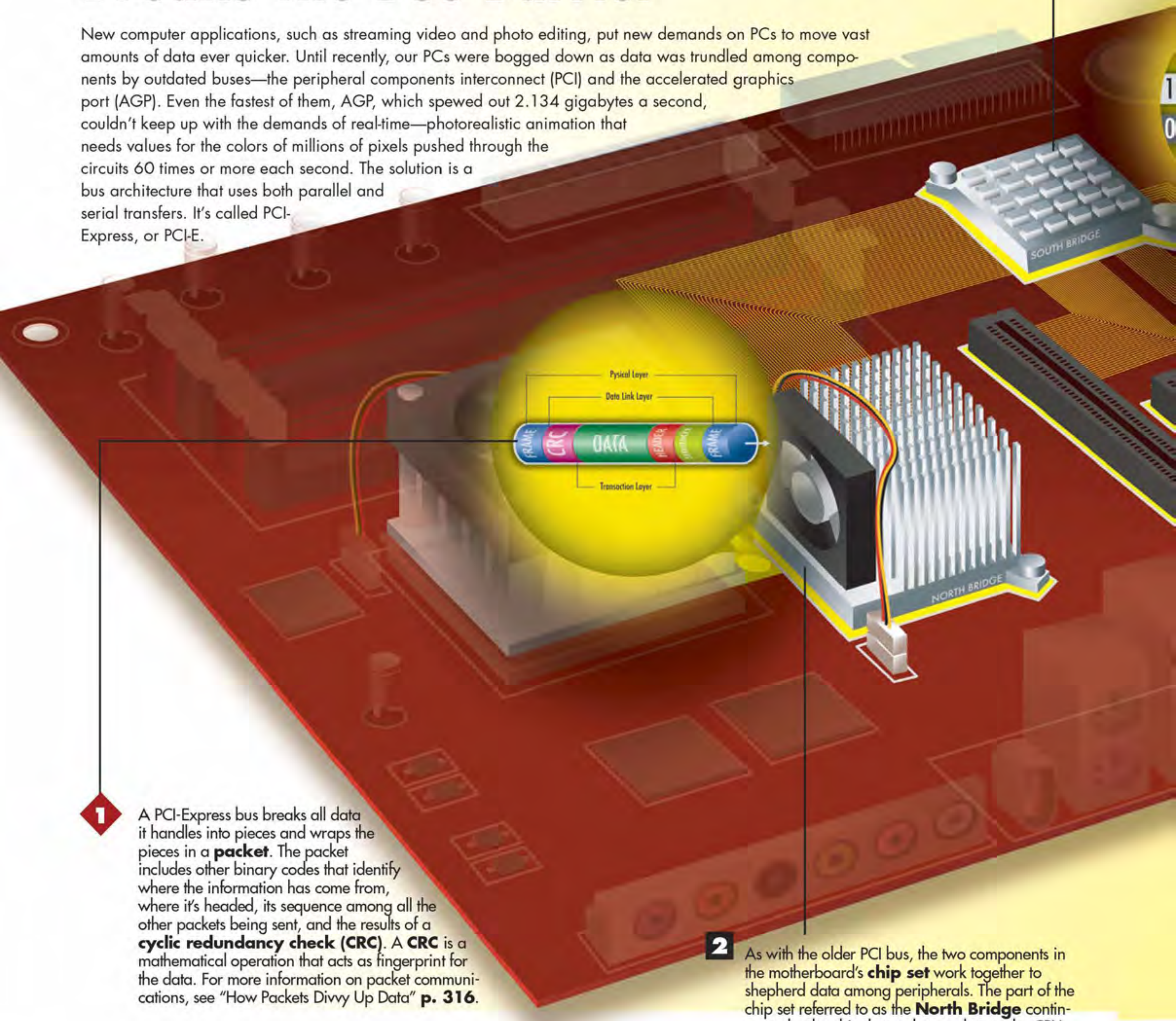
3 Although South Bridge input/output is leisurely compared to that of the North Bridge, the frenzied electron traffic in both generates enough heat to require some sort of cooling device, such as a fan or heat sink, to stop the chips from overheating.

4 Some South Bridge chips incorporate audio capabilities good enough to support Dolby Digital and THX multimedia audio.

5 Previously the North Bridge worked with the accelerated graphics port (AGP), providing a quick transfer of bitmaps from RAM to the AGP card's own memory. Now, however, the still faster PCI-Express (PCI-E) interface is replacing AGP video as you'll see when you turn the page.

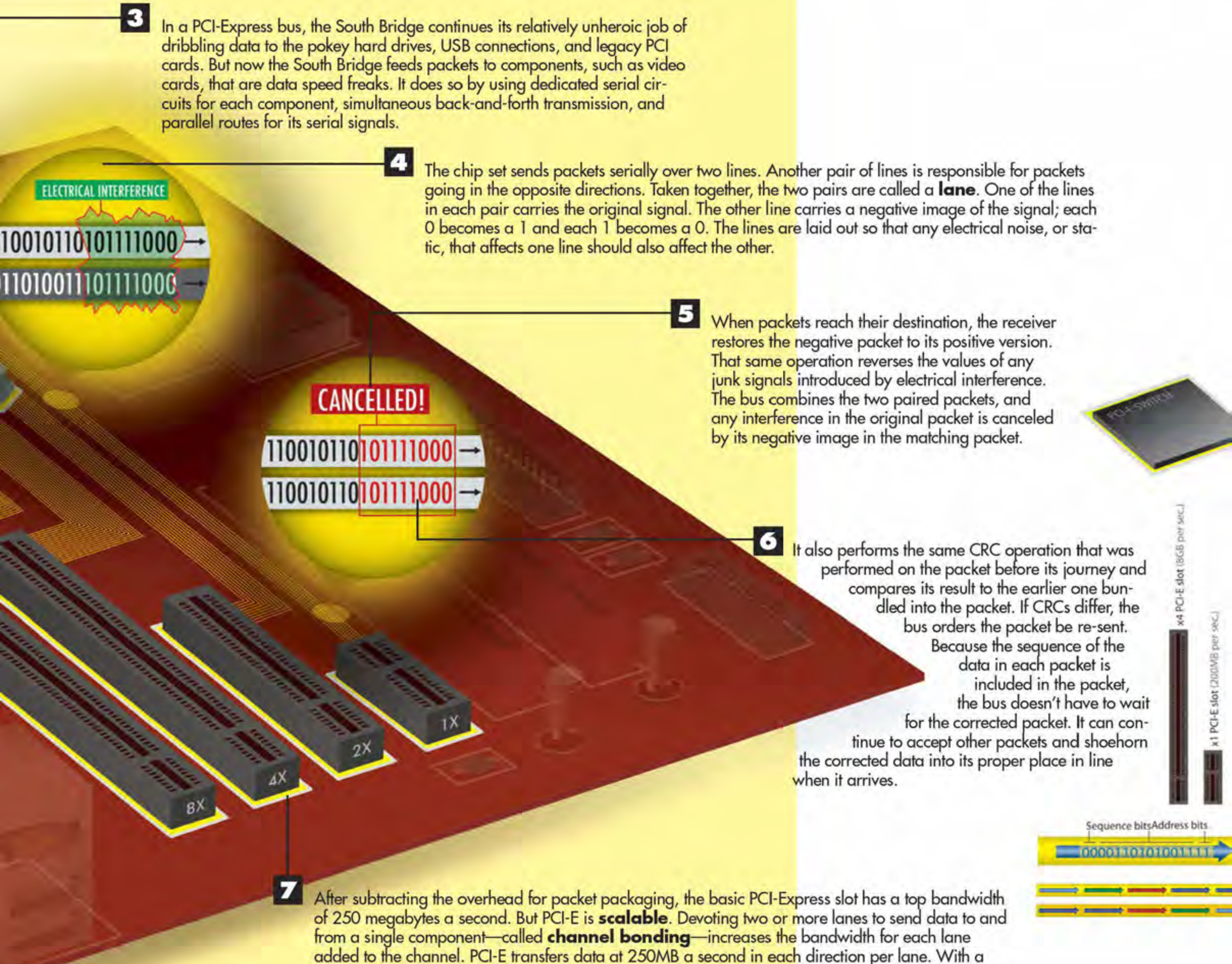
How PCI-Express Breaks the Bus Barrier

New computer applications, such as streaming video and photo editing, put new demands on PCs to move vast amounts of data ever quicker. Until recently, our PCs were bogged down as data was trundled among components by outdated buses—the peripheral components interconnect (PCI) and the accelerated graphics port (AGP). Even the fastest of them, AGP, which spewed out 2.134 gigabytes a second, couldn't keep up with the demands of real-time—photorealistic animation that needs values for the colors of millions of pixels pushed through the circuits 60 times or more each second. The solution is a bus architecture that uses both parallel and serial transfers. It's called PCI-Express, or PCI-E.



1 A PCI-Express bus breaks all data it handles into pieces and wraps the pieces in a **packet**. The packet includes other binary codes that identify where the information has come from, where it's headed, its sequence among all the other packets being sent, and the results of a **cyclic redundancy check (CRC)**. A **CRC** is a mathematical operation that acts as fingerprint for the data. For more information on packet communications, see "How Packets Divvy Up Data" **p. 316**.

2 As with the older PCI bus, the two components in the motherboard's **chip set** work together to shepherd data among peripherals. The part of the chip set referred to as the **North Bridge** continues to be the chip that rushes packets to the CPU and RAM, the components for which speed is most critical. It has traditionally passed less urgent packets to the **South Bridge** for handling.



3 In a PCI-Express bus, the South Bridge continues its relatively unheroic job of dribbling data to the pokey hard drives, USB connections, and legacy PCI cards. But now the South Bridge feeds packets to components, such as video cards, that are data speed freaks. It does so by using dedicated serial circuits for each component, simultaneous back-and-forth transmission, and parallel routes for its serial signals.

4 The chip set sends packets serially over two lines. Another pair of lines is responsible for packets going in the opposite directions. Taken together, the two pairs are called a **lane**. One of the lines in each pair carries the original signal. The other line carries a negative image of the signal; each 0 becomes a 1 and each 1 becomes a 0. The lines are laid out so that any electrical noise, or static, that affects one line should also affect the other.

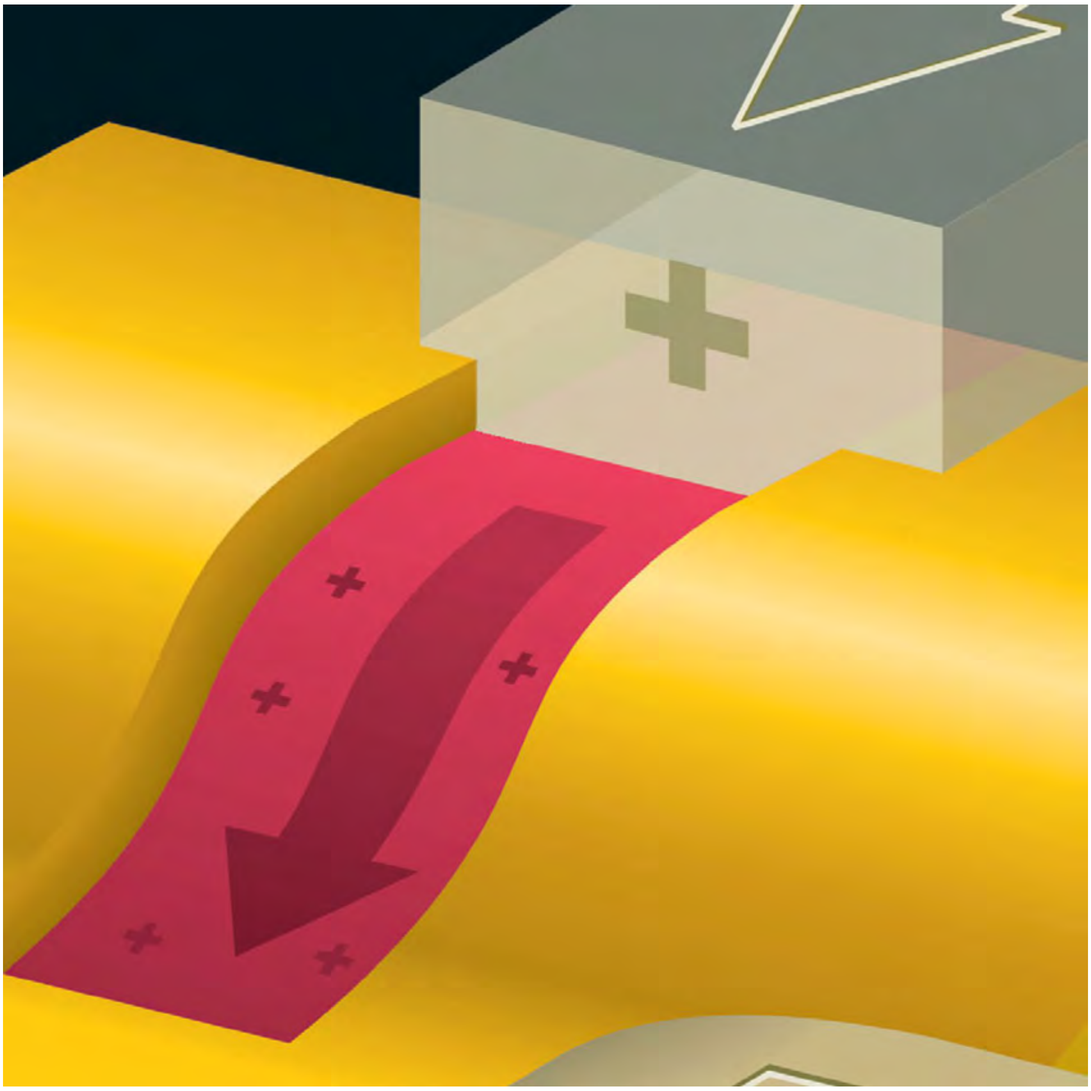
5 When packets reach their destination, the receiver restores the negative packet to its positive version. That same operation reverses the values of any junk signals introduced by electrical interference. The bus combines the two paired packets, and any interference in the original packet is canceled by its negative image in the matching packet.

6 It also performs the same CRC operation that was performed on the packet before its journey and compares its result to the earlier one bundled into the packet. If CRCs differ, the bus orders the packet be re-sent. Because the sequence of the data in each packet is included in the packet, the bus doesn't have to wait for the corrected packet. It can continue to accept other packets and shoehorn the corrected data into its proper place in line when it arrives.

7 After subtracting the overhead for packet packaging, the basic PCI-Express slot has a top bandwidth of 250 megabytes a second. But PCI-E is **scalable**. Devoting two or more lanes to send data to and from a single component—called **channel bonding**—increases the bandwidth for each lane added to the channel. PCI-E transfers data at 250MB a second in each direction per lane. With a maximum of 32 lanes, PCI-E allows for a total combined transfer rate of 8GB a second in each direction. That gives a single channel nearly twice the bandwidth of the older PCI and an eight lane slot a data rate comparable to the fastest version of AGP. You can identify the expansion slots with the increased bandwidth by comparing the slots' lengths. The basic PCI-E slot is about 24.5mm long. Each 13.5mm added to other slots represents another 250MB added to their bandwidth.

Goodbye to the Party Line

In the older PCI bus, all the devices share the same parallel circuits and receive the same data. The data includes an identifier that says which device the signals are destined for. All other devices simply ignore them. But like telephone users on a party line, the components can't receive data while some other device monopolizes the connection. The links in PCI-E are **point-to-point**. The South Bridge uses a **crossbar switch** to route incoming signals from one point to another down circuit lines dedicated to specific components. Data goes to several components at the same time. It's like talking on a private, single-line phone.



1350 B.C.
The Chinese use the first decimal. The floating-point microprocessor, an American invention that's dependent upon using decimals, follows 3,330 years later.

945 B.C.
The churchman Gerbert, who later becomes Pope Sylvester II, introduces the abacus and Hindu-Arabic math to Europe. But this new method for writing numbers does not catch on.

1623
Wilhelm Schickardt invents the Calculating Clock, the first mechanical calculator, based on the idea of Napier's Bones—rods used as mechanical aids to calculation and envisioned by John Napier in 1614.

1671
Gottfried Leibniz introduces the Step Reckoner, a device that can multiply, divide, and evaluate square roots.

1679
Gottfried Leibniz introduces binary arithmetic—a fundamental discovery showing that every number can be represented by the symbols 0 and 1 only.

1820
Charles Xavier Thomas de Colmar develops the "arithometer," the first practical and reliable device capable of the four basic arithmetic functions: addition, subtraction, multiplication, and division.

1823
Baron Jons Jakob Berzelius isolates silicon (Si), later to become the basic constituent of microchips.

1854
Augustus DeMorgan, in conjunction with Boole, formalizes a set of logical operations now known as DeMorgan transformations.

1886
Heinrich Rudolf Hertz, of megahertz (MHz) fame, proves that electricity is transmitted at the speed of light.

PART 2

How Microchips are the PC's Brain

CHAPTERS

1903
Nikola Tesla, a Yugoslavian scientist and inventor, patents electrical logic circuits called "gates" or "switches."

1904
John Ambrose Fleming experiments with Edison's diode vacuum tube (an invention Edison did not pursue), developing the first practical radio tubes. The vacuum tube's application to computers is not exploited until the late 1930s.

1926
First patent for semiconductor transistor. The transistor allows electrical currents to flow through a computer, allowing data to be passed through the machine.

1939
John Atanasoff conceptualizes the ABC—the firstprototype machine to use vacuum tubes. It isthe first electronic digital computer.

1943
First electronic general-purpose computer, the ENIAC, has 19,000 vacuum tubes, 1,500 relays, and consumes almost 200 Kilowatts of electricity.

1946
After leaving the University of Pennsylvania due to disagreements about patent ownership, J. Presper Eckert and John Mauchly—the two men who headed the ENIAC project—launch the first commercial computer company, Electronic Control Company. They begin work on the UNIVAC (Universal Automatic Computer) for the U.S. Census Bureau.

1947
Jay Forrester extends the life of a vacuum tube from 500 hours to 500,000, improving the reliability of the vacuum-based computers of the era.

1948
John Bardeen, Walter Brattain, and William Shockley of Bell Labs file for a patent on the first transistor.

"But what ... is it good for?"

—Engineer at the Advanced Computing Systems Division of IBM, 1968, commenting on the microchip

THOMAS Edison in 1883 noticed that electrical current flowing through a light bulb's filament could make the wire so hot that electrons boiled off, sailing through the vacuum inside the bulb to a metal plate that had a positive charge. Because Edison didn't see any way the phenomenon would help him perfect the light bulb, he only made a notation of the effect, which he named after himself. The effect sat on the shelf until 1904, when a former Edison employee, inventor John Fleming, went to work for the Marconi Radio Company. For his first assignment, finding a better way to receive distant radio signals, Fleming began experimenting with the Edison effect. He discovered that radio waves passing through an airless tube created a varying direct current, which could be used with headphones to reproduce the sound carried by the waves. Fleming named it the oscillation valve and applied for a patent. Marconi, though, chose another, less expensive technology: a crystal wave detector.

The discoveries sat on the shelf until radio pioneer Lee DeForest read about Fleming's valve and built one himself. The valve he created in 1906 had something new: a grid made of nickel wire placed between the filament and the plate. Applying even a small electrical charge to the grid disrupted the flow of electrons from the filament to the plate. It was the beginning of the vacuum tube, which essentially let a small amount of electrical current control a much larger flow of current.

If you're not at least old enough to be part of the baby boom generation, you might never have seen more than one type of vacuum tube—the cathode ray tube (CRT) that displays images on desktop PC monitors and the ordinary TV screen. Except for CRTs—and in the sound systems of audiophiles who swear vacuum tube amplifiers are better than transistorized amps—vacuum tubes are rarely used in modern electronics. It isn't the amplifying capabilities of the vacuum tube that have made it one of the seminal discoveries of science; It is the vacuum tube's capability to act as a switch. When a small amount of current was applied to the grid, it turned off a much stronger current. Turn off the electricity going to the grid, and the larger current is switched back on. On, off. Off, on. Simple.

1950
Whirlwind—the biggest computer project of its time—becomes operational. Whirlwind is not only fast, but uses only 400 vacuum tubes (compared to the nearly 18,000 in ENIAC).

1953
After spending four years in development, Jay Forrester and a team at the Massachusetts Institute of Technology install magnetic core memory into the Whirlwind computer, giving it a twice-as-fast access time of six microseconds.

1954
Texas Instruments announces the start of commercial production of silicon transistors.

1956
The Nobel Prize in physics is awarded to John Bardeen, Walter Brattain, and William Shockley for their work on the transistor.

1956
The first transistorized computer is completed, the TX-O (Transistorized Experimental computer), at the Massachusetts Institute of Technology.

1958
At Texas Instruments, Jack Kilby comes up with the idea of creating a monolithic device, an integrated circuit (IC) that would contain resistors and capacitors on a single piece of silicon. Kilby builds the first integrated circuit which contains five components connected by wires on a sliver of germanium half an inch long and thinner than a toothpick.

1959
Robert Noyce of Fairchild Semiconductors seeks a patent for a new invention: an integrated circuit with components connected by aluminum lines on a silicon-oxide surface layer on a plane of silicon.

1960
IBM develops the first automatic mass-production facility for transistors, in New York.

Essentially, a computer is just a collection of on/off switches, which at first doesn't seem very useful. But imagine a large array of light bulbs—say, 10 rows that each have 100 light bulbs in them. Each bulb is connected to a light switch. If you turn on the right combination of switches, you can put your name in lights.

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

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

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

The first vacuum tubes, such as this one made in 1915, were used to amplify radio signals. It wasn't until 1939 that tubes were used as switches in calculating machines.
Courtesy of AT&T

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

1960
The first integrated circuits reach the market, costing $120. NASA selects Noyce's invention for the on-board computers of the Gemini spacecraft.

1961
Fairchild Semiconductor releases the first commercial integrated circuit.

1964
Intel founder Gordon Moore suggests that integrated circuits would double in complexity every year. This later becomes known as Moore's Law.

1964
The first integrated circuit sold commercially is used in a Zenith hearing aid.

1968
Intel Corporation is founded in Santa Clara, CA, by Fairchild veterans Robert Noyce and Gordon Moore, employees #1 and #2. Andy Grove leaves Fairchild to become Intel's employee #4.

1969
Advanced Micro Devices Incorporated is founded.

1969
Intel sells its first commercial product, the 3101 Schottky bipolar 64-bit static random access memory (SRAM) chip. It is moderately successful.

1969
Intel's Marcian (Ted) Hoff designs an integrated circuit chip that can receive instructions and perform simple functions on data. Intel also announces a 1K RAM chip, a significantly larger capacity for memory chips.

took up an entire room now fits neatly on your lap. All microchips, whether they're microprocessors, a memory chip, or a special-purpose integrated circuit, are basically vast collections of transistors—switches—arranged in different patterns so that they accomplish different tasks. Doesn't sound like much but it's turning out to be nearly everything.

KEY CONCEPTS

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

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

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

ASCII Acronym for American Standard Code for Information Interchange.

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

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

bunny suit A total-body garment worn by personnel in a clean-room to reduce release of particles and contaminants into the air.

burn-in The process of exercising an integrated circuit at elevated voltage and temperature. This process accelerates failure normally seen as "infant mortality" in a chip. The resultant tested product is of high quality.

cache A block of high-speed memory where data is copied when it is retrieved from RAM. Then, if the data is needed again, it can be retrieved from the cache faster than from RAM. A *Level 1* cache is located on the CPU die. A *Level 2* is either a part of the processor die or packaging.

capacitor A component that stores an electrical charge.

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

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

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

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

logic A collection of circuit elements that perform a function, especially a set of elements that use digital logic and perform Boolean logic functions.

1971
Intel introduces its 4-bit bus, 108-KHz 4004 chip—the first microprocessor. Initial price is $200. Speed is 60,000 operations per second. It uses 2,300 transistors connected by circuits 10 microns wide. It can address 640 bytes of memory. The dimensions for the chip are 3×4 mm.

1972
Intel introduces the 8008, the first 8-bit microprocessor. Don Lancaster, a dedicated computer hobbyist, used the 8008 to create a predecessor to the first personal computer, a device Radio Electronics dubbed a "TV typewriter." It was used as a dumb terminal.

1974
The Intel 8080 microprocessor becomes the brains of the first personal computer: the Altair. Computer hobbyists could purchase a kit for the Altair for $367.

1975
The January edition of *Popular Electronics* features on its cover the Altair 8800 computer kit, based on Intel's 8080 microprocessor. Within months, it sells tens of thousands, creating the first PC back orders in history. Bill Gates and Paul Allen licensed BASIC as the software language for the Altair.

1979
A pivotal sale to IBM's new personal computer division makes the Intel 8088 processor the brains of IBM's new hit product: the IBM PC.

logic design Techniques used to connect logic building blocks or primitives (that is, AND gates, OR gates, and so on) to perform a logical operation.

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

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

micrometer A metric unit of linear measure that equals 1/1,000,000 meter, or 10,000 angstroms. The width of microprocessor circuits are measured in micrometers. The diameter of a human hair is approximately 75 micrometers. Also called "micron."

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

pin In plastic and metal wafer carriers, a protrusion of the wafer that fits into a matching hole in the wafer carrier for alignment when wafers are transferred.

pin grid array (PGA) A connection arrangement for microchips that features plug-in electrical terminal pins arranged in a matrix format, or an array.

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

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

register A set of transistors in a processor where data is stored temporarily while the processor makes calculations involving that data—a sort of electronic scratch pad.

SIMD (Single Instruction Multiple Data) A processor architecture that allows the same operation to be performed on multiple pieces of data simultaneously.

semiconductor A material (such as silicon) that can be altered to either conduct electrical current or block its passage. Microchips are typically fabricated on semiconductor materials such as silicon, germanium, or gallium arsenide.

silicon A brownish crystalline semimetal used to make the majority of semiconductor wafers.

source The part of a transistor from which electrical current flows when the transistor is closed.

transistor A microscopic switch that controls the flow of electricity through it, depending on whether a different electrical charge has opened or closed the switch.

wafer In semiconductor technology, a very thin piece of silicon that functions as the base material for building microchips. Also called a "slice."

1982
The Intel 80286 is the first Intel processor that can run all the software written for its predecessor. This software compatibility remains a hallmark of Intel's family of microprocessors. Within six years of its release, an estimated 15 million 286-based personal computers are installed around the world.

1985
Motorola announces the 68040, a 32-bit 25MHz microprocessor.

1985
The Intel 80386 microprocessor features 275,000 transistors—more than 100 times as many as the original 4004. It handles data 32 bits at a time and multitasks, meaning it can run multiple programs at the same time.

1989
The Intel 80486 DX makes point-and-click computing practical. The 486 is the first processor to offer a built-in math coprocessor, which speeds up computing because it offers complex math functions from the central processor.

1991
Advanced Micro Devices introduces the AM386 microprocessor family in direct competition with Intel's x86 processor line.

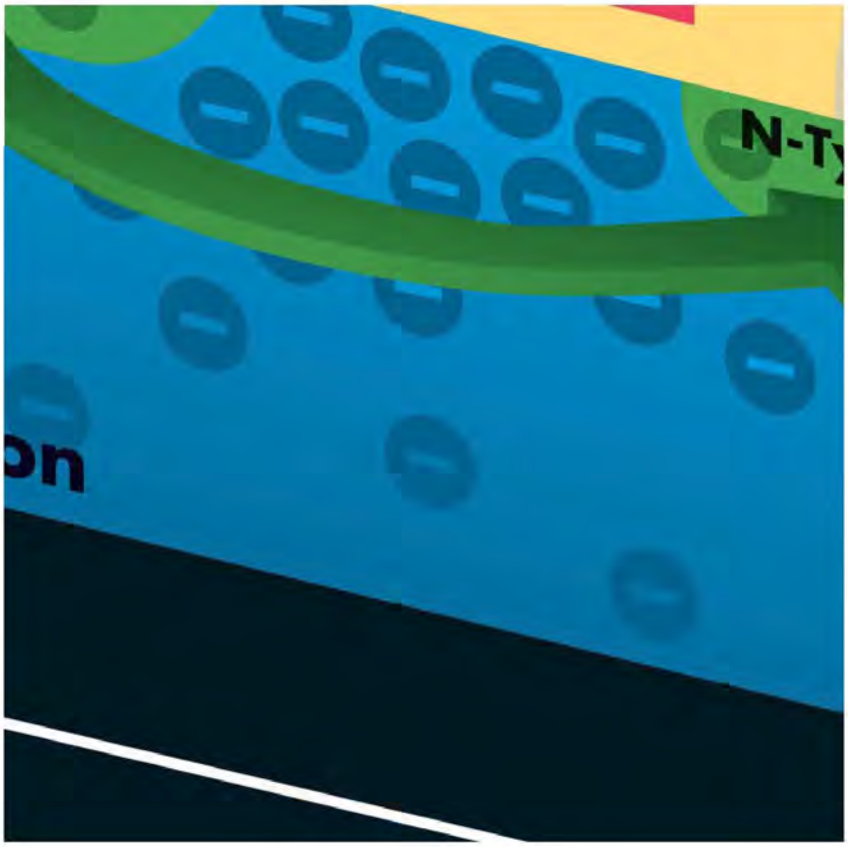
1993
Intel's new Pentium processor allows computers to more easily incorporate real-world data such as speech, sound, handwriting, and photographic images.

2003
AMD introduces the Athlon 64, the first 64-bit processor targeted for use in home computers.

2005
Both Intel and AMD release their first multicore processors.

CHAPTER 5

How Transistors Manipulate Data



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

Binary notation starts off simply enough:

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

Personal computers, such as the original IBM PC and AT systems based on the Intel 8088 and 80286 microprocessors, are 16-bit PCs. That means they can work directly with binary numbers of up to 16 places, or bits. That translates to the decimal number 65,535. If an operation requires numbers larger than that, the PC must first break those numbers into smaller components, perform the operation on each component, and then recombine the results into a single answer. More powerful PCs, such as those based on the Intel 80386, 80486, and Pentium, are 32-bit computers, which means they can manipulate binary numbers up to 32 bits wide—the equivalent in decimal notation of 4,294,967,295. The capability to work with 32 bits at a time helps make these PCs much faster and capable of directly using more memory.

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

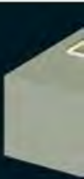
In addition, transistors make it possible for a small amount of electrical current to control a second, much stronger current—just as the small amount of energy needed to throw a wall switch can control the more powerful energy surging through the wires to give life to a spotlight.

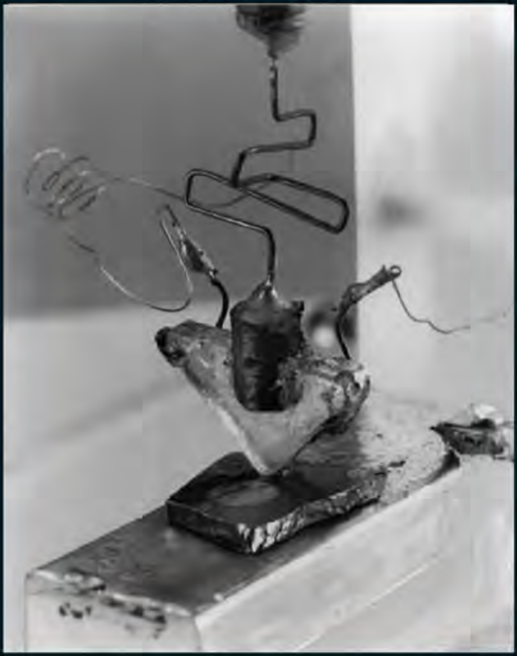
How a Little Transistor Does Big Jobs

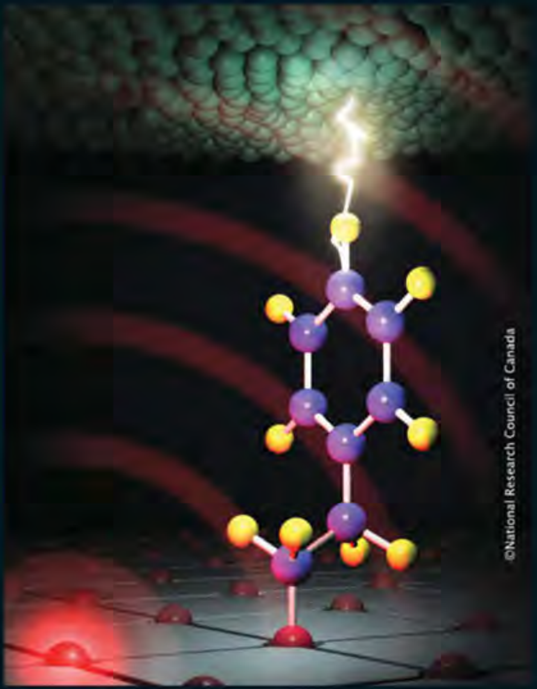
1 A small, positive electrical charge is sent down one aluminum lead that runs into the transistor. The positive charge spreads to a layer of electrically conductive polysilicon buried in the middle of nonconductive silicon dioxide. Silicon dioxide is the main component of sand and the material that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

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

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



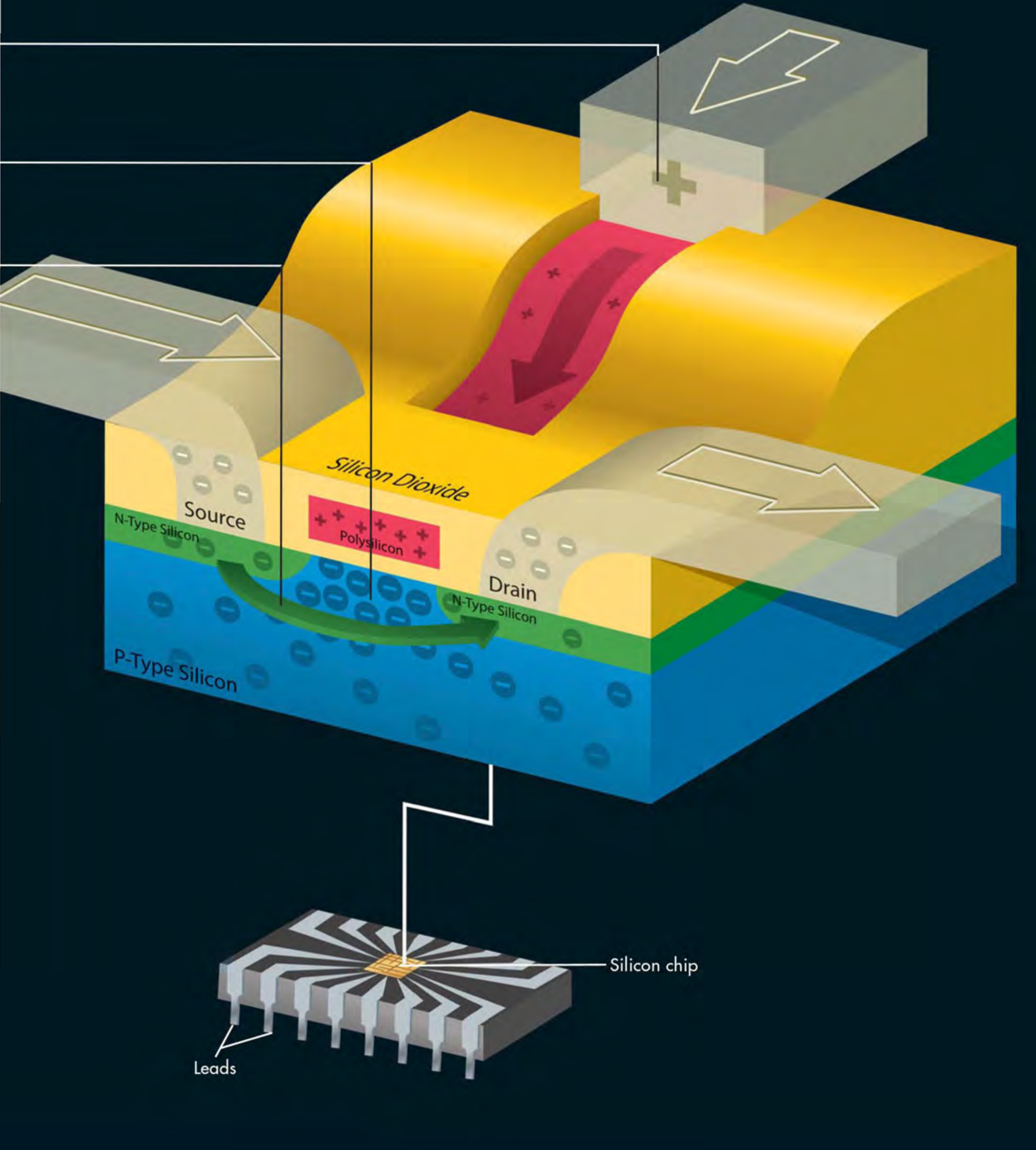




©National Research Council of Canada

Creating a Chip from Transistors

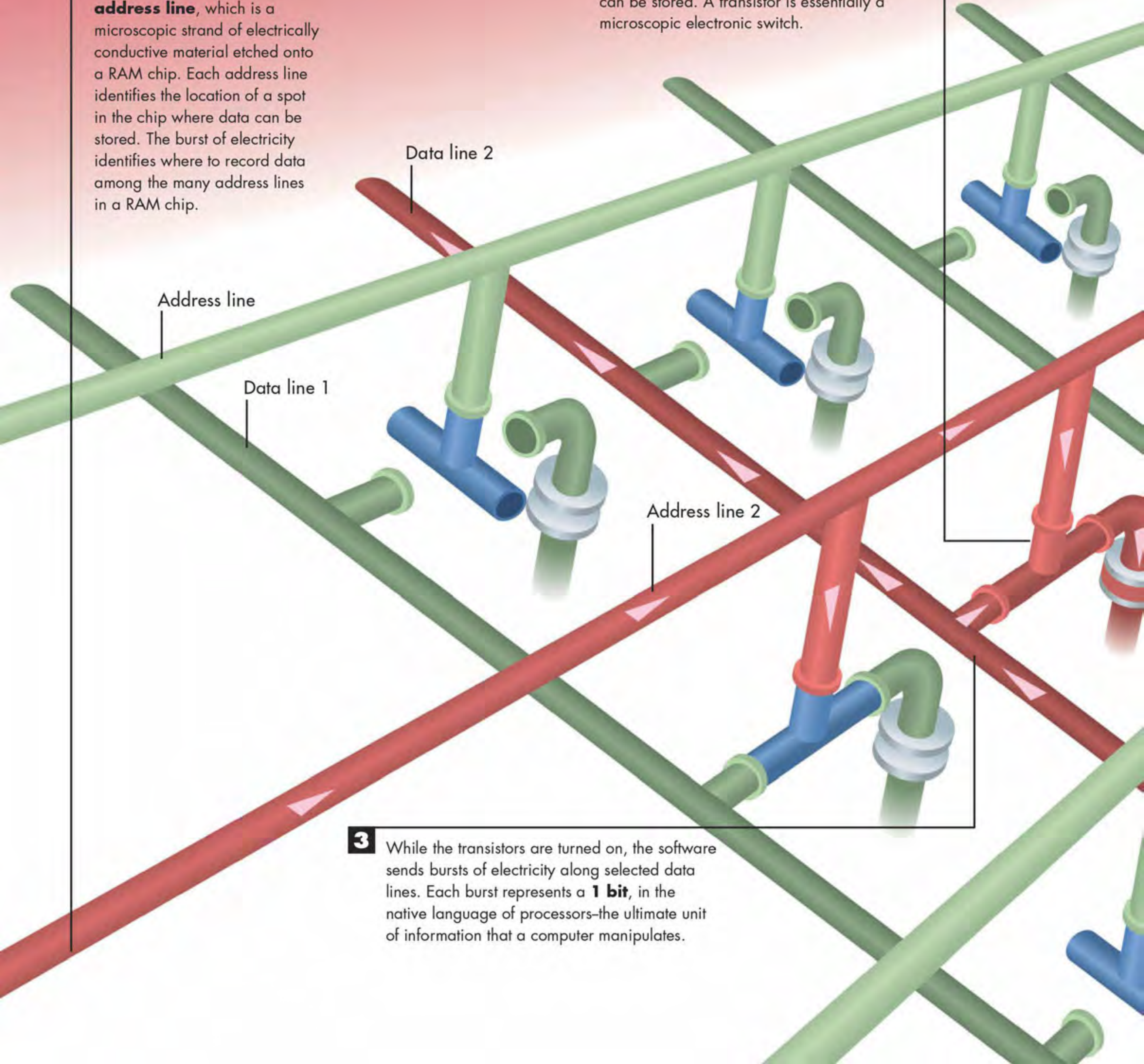
Thousands of transistors are connected on a single slice of silicon. The slice is embedded in a piece of plastic or ceramic material and the ends of the circuitry are attached to metal leads that expand to connect the chip to other parts of a computer circuit board. The leads carry signals into the chip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


Silicon Dioxide
Source
N-Type Silicon
Polysilicon
Drain
N-Type Silicon
P-Type Silicon
Silicon chip
Leads

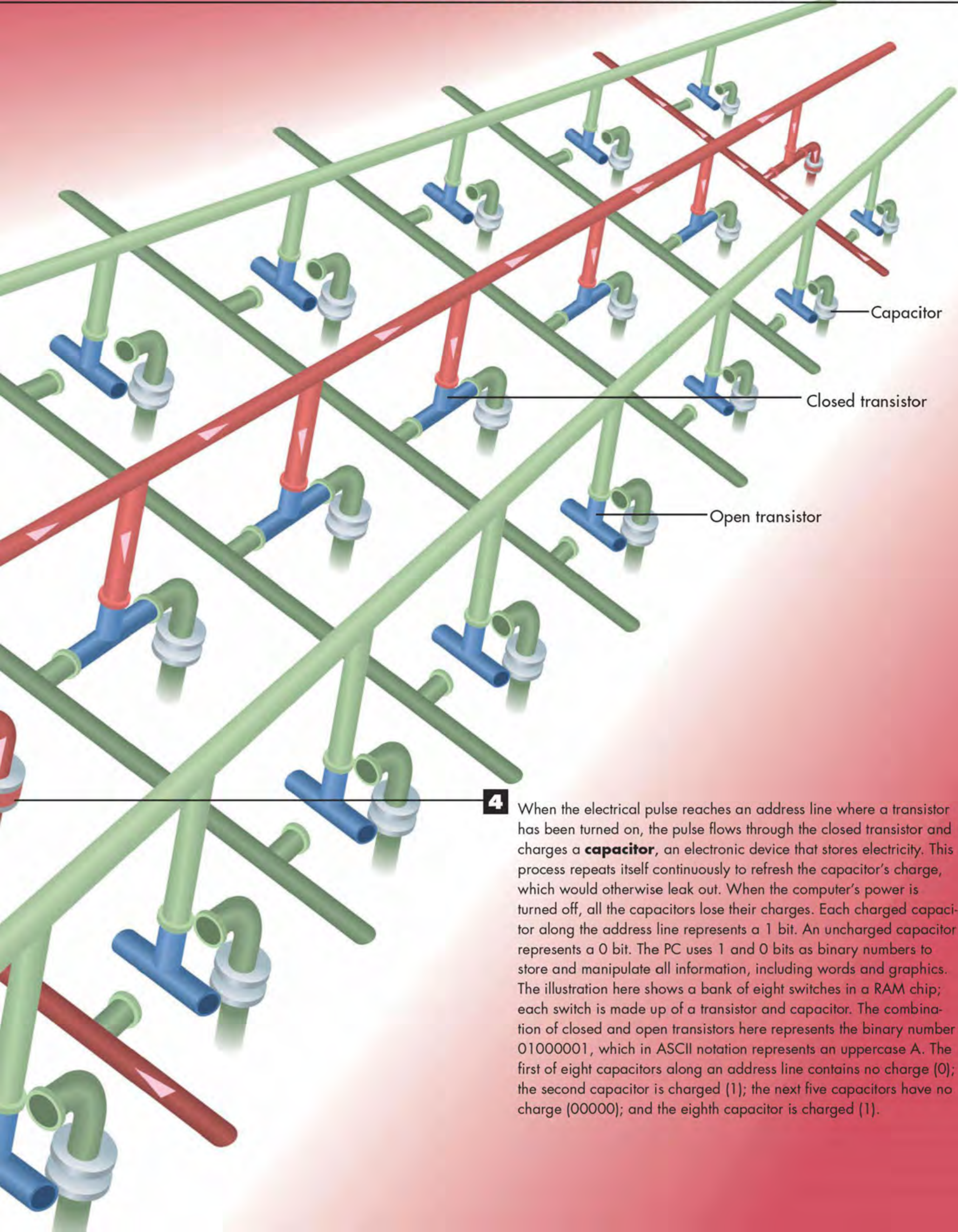
Writing Data to RAM

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

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



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

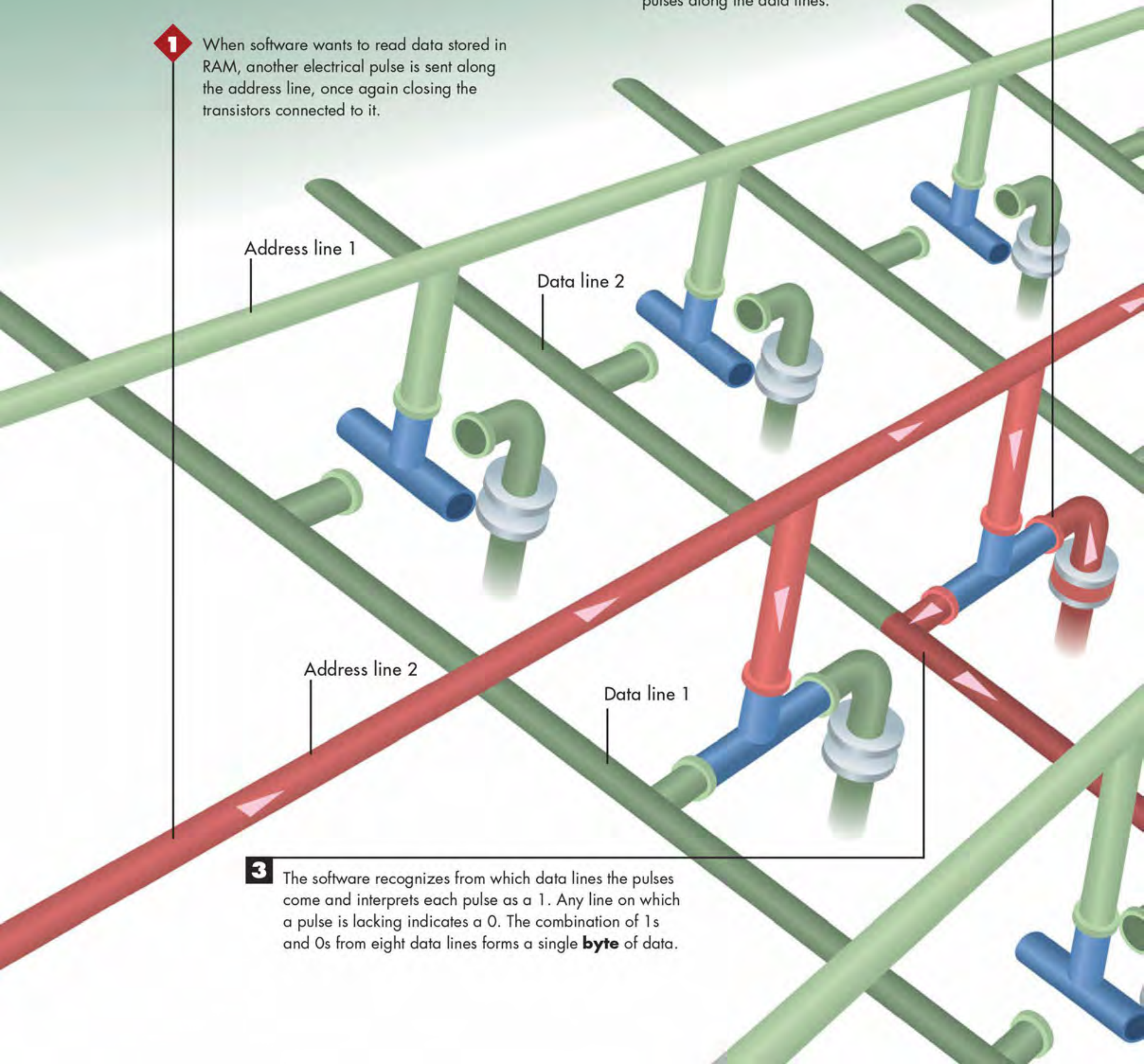


4 When the electrical pulse reaches an address line where a transistor has been turned on, the pulse flows through the closed transistor and charges a **capacitor**, an electronic device that stores electricity. This process repeats itself continuously to refresh the capacitor's charge, which would otherwise leak out. When the computer's power is turned off, all the capacitors lose their charges. Each charged capacitor along the address line represents a 1 bit. An uncharged capacitor represents a 0 bit. The PC uses 1 and 0 bits as binary numbers to store and manipulate all information, including words and graphics. The illustration here shows a bank of eight switches in a RAM chip; each switch is made up of a transistor and capacitor. The combination of closed and open transistors here represents the binary number 01000001, which in ASCII notation represents an uppercase A. The first of eight capacitors along an address line contains no charge (0); the second capacitor is charged (1); the next five capacitors have no charge (00000); and the eighth capacitor is charged (1).

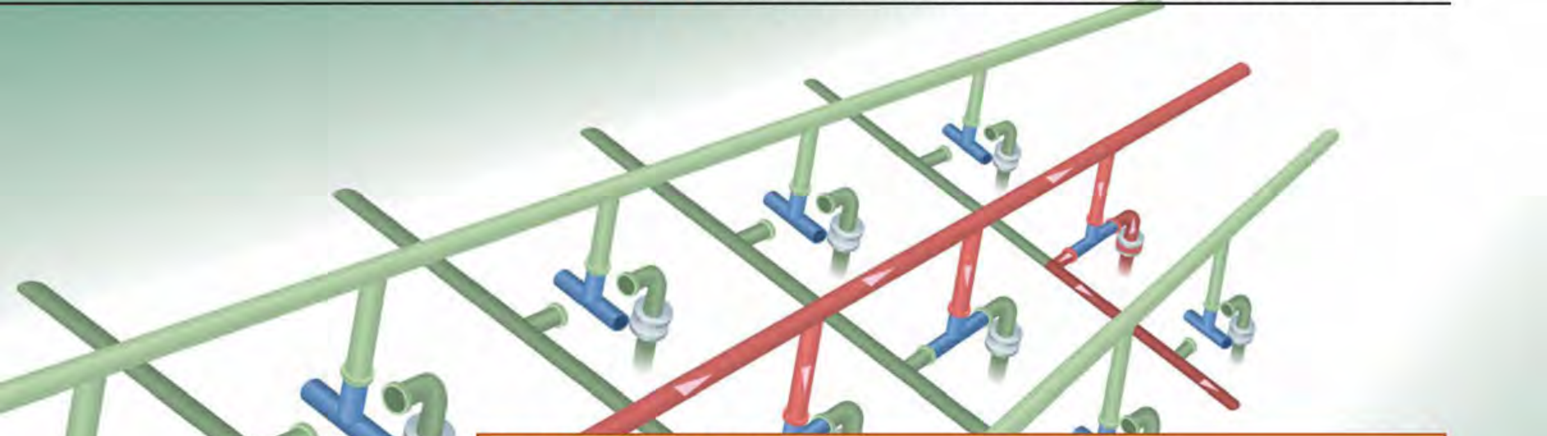
Reading Data from RAM

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

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



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

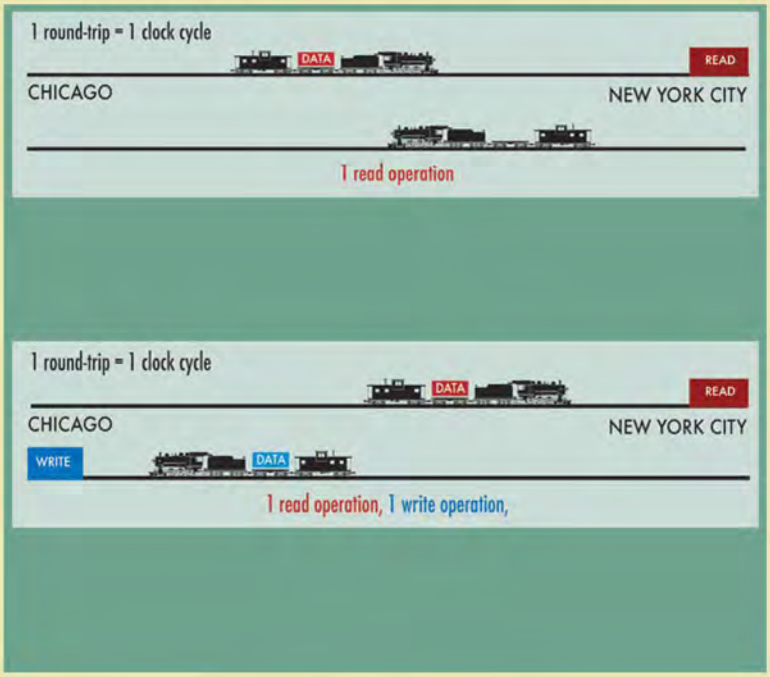


How DDR2 RAM Doubles Times

The fastest processors are limited by how fast memory feeds them data. Traditionally, the way to pump out more data was to increase the clock speed. With each cycle, or tick, of the clock regulating operations in the processor and movement of memory data, SDRAM memory—the kind illustrated here—could store a value or move a value out and onto the data bus headed to the processor. But the speeds of processors outstripped that of RAM. Memory caught up to processors two ways.

One is **double data rate (DDR)**. Previously, a bit was written or read on each cycle of the clock. It's as if someone loaded cargo (writing data) onto a train traveling from Chicago to New York, unloaded that cargo (reading), and then had to send the empty train back to Chicago again, despite having fresh cargo in New York that could hitch along for the return trip. With DDR, a handler could unload that same cargo when the train arrives in New York and then load it back up with new cargo again before the train makes its journey back to Chicago. This way, the train is handling twice as much traffic (data) in the same amount of time. Substitute *memory controller* for the persons loading and unloading cargo and *clock cycle* for each round-trip of the train, and you have DDR.

The other method is **dual channel architecture**—the 2 in DDR2. With double data rates alone, there are times when no data is ready to be stored or read from memory locations. It's as if the train reached one end of its line and the handler there hadn't found any cargo to put on the train. Dual channel adds another pipeline to supply memory to help ensure there is data for the train.


1 round-trip = 1 clock cycle
DATA
READ
CHICAGO
NEW YORK CITY
1 read operation
1 round-trip = 1 clock cycle
DATA
READ
CHICAGO
NEW YORK CITY
WRITE
DATA
1 read operation, 1 write operation,

How Memory Cards and Flash Drives Work

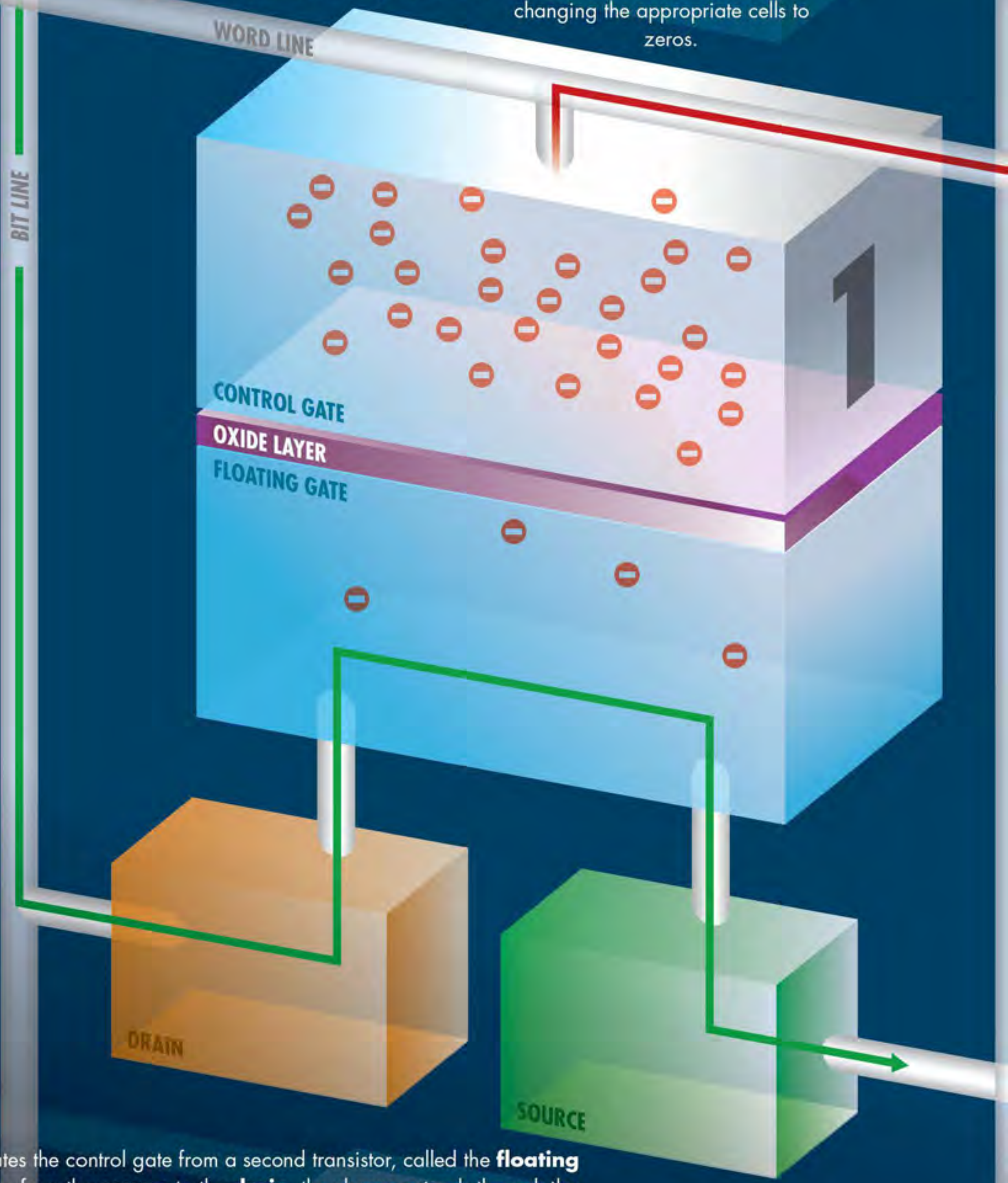
1 You know what happens when you turn off your PC—anything in RAM disappears. The next time you turn it on, the PC's memory is a blank slate, ready take on the form of the programs you run. **Flash memory** is different. When you turn off the computer, camera, phone, or MP3 player using flash memory, the documents, photos, numbers, and songs are still there when you turn it back on.

2 Flash memory is laid out along a grid of printed circuits running at right angles to each other. In one direction, the circuit traces are **word addresses**; circuits at a right angle to them represent the **bit addresses**. The two addresses combine to create a unique number address called a **cell**.

3 The cell contains two transistors that together determine if an intersection represents a 0 or a 1. One transistor—the **control gate**—is linked to one of the passing circuits called the **word line**, which determines the word address.

4 A thin layer of **metal oxide** separates the control gate from a second transistor, called the **floating gate**. When an electrical charge runs from the **source** to the **drain**, the charge extends through the floating gate, on through the metal oxide, and through the control gate to the word line.

5 A **bit sensor** on the word line compares the strength of the charge in the control gate to the strength of the charge on the floating gate. If the control voltage is at least half of the floating gate charge, the gate is said to be **open**, and the cell represents a 1. Flash memory is sold with all cells open. Recording to it consists of changing the appropriate cells to zeros.



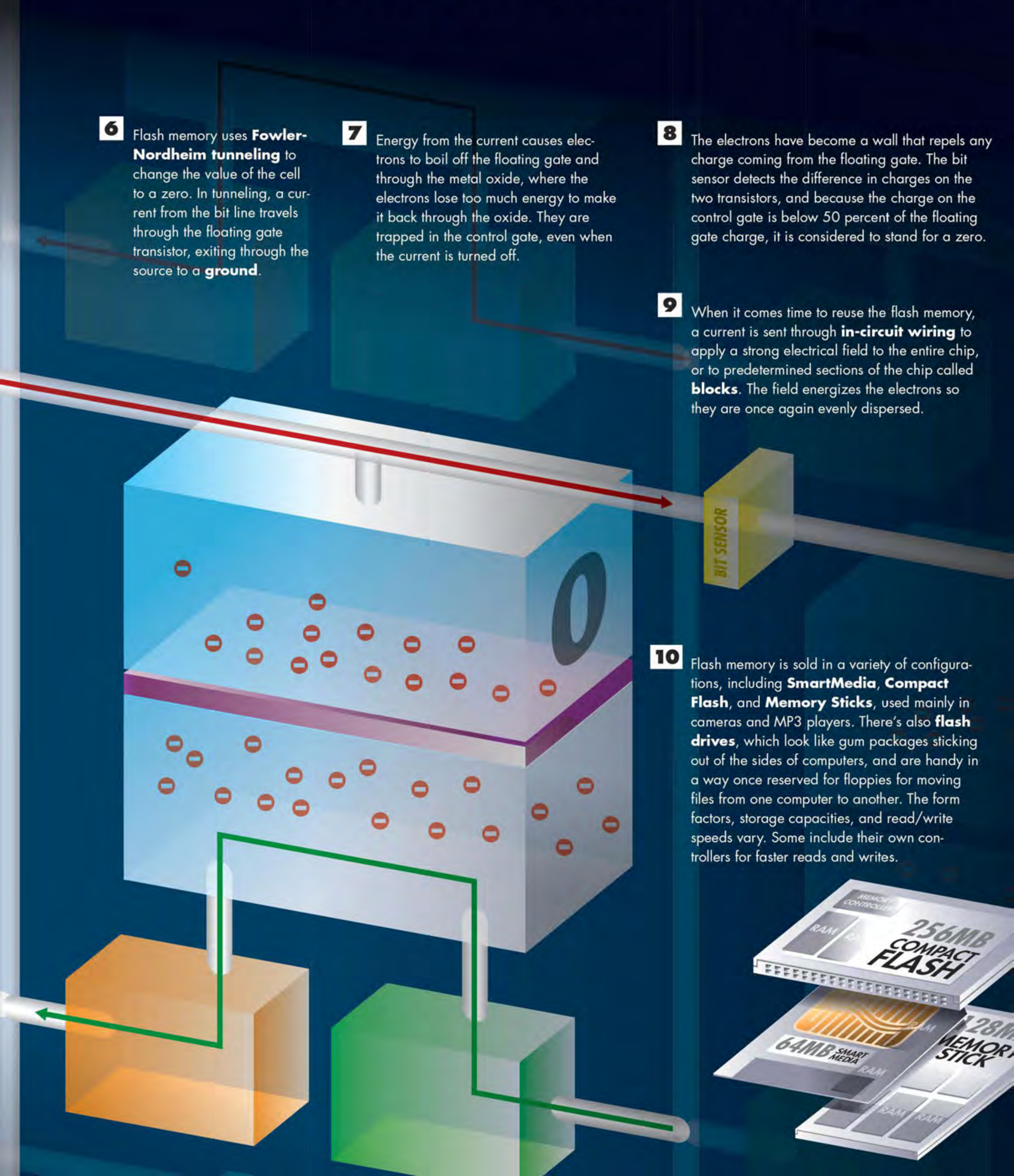
6 Flash memory uses **Fowler-Nordheim tunneling** to change the value of the cell to a zero. In tunneling, a current from the bit line travels through the floating gate transistor, exiting through the source to a **ground**.

7 Energy from the current causes electrons to boil off the floating gate and through the metal oxide, where the electrons lose too much energy to make it back through the oxide. They are trapped in the control gate, even when the current is turned off.

8 The electrons have become a wall that repels any charge coming from the floating gate. The bit sensor detects the difference in charges on the two transistors, and because the charge on the control gate is below 50 percent of the floating gate charge, it is considered to stand for a zero.

9 When it comes time to reuse the flash memory, a current is sent through **in-circuit wiring** to apply a strong electrical field to the entire chip, or to predetermined sections of the chip called **blocks**. The field energizes the electrons so they are once again evenly dispersed.

10 Flash memory is sold in a variety of configurations, including **SmartMedia**, **Compact Flash**, and **Memory Sticks**, used mainly in cameras and MP3 players. There's also **flash drives**, which look like gum packages sticking out of the sides of computers, and are handy in a way once reserved for floppies for moving files from one computer to another. The form factors, storage capacities, and read/write speeds vary. Some include their own controllers for faster reads and writes.



CHAPTER 6

How a Microprocessor Works

L1
I-Cache
BTB
BIU
Buffer
Uops
Decoders
Instructions
FETCH
DECODE
Tail
L1
D-Cache
ALU
DONE
Dispatch/Execute

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

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

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

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

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

The Brains

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

The first processor in an IBM PC was Intel's 8088 (the CPU itself was a follow-up to Intel's 8086). The generations of Intel processors that followed it were in the 80x86 family—80286, 80386, and 80486. All were more elaborate versions of the original 8088, but improved on its performance by one of two ways: operating faster or handling more data simultaneously. The 8088, for example, operated at 4.7MHz, or 4.7 million frequency waves a second; some 80486 chips go as fast as 133MHz. The 8088 could handle 8 bits of data at a time, and the 80486 handles 32 bits internally.

Intel and Advanced Micro Devices (AMD) are the only companies that make processors for Windows-based personal computers. The current standard for Intel processors is the Core 2 chip, the most recent being the Core 2 Quad. The combined chips cover less than a couple of square inches but hold more than 582 million **transistors**. All the operations of the Core 2 are performed by signals turning on or off different combinations of those switches. In computers, transistors are used to represent zeros and ones, the two numbers that make up the binary number system. These zeros and ones are commonly known as **bits**. Various groupings of these transistors make up the subcomponents within the Core 2, as well as those in coprocessors, memory chips, and other forms of digital silicon.

There are Core 2 processors designed to fill every market niche, from the bargain basement to the network server room. At the lowest end are Celeron processors with limited internal cache. They provide the function of the Pentium architecture with less speed. At the high end are Extreme Editions, which include large caches, and move data more quickly between it and the motherboard's chipset.

How the Processor Uses Registers

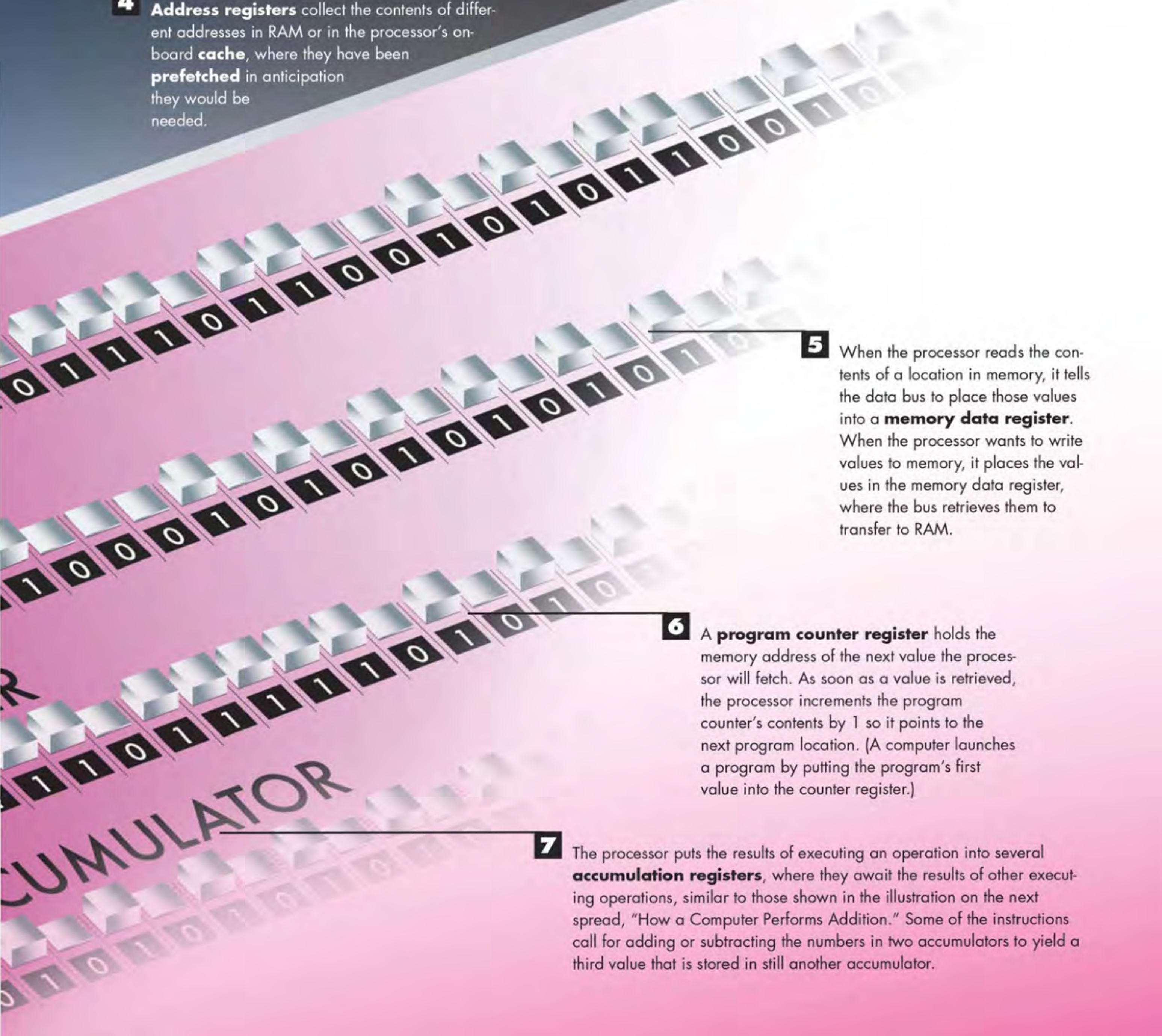
Few of us can do complex math in our heads. Even for something as simple as adding several rows of numbers, we need a pencil and paper to keep track of our operations on individual numbers. Microprocessors are not all that different in this regard. Although they are capable of performing intricate math 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.

ADRESSES
0 1 1 1 0 1 0 0 0 1
MDR
1 1 0 1 0 1 0
COUNTE
0 0 0 1
ACC
ALU
CONTROL UNIT

2 The processor's **control unit** directs the fetching and execution of program instructions. (See "How a Microprocessor Moves Data", on page **70**, for more information.) 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 bit. 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 in which registers the control unit stores them.



4 **Address registers** collect the contents of different addresses in RAM or in the processor's onboard **cache**, where they have been **prefetched** in anticipation 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.

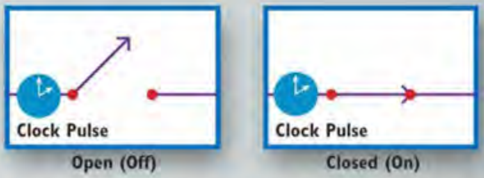
6 A **program counter register** holds the memory address of the next value the processor will fetch. As soon as a value is retrieved, the processor increments the program counter's contents by 1 so it points to the next program location. (A computer launches a program by putting the program's first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

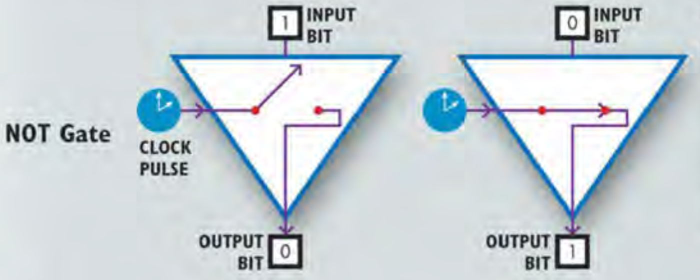
1 All information—words and graphics as well as numbers—is stored in and manipulated by a PC in the form of binary numbers. In the binary numerical system, there are only two digits—0 and 1. All numbers, words, and graphics are formed from different combinations of those digits.

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.

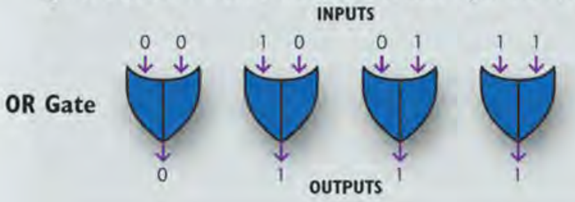
3 The simplest operation that can be performed with a transistor is called a NOT logic gate, made up of only a single transistor. This NOT gate is designed to take one *input* from the clock and one from another transistor. The NOT gate produces a single *output*—one that's always the opposite of the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

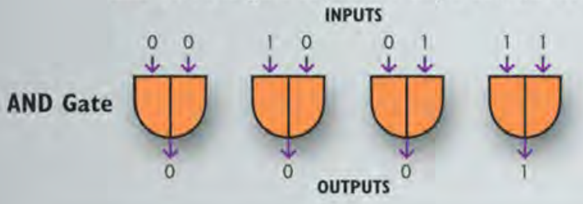
4 NOT gates strung together in different combinations create other logic gates, all of which have a line to receive pulses from the clock and two other input lines for pulses from other logic gates. The OR gates create a 1 if either the first or second input is a 1, and put out a 0 only if both inputs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



OR Gate Operations

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

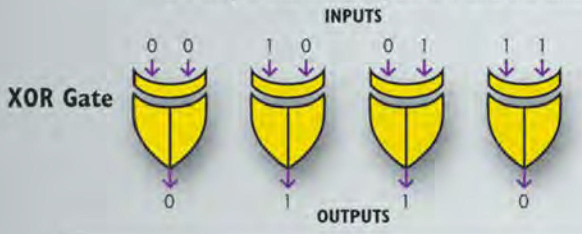
5 An AND gate outputs a 1 only if *both* the first and the second inputs are 1s.



AND Gate Operations

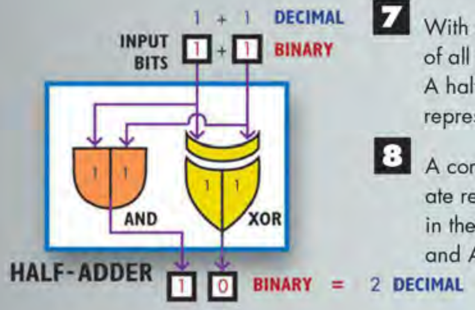
1ST INPUT	2ND INPUT	OUTPUT
0	0	0
1	0	0
0	1	0
1	1	1

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.



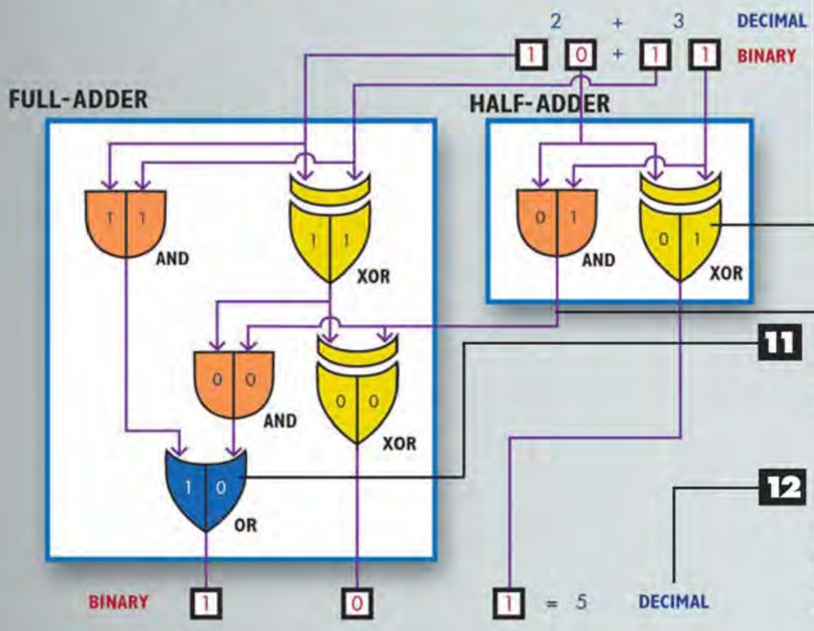
XOR Gate Operations

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. A full-adder consists of half-adders and other switches.

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



9 The result of the XOR operation (1) becomes the rightmost digit of the result.

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

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

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