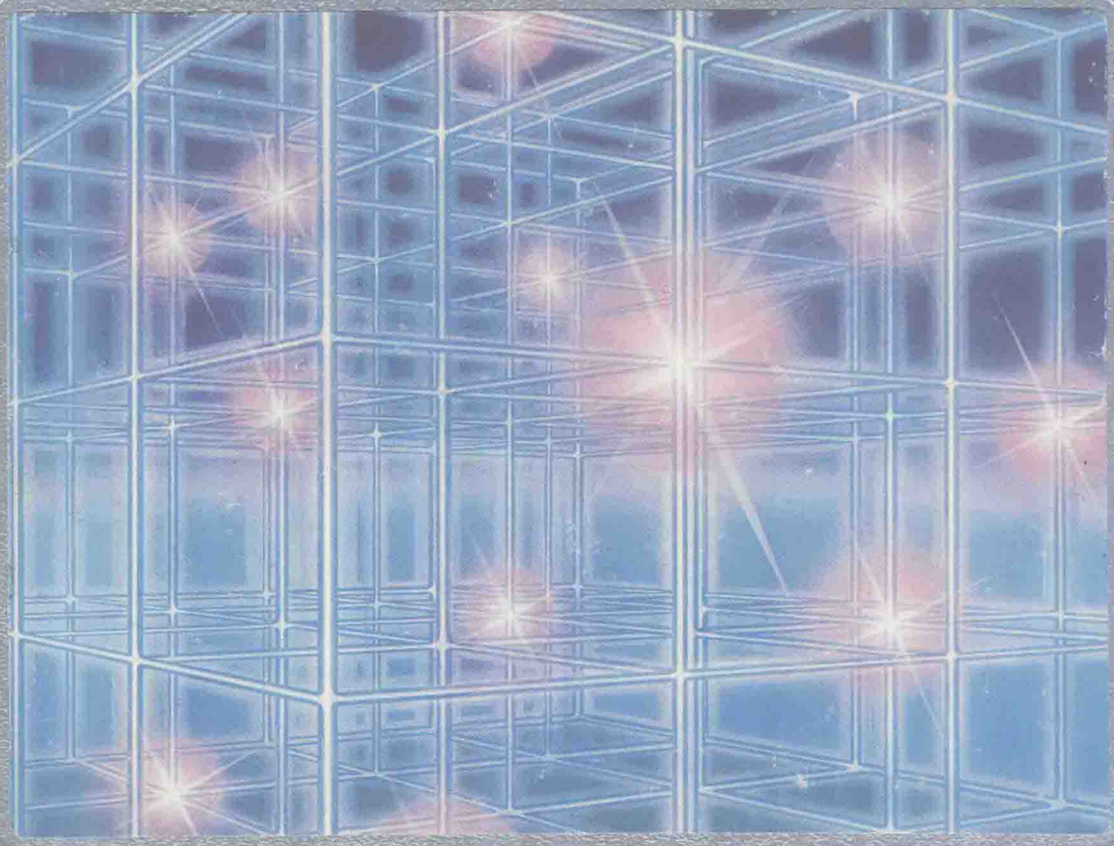


Memory and Storage

UNDERSTANDING COMPUTERS

Memory and Storage

BY THE EDITORS OF TIME-LIFE BOOKS
TIME-LIFE BOOKS, ALEXANDRIA, VIRGINIA

Memory and Storage

TIME
LIFE ®

Other Publications:
AMERICAN COUNTRY
VOYAGE THROUGH THE UNIVERSE
THE THIRD REICH
THE TIME-LIFE GARDENER'S GUIDE
MYSTERIES OF THE UNKNOWN
TIME FRAME
FIX IT YOURSELF
FITNESS, HEALTH & NUTRITION
SUCCESSFUL PARENTING
HEALTHY HOME COOKING
LIBRARY OF NATIONS
THE ENCHANTED WORLD
THE KODAK LIBRARY OF CREATIVE PHOTOGRAPHY
GREAT MEALS IN MINUTES
THE CIVIL WAR
PLANET EARTH
COLLECTOR'S LIBRARY OF THE CIVIL WAR
THE EPIC OF FLIGHT
THE GOOD COOK
WORLD WAR II
HOME REPAIR AND IMPROVEMENT
THE OLD WEST

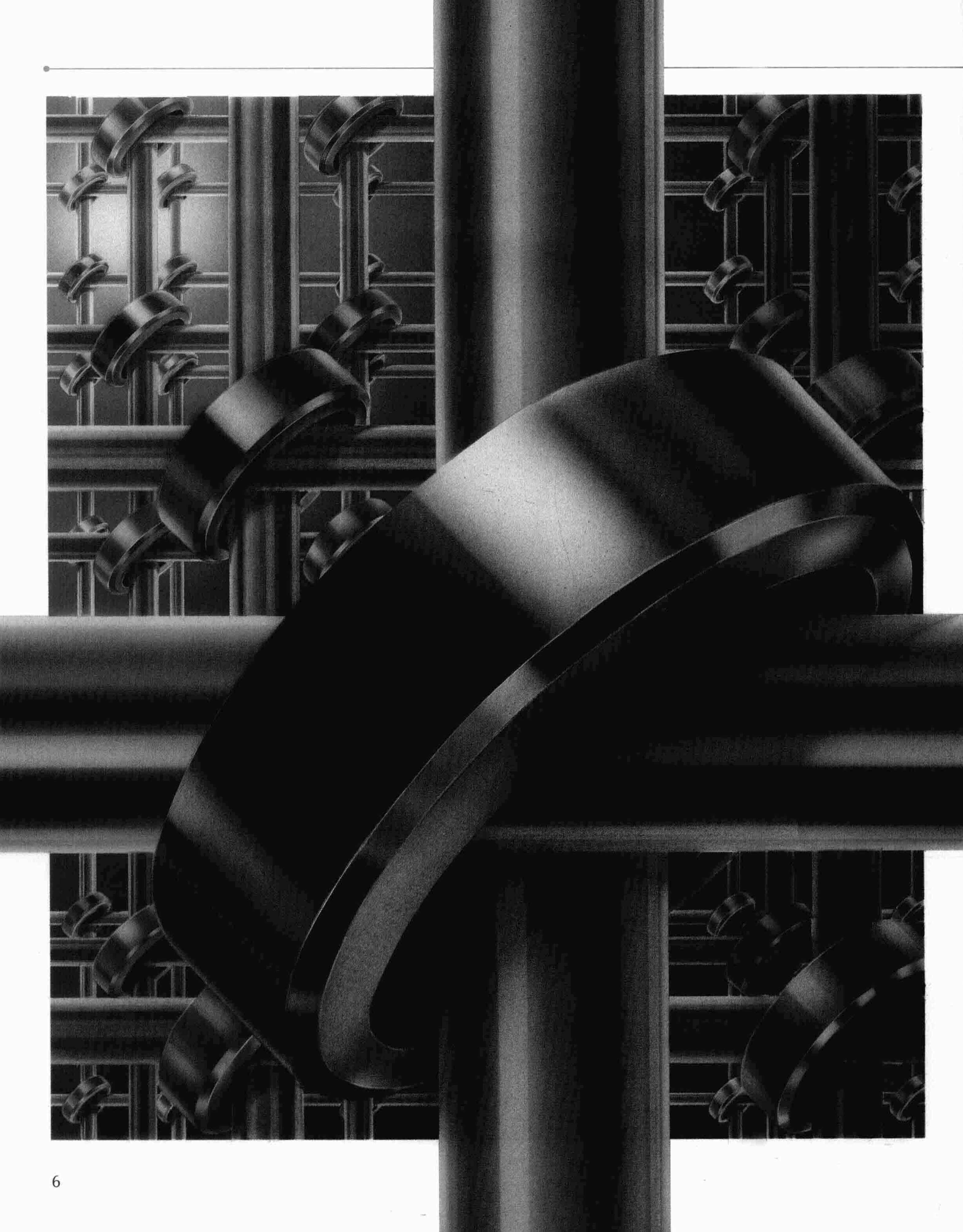
This volume is one of a series that examines various aspects of computer technology and the role computers play in modern life.

UNDERSTANDING COMPUTERS

Memory and Storage

BY THE EDITORS OF TIME-LIFE BOOKS
TIME-LIFE BOOKS, ALEXANDRIA, VIRGINIA

Contents



The Power of Recall

Next to the speed with which a computer can perform its basic functions—adding, subtracting, multiplying, dividing, and comparing one value with another—no measure of such a machine's potential is more revealing than its capacity for storing, and recalling on cue, information and the instructions for handling it. Without the ability to squirrel away data and programs, a computer would not deserve the name. Like the simplest electronic calculator, it could handle only two numbers and one operation at a time, an unacceptable limitation for any but the simplest of data-processing tasks.

Computers store information and programs by means that are, in essence, either electronic or mechanical. Electronic methods, generally called memory, are highly valued for their ability to keep pace with the computer's central processing unit, or CPU, which typically shuttles bits of information and program instructions in and out of memory several million times a second. Because these electronic circuits are expensive and because the most common varieties lose their contents when power to the computer is interrupted for even a split second, the role of memory is that of a temporary niche for data and instructions that must flow quickly in and out of the CPU.

The memories of computers that were constructed during the early part of the 1940s rarely exceeded a handful or two of bytes, a unit of measurement equal to eight bits, or binary digits. (In their formative years, computers often handled bits—the basic on-off currency of their circuits—one at a time. It is common nowadays for a computer to work with groups of bits—known as words—of two, four, and even eight bytes, a feature that by itself can increase the speed of a computer many times over.)

Memory grew slowly, taking until the mid-1960s to pass the megabyte (one-million-byte) mark, but the advent in the 1960s of integrated-circuit chips—microscopic assemblages of transistors and other electronic components—brought with it a tremendous increase in the memory capacity of computers. Today, even the most unpretentious desktop computer may contain a million bytes of memory and have the potential for controlling several million more. Some supercomputers have access to more than four gigabytes of memory—or four billion bytes.

Nonetheless, it is almost an axiom in the world of computers that there is never available a satisfactory amount of memory. Programs and data tend to grow, first to fill and then to exceed whatever memory a computer may command. To handle the overflow, mechanical methods are brought to bear. Commonly called storage, these devices once retained information in the form of holes punched in paper tape or cards. After decades of applied ingenuity, holes have been supplanted by infinitesimally small regions of magnetism on recording tape and related media or by microscopic, light-scattering bumps embossed in a compact disk—the same kind of silvery platter that has brought such an uncanny realism to the reproduction of music.

Storage devices for computers are mechanical in the sense that retrieving a parcel of magnetically or optically stored information requires the use of machinery such as tape recorders and disk players in order to locate the data and convert it into pulses of electricity acceptable to the computer's memory and CPU. A reversal of the process is necessary to find unused space for storing the information and to record it there.

As rapidly as these mechanical processes may occur, storage devices at their fastest are many times slower than the purely electronic circuits of memory. Yet in a wide range of applications, this limitation is more than compensated for by the huge capacity of the devices, available at a small fraction of memory's cost per word. A further advantage of storage is permanence. Data committed to magnetic tape, for example, or to a compact disk will be retained for many years. Such media are ideal for archiving records that may be vitally important, though rarely consulted.

Storage has always been a relatively plentiful commodity. As early as 1890, the census of the entire United States was represented as holes punched in cards about the size of a dollar bill, and there was one for each of the nation's inhabitants. Had anyone wished to attempt the feat, details about the entire world's population could have been recorded in the same manner. There was virtually no limit to the amount of storage available.

The tabulating and sorting machines that were used to tally responses to that *fin de siècle* survey of the nation had no memory at all, unless the dials that showed totals could be classed as such. More than five decades later, computer scientists striving to develop a satisfactory memory would invent curious strains of electronic exotica: glass tubes filled with mercury, television-like devices flecked with luminous blips that represented data, and tiny doughnuts, or cores, of an easily magnetized iron alloy suspended at the intersections in a grid of fine wires. The last, called magnetic core memory, would find the widest use. Its ascendancy would spark a fierce battle to patent and market the technology and would pit the emerging computer giant, IBM, against three men who claimed credit for the invention.

HOBBLED BY WIRES AND SWITCHES

Perfecting an economical method for expanding memory was a matter of considerable urgency to pioneer computer scientists, as demonstrated by ENIAC, the first electronic digital computer. Assembled at the University of Pennsylvania's Moore School of Electrical Engineering between 1943 and 1945, ENIAC's purpose was to calculate trajectories of artillery shells. From this data were constructed aiming tables for use by American artillerymen. To program the machine, an operator had to set hundreds of switches and link various parts of the computer with a spaghettilike skein of cables, a chore that might take days to complete. This wires-and-switches approach to telling the computer what to do was so unwieldy that it discouraged the machine's use for solving a wider variety of problems.

Other computers of that era replaced ENIAC's wires and switches with programming obstacles of a different kind. Their programs were stored as holes punched in a paper tape, its ends pasted together to form a closed loop. A tape reader, which was connected to the computer, translated the holes into instructions for the CPU. Compared with the method that used wires and switches, programs that were stored on tape speeded the adaptation of a computer from one purpose to another; revising the machine's marching orders required nothing more complicated than changing tapes. But there was a cost. Access time, the interval needed for a computer to read an instruction from the tape, was so long that the CPU often spent more time waiting for commands than it did executing them. Part of the problem lay in the slow speed at which tape readers operated. Perhaps even more important was the fact that access to the instructions usually was serial in nature. After executing a command, a computer could move only to the next instruction on the tape, even when some other operation might be more appropriate for conditions that arose during the computations. No provision was made for random, or direct, access to the program—that is, for skipping directly to the next useful instruction.

From these experiences emerged a concept of memory that endures to this day. First, it should be erasable—one set of data or programs stored there should be easy to replace with another. It should be reliable, not inclined to erase or rearrange itself. Fast access was just as crucial; memory must not delay the computer's CPU. Furthermore, memory should be inexpensive, something that any computer could have plenty of.

As World War II drew to a close, a computer memory that satisfied these requirements remained elusive. The few half-promising candidates would all suffer, in one way or another, from overlong access times. An example was the electromechanical relay. In essence, this was a device consisting of a small iron bar that was wound with a coil of wire. When current flowed through the wire, it magnetized the bar, which then snapped open a spring-loaded switch. When the current was shut off, the iron bar was demagnetized and the spring snapped the switch closed. The relay's alternate positions—open and closed—could symbolize ones and zeros, the binary vocabulary of a computer. Though substantially quicker than paper tape, a relay's mechanical switch was still much slower than the electronic switches that made up a CPU, so this approach to memory was soon abandoned.

INTENTIONAL DELAYS

The next significant advance in computer memory was an American brainchild, the acoustic delay line. Developed at Bell Labs in the early 1940s by William Shockley, later to share a Nobel Prize for the invention of the transistor, the acoustic delay line stored binary numbers as sound waves. Numbers representing either data or instructions to the computer were held in the delay line until needed for processing by the CPU.

Typically, a delay line consisted of a glass tube, filled with mercury and plugged at each end with a quartz crystal. Electrical pulses representing bits of data or program instructions to be stored in the memory were fed to the crystal at one end of the tube, causing the quartz to vibrate in response. The resulting sound waves advanced through the mercury to the quartz crystal at the other end of the tube, causing it to vibrate in turn and reconvert the data into electrical pulses, which could be rerouted indefinitely through the mercury until the computer required them for processing.

As sound, the bits passed through the mercury 200,000 times more slowly than they zipped along wires as pulses of electricity. Thus, a tube a couple of yards long delayed—or stored—the bit stream for about one thousandth of a second, a lengthy period compared with the speed of a CPU, which could execute a thousand instructions during the same interval.

Although access to data stored in an acoustic delay line was speedier than what had been provided by the electromechanical relays it replaced, this kind of memory exhibited certain objectionable traits. For example, the bit stream's trip through the mercury could not be interrupted. Once the sound waves had begun their passage, the data that they represented could be retrieved only after it had crawled the full length of the tube. Moreover, the very nature of the delay line made it a serial device: There was simply no possibility of random access to data or programs stored there.

A CLIMATE-CONTROLLED MEMORY

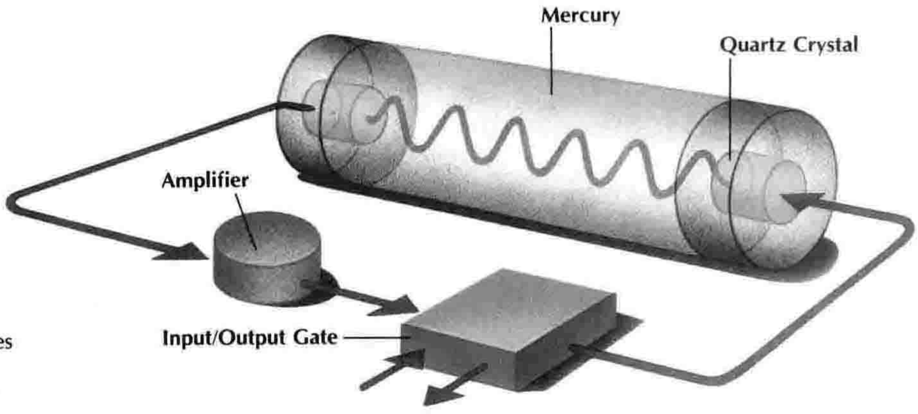
Delay lines were also sensitive to the weather. Because the speed of sound in mercury is higher at cool temperatures than at warm ones, sound waves traveling through a delay line tended to accelerate or slow down in response to fluctuations in the ambient temperature. A change as small as one degree caused the computer, which expected data to exit from a tube punctually and at a prescribed rate, to misread the information. The result was loss or misinterpretation of data and program instructions.

Engineers attempted to solve the problem by isolating the delay line in a warm oven to stabilize the temperature of the mercury. Doing so, however, required an exasperatingly long warm-up period each time the computer was turned on. Other, less temperature-sensitive fluids—and even solid quartz—were tested as substitutes for mercury. The British computing pioneer Alan Turing suggested that, based on his theoretical studies of alcohol-and-water mixtures, the ideal delay-line medium would be gin. Unfortunately, the lower density of such materials reduced their ability to retard sound waves, so they saw only the most limited use for computer memory.

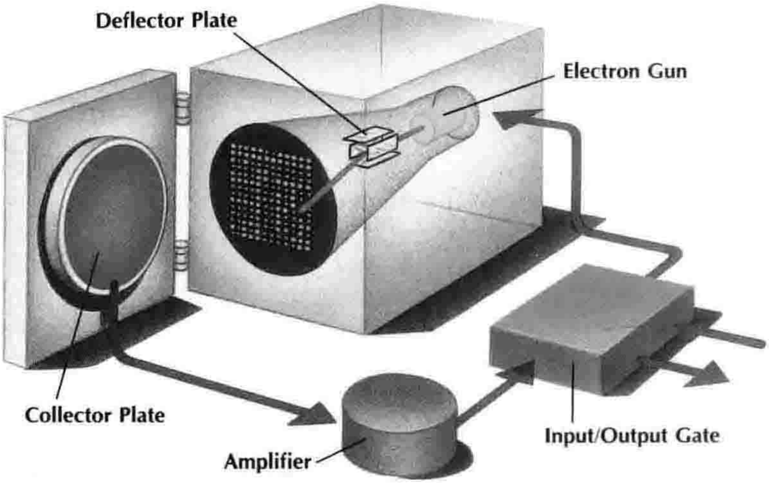
It fell to Herman Lukoff, a twenty-three-year-old research engineer at the University of Pennsylvania's Moore School, to overcome mercury's temperature-related handicap. For a computer called EDVAC, designed in 1945 by a team of engineers at the school, Lukoff devised a special package of electronics called a temperature compensator. His invention expanded or condensed the spacing between pulses in the bit stream en route to the delay line according to the temperature of the mercury in the tube. Doing so guaranteed that the sound wave associated with each pulse of data arrived at the end of the tube on schedule, regardless of how warm or cool the mercury might become.

By March 1947 Lukoff was ready to display his system at a convention sponsored by the Institute of Radio Engineers in New York City. No sooner had Lukoff begun his demonstration, however, than the mercury exhibited extraneous pulses that garbled the data stored there. The mystery pulses continued to course through the delay line every two seconds until Lukoff spotted the source of the trouble—a rotating radar antenna in the exhibit booth of the Army Signal Corps on the far side of the convention hall. The radar beam had struck the delay line with each turn of the antenna, triggering random pulses in the memory's circuitry. The army obligingly turned off the radar beam, and the new memory

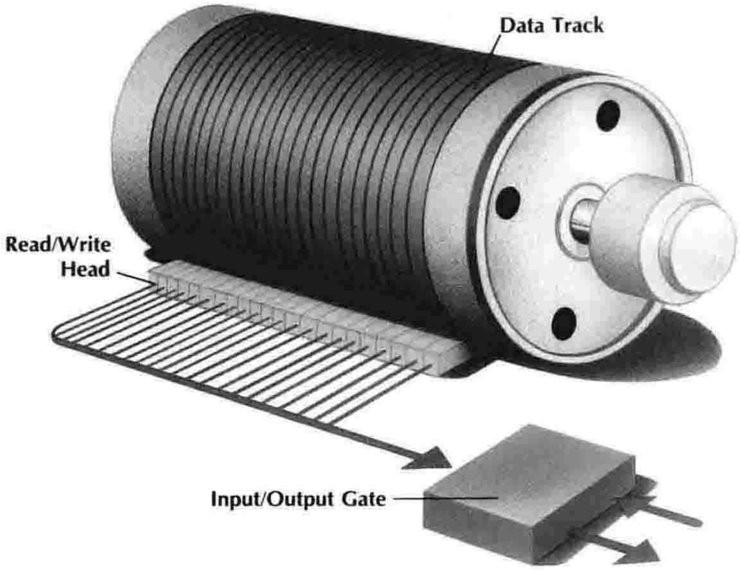
Early Memory Workhorses



Acoustic delay line. This system, the first computer memory to see widespread use, was invented in the late 1940s. Electrical pulses, representing binary digits from the computer's central processing unit (CPU), are fed through an input/output gate to a quartz crystal, causing it to vibrate. The crystal's oscillations convert the pulses into sound waves that move along a column of mercury to another quartz crystal, which transmutes them into electrical pulses. After amplification, the data can be returned to the CPU for processing or sent on another circuit of the delay line.



Williams tube. In this scheme, a cathode-ray tube (CRT) uses electrostatic charges to store binary data. A burst of electrons, emitted by a gun at the rear of the tube, is directed by computer-controlled deflector plates to predetermined spots on the CRT screen, where the burst leaves either a charged dot or dash. Because the charge dissipates quickly, the beam must re-create the dots and dashes five times each second. To retrieve data from memory, a metallic screen called a collector plate, positioned against the face of the tube, converts dots and dashes into electric pulses. They are amplified and returned through an input/output gate to the computer.



Magnetic drum. This device, invented in the early 1950s, uses the magnetic properties of electric pulses *(pages 40-42)* to store data on the polished metal surface of a rapidly rotating cylinder. A series of read/write heads, each corresponding to a data track around the circumference of the drum, records and recovers the data, which passes between computer and drum through an input/output gate. Drums were so slow that they rarely saw duty as computer memory. Instead, they were used as storage devices until supplanted in the 1960s by modern, high-performance hard-disk drives *(pages 60-61).*

performed flawlessly—but to no great end. Much to Lukoff's disappointment, the EDVAC team turned thumbs down on his invention, choosing instead the well-proven oven method of temperature control.

MEMORY GOES TUBULAR

While Lukoff struggled to perfect the serial delay line, Fred Williams, a researcher at England's Manchester University, had begun crafting another apparatus, also serial in nature, that would give rise to the first random-access computer memory. Built by Williams in 1946, the device exploited the workings of the cathode-ray tube, or CRT, which is best known today as the display screen of televisions and personal computers.

Williams knew that at every point where the electron beam of a CRT struck the phosphor-coated interior of the tube, a charged "dot" was created that would linger there briefly before it faded away. If the beam was moved slightly instead of being kept still, it left a charged "dash." By depositing such dots and dashes on the tube in a series of horizontal sweeps, the electron beam could be used to write, or store, the ones and zeros of binary information. The recording method was serial because the beam wrote the binary digits of a computer word one after another in rows on the cathode-ray tube.

The electron beam functioned as a reading mechanism, too. At low power (in order not to alter the previously recorded information), the beam swept back and forth across the charged spots inside the tube, causing a current to leak through the glass at those points to the front of the tube. The strength of the current differed slightly for a dot or a dash, enough for a metallic screen fitted over the face of the tube to distinguish between ones and zeros and then pass them to a computer's CPU.

There was one difficulty still remaining: The dots and dashes faded from the tube just two-tenths of a second after they appeared. If Williams's new kind of memory was to be of any consequence, he would have to find a way to correct this spontaneous self-erasure. Williams discovered that the writing process left a sprinkling of electrons surrounding each dot and dash on the tube. He devised a system in which the low-power beam constantly scanned the face of a tube on which data had been recorded. Whenever an electron smudge signaled the proximity of a dot or dash, a distinctive current was generated in the metallic screen. Instantly, the power of the beam was boosted to rewrite the information in the same location.

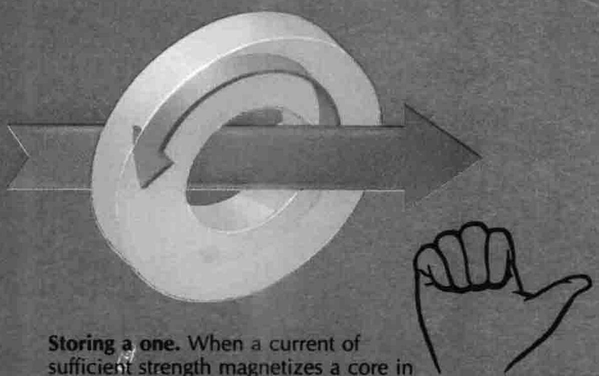
By the end of 1947, Williams and a colleague of his, Tom Kilburn, had produced a prototype Williams tube—at some point the invention had acquired the name of the inventor—that was capable of storing 1,000 bits of information. Only six months later, the two unveiled a similar device boasting a capacity of several thousand bits and access to data, though still serial, ten times faster than that offered by a delay line.

During the spring of 1948, word of the breakthrough that Fred Williams had achieved reached the Institute for Advanced Study in Princeton, New Jersey. There a team of engineers working under the leadership of John von Neumann had reached an impasse in their effort to construct a new general-purpose scientific machine that was to be called the IAS computer. In the course of the preceding two years the von Neumann team had considered a number of

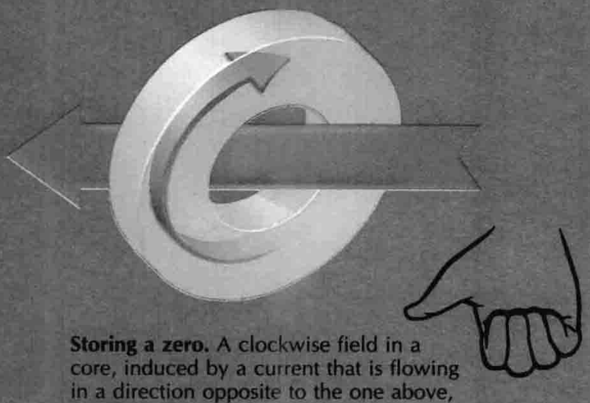
Magnetic Memory for Computers

Core memory takes advantage of the intimate relationship between electricity and magnetism to provide total recall for computers. A current in a wire creates a magnetic field around it. The orientation of the magnetism is determined by the current flow, according to what physicists call the right-hand rule: With the thumb pointed in the direction of the current, the fingers of the right hand curl in the direction of the magnetic field. This field can permanently magnetize a doughnut-shaped core made of ferromagnetic materials. Magnetizing the core in one direction—counterclockwise as shown on this and the following pages—stores a one. Reversing the current magnetizes the core in the other direction, storing a zero.

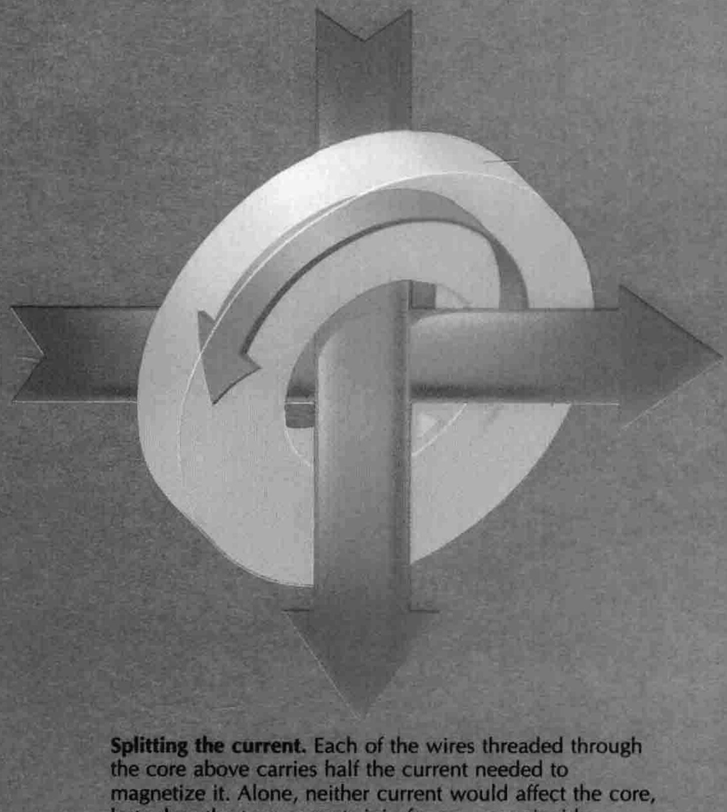
For a core to be magnetized, the current must exceed the magnetic threshold of the ferromagnetic material in the doughnut. Yet the current need not be supplied by a single wire. Two wires, for example, can each supply half the current. Furthermore, the wires need not be parallel; as long as current flows through the wires in the direction indicated by the right-hand rule, they may even cross at right angles to each other *(below)*. The concept of coincident currents—two currents, each contributing half the field necessary to magnetize a core—makes this type of memory practical *(overleaf)*.



Storing a one. When a current of sufficient strength magnetizes a core in a counterclockwise direction, according to the right-hand rule, the magnetic field in the core represents a one.



Storing a zero. A clockwise field in a core, induced by a current that is flowing in a direction opposite to the one above, stands for a zero.



Splitting the current. Each of the wires threaded through the core above carries half the current needed to magnetize it. Alone, neither current would affect the core, but when the two currents join forces—or coincide—the core is magnetized according to a double application of the right-hand rule, in this case as a one. Reversing the current in both wires flips the magnetic field of the core, marking it as a zero.

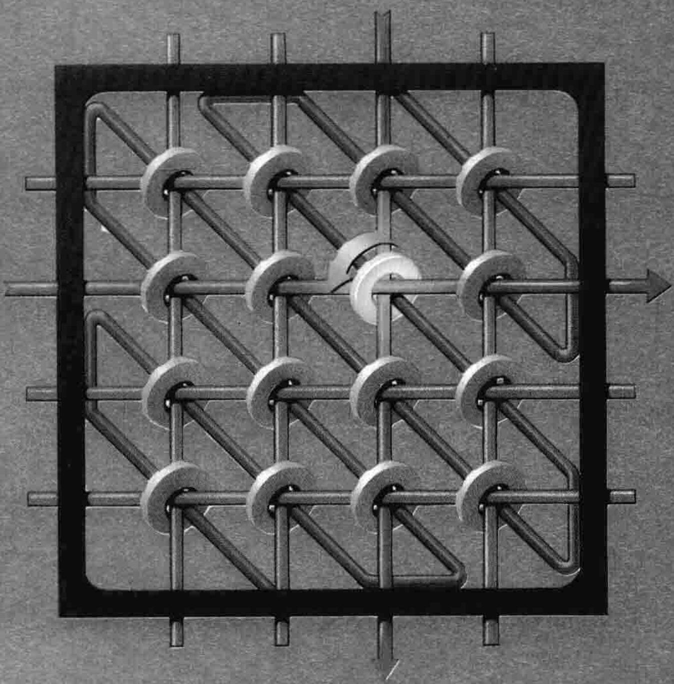
Storing Bits and Getting Them Back

The magnetization of cores by means of currents traveling through two wires simultaneously is the basis for turning a scientific curio into a potent computer memory. The simplest version is a two-dimensional grid. It consists of an insulating frame filled in with a weave of horizontal and vertical wires. At each intersection hangs a tiny ring of ferromagnetic material no more than 1/50 inch in diameter.

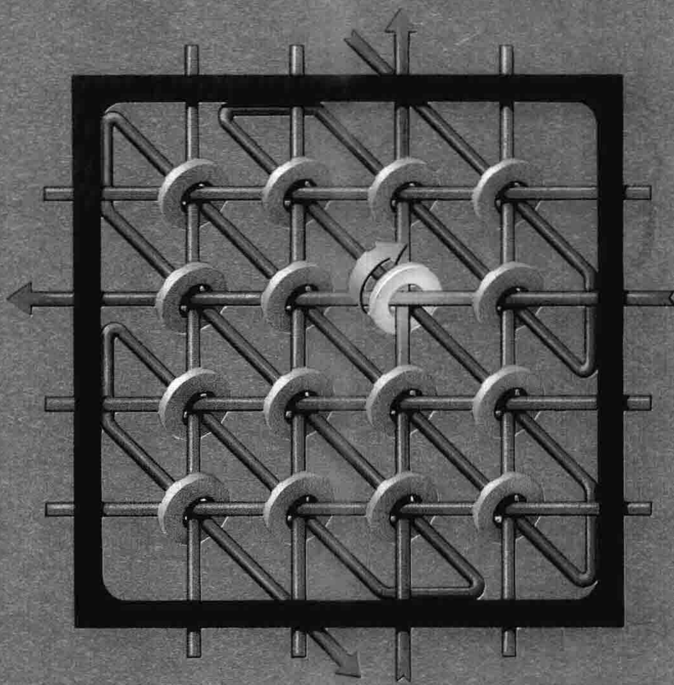
Any one of the rings in the grid can be selected individually to store a bit—either a one or a zero—simply by energizing the wires that pass through it *(below, left)*. Because the rings can be activated in any sequence imaginable, core memory provides true random access to data.

Recalling the contents of core memory combines the principle of coincident current with the fact that a changing magnetic field generates a current in a nearby wire. By being programmed to note the presence or absence of this current, a computer can discern whether a core contains a one or a zero *(below, right)*.

As a practical matter, the capacity of two-dimensional core memories is limited by their size. A memory that had a capacity of one million bits would require 1,000 vertical wires and 1,000 horizontal wires, each of which would have to have a control circuit within the computer. In order to reduce the number of these circuits, there are smaller two-dimensional panels of core memory lined up one behind the other in a three-dimensional array *(right)*. In this kind of arrangement, sixteen panels of 256 horizontal wires and 256 vertical wires each yield more than one million bits of storage; instead of there being more than 2,000 wires to control, however, there are fewer than 600.



Writing a bit. A panel of core memory consists of horizontal and vertical write wires, threaded through cores, that are used to store—or write—data bits into the memory by magnetizing individual cores. A sense wire *(blue)*, snaking diagonally through the panel, is used for reading the memory *(right)*. To store a bit at a core, the two wires passing through it are energized *(red)*, each with half the current necessary to magnetize the core in accordance with the right-hand rule. The core highlighted in this drawing is being used to store a one.



Reading the bit. To determine whether a core contains a one or a zero, the appropriate wires are energized with the currents necessary to magnetize the doughnut as a zero. Thus, a core storing a one has the magnetization reversed, the one becoming a zero. This event generates a voltage pulse in the sense wire. Detecting the pulse, the computer notes the content of the core as a one, then reverses the currents through the wires to erase the zero and restore the one. When the same technique is used to read a core containing a zero, the magnetization is unchanged, and the computer interprets the absence of a voltage pulse as the opposite of a one.

A multiple-panel memory. To store a bit in a three-dimensional core memory, simplified in this illustration by the exclusion of the sense wire used for reading, the write wires are arranged so that energizing any two of them can affect the magnetization of a single core in each panel. In this example, a core in the center panel is being magnetized as a zero, while a core in the rear panel becomes a one. The core in the front panel is unaffected; a current flowing through an inhibit wire *(green)* cancels the magnetizing effect of the current in the horizontal wire, leaving a field only half the strength required to magnetize the core.