

# COMPUTING

BY MARTIN CAMPBELL-KELLY

THE  
ORIGIN

## The information age began with the realization that machines could emulate the power of minds

In the standard story, the computer's evolution has been brisk and short. It starts with the giant machines warehoused in World War II-era laboratories. Microchips shrink them onto desktops, Moore's Law predicts how powerful they will become, and Microsoft capitalizes on the software. Eventually small, inexpensive devices appear that can trade stocks and beam video around the world. That is one way to approach the history of computing—the history of solid-state electronics in the past 60 years.

But computing existed long before the transistor. Ancient astronomers developed ways to predict the motion of the heavenly bodies. The Greeks deduced the shape and size of Earth. Taxes were summed; distances mapped. Always, though, computing was a human pursuit. It was arithmetic, a skill like reading or writing that helped a person make sense of the world.

The age of computing sprang from the abandonment of this limitation. Adding machines and cash registers came first, but equally critical was the quest to organize mathematical computations using what we now call “programs.” The idea of a program first arose in the 1830s, a century before what we traditionally think of as the birth of the computer. Later, the modern electronic computers that came out of World War II gave rise to the notion of the universal computer—a machine capable of any kind of information processing, even including the manipulation of its own programs. These are the computers that power our world today. Yet even as computer technology has matured to the point where it is omnipresent and seemingly limitless, researchers are attempting to use fresh insights from the mind, biological systems and quantum physics to build wholly new types of machines.

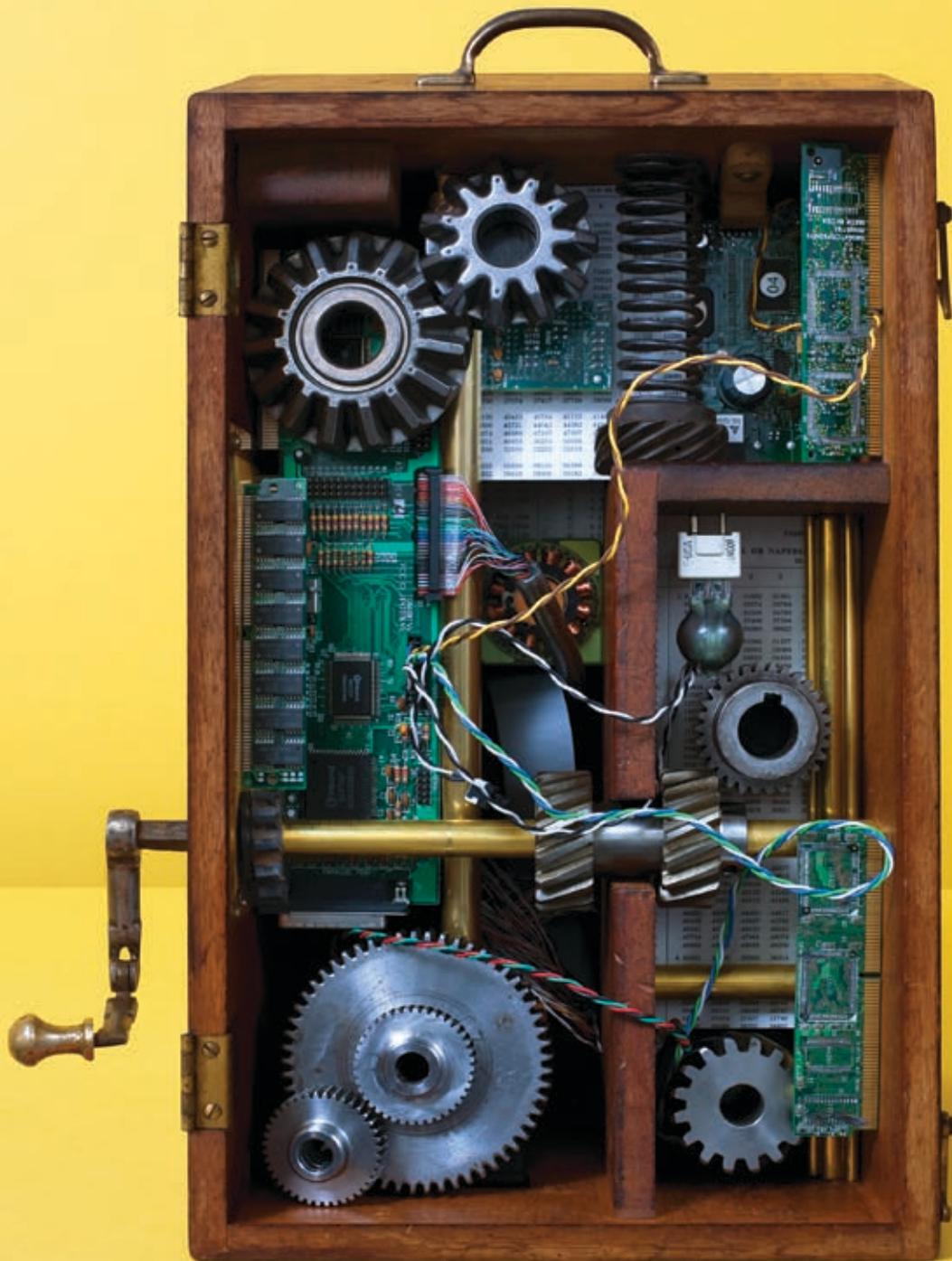
### The Difference Engine

In 1790, shortly after the start of the French Revolution, Napoleon Bonaparte decided that the republic required a new set of maps to establish a fair system of property taxation. He also ordered a switch

### KEY CONCEPTS

- The first “computers” were people—individuals and teams who would tediously compute sums by hand to fill in artillery tables.
- Inspired by the work of a computing team in revolutionary France, Charles Babbage, a British mathematician, created the first mechanical device that could organize calculations.
- The first modern computers arrived in the 1950s, as researchers created machines that could use the result of their calculations to alter their operating instructions.

HOLLY LUNDEM (photo/illustration); GENE BURKHARDT (styling)

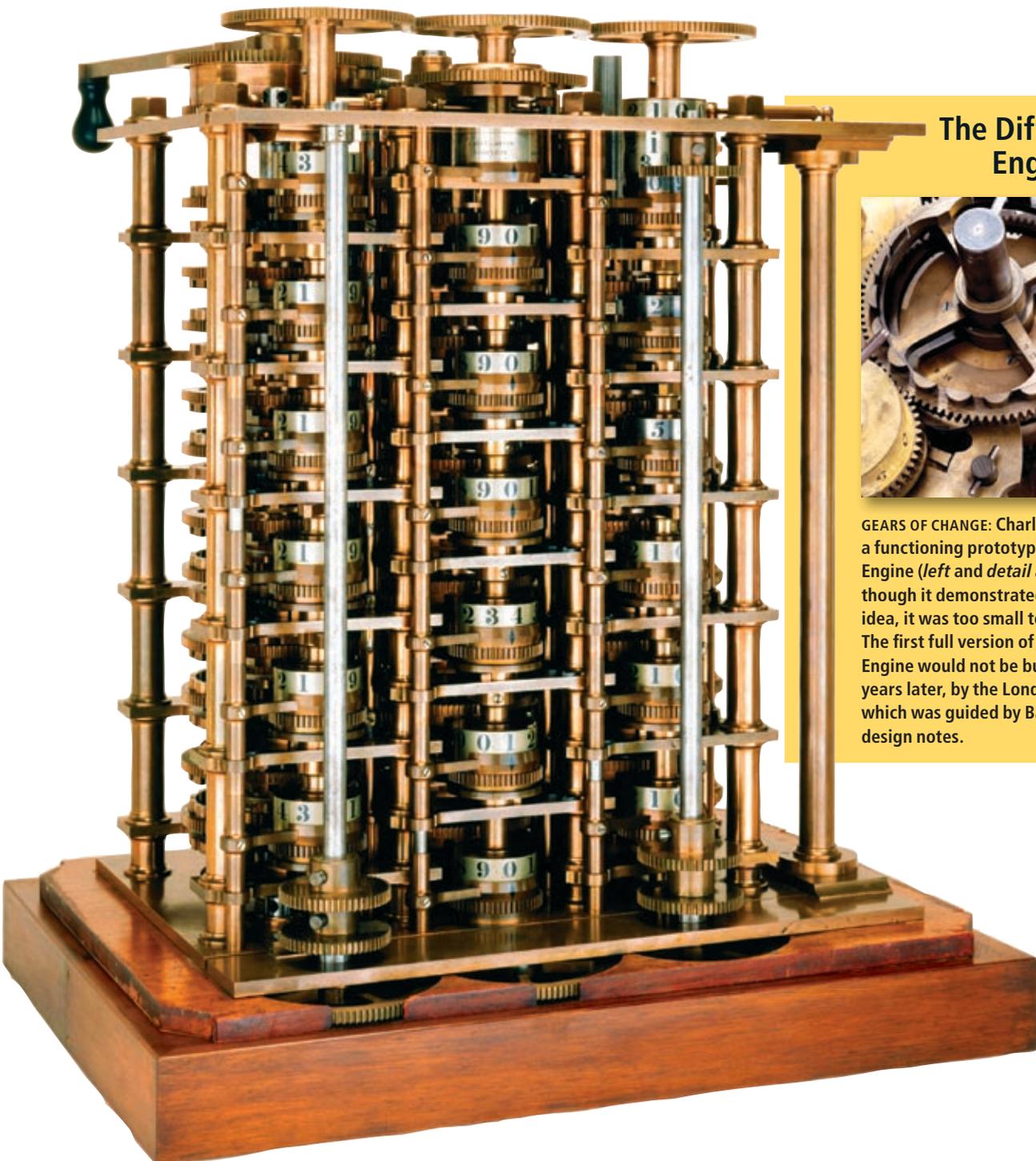


from the old imperial system of measurements to the new metric system. To aid the engineers and mathematicians making the change, the French ordinance survey office commissioned a fresh set of mathematical tables.

In the 18th century, however, computations were done by hand. A “factory floor” of between 60 and 80 human computers added and subtracted numbers to fill in line after line of the tables for the survey’s *Tables du Cadastre* proj-

ect. It was grunt work, demanding no special skills above basic numeracy and literacy. In fact, most computers were hairdressers who had lost their jobs—aristocratic hairstyles being the sort of thing that could endanger one’s neck in revolutionary France.

The project took about 10 years to complete, but by then the war-torn republic did not have the funds necessary to publish the work. The manuscript languished in the *Académie des Sciences*



## The Difference Engine



**GEARS OF CHANGE:** Charles Babbage produced a functioning prototype of his Difference Engine (left and detail above) in 1832. Although it demonstrated the feasibility of his idea, it was too small to be of practical use. The first full version of a working Difference Engine would not be built until 1991, 159 years later, by the London Science Museum, which was guided by Babbage’s detailed design notes.

for decades. Then, in 1819, a young British mathematician named Charles Babbage would view it on a visit to Paris. Babbage was 28 at the time; three years earlier he had been elected to the Royal Society, the most prominent scientific organization in Britain. He was also very knowledgeable about the world of human computers—at various times he personally supervised the construction of astronomical and actuarial tables.

On his return to England, Babbage decided he would replicate the French project not with human computers but with machinery. England at the time was in the throes of the Industrial Revolution. Jobs that had been done by human or animal labor were falling to the efficiency of the machine. Babbage saw the power of mechanization and realized that it could replace not just muscle but the work of minds.

He proposed the construction of his Calculating Engine in 1822 and secured government funding in 1824. For the next decade he immersed himself in the world of manufacturing, seeking the best technologies with which to construct his engine.

The year 1832 was Babbage's *annus mirabilis*. That year he not only produced a functioning model of his calculating machine (which he called the Difference Engine) but also published his classic *Economy of Machinery and Manufactures*, establishing his reputation as the world's leading industrial economist. He held Saturday evening soirees at his home in Dorset Street in London, which were attended by the front rank of society. At these gatherings the model Difference Engine was placed on display as a conversation piece.

A year later Babbage abandoned the Difference Engine for a grander vision that he called the Analytical Engine. Whereas the Difference Engine had been limited to the single task of table making, the Analytical Engine would be capable of any mathematical calculation. Like a modern computer, it would have a processor that performed arithmetic (the "mill"), memory to hold numbers (the "store"), and the ability to alter its function via user input, in this case by punched cards. In short, it was a computer conceived in Victorian technology.

Babbage's decision to abandon the unfinished Difference Engine was not well received, however, and the government demurred to supply him with additional funds. Undeterred, he produced thousands of pages of detailed notes and machine drawings in the hope that the government would one day fund construction. It was

not until the 1970s, well into the computer age, that scholars studied these papers for the first time. The Analytical Engine was, as one of those scholars remarked, almost like looking at a computer designed on another planet.

### The Dark Ages

Babbage's vision, in essence, was digital computing. Like today's devices, such machines manipulate numbers (or digits) according to a set of instructions and produce a precise numerical result.

Yet after Babbage's failure, computation entered what English mathematician L. J. Comrie called the Dark Age of digital computing—a period that lasted into World War II. During this time, machine computation was done primarily with so-called analog computers. These devices model a system using a mechanical analog. Suppose, for example, one wanted to predict the time of a solar eclipse. To do this digitally, one would numerically solve Kepler's laws of motion. Before digital computers, the only practical way to do this was hand computation by human computers. (From the 1890s to the 1940s the Harvard Observatory employed just such a group of all-female computers.) One could also create an analog computer, a model solar system made of gears and shafts that would "run" time into the future [see box on next page].

Before World War II, the most important analog computing instrument was the Differential Analyzer, developed by Vannevar Bush at the Massachusetts Institute of Technology in 1929. At that time, the U.S. was investing heavily in rural electrification, and Bush was investigating electrical transmission. Such problems could be encoded in ordinary differential equations, but these were very time-consuming to solve. The Differential Analyzer allowed for an approximate solution without any numerical processing. The machine was physically quite large—it filled a laboratory—and was something of a Rube Goldberg construction of gears and rotating shafts. To "program" the machine, researchers connected the various components of the device using screwdrivers, spanners and lead hammers. Though laborious to set up, once done the apparatus could solve in minutes equations that would take several days by hand. A dozen copies of the machine were built in the U.S. and England.

One of these copies belonged to the U.S. Army's Aberdeen Proving Ground in Maryland, the facility responsible for readying field weap-

## TEAMWORK



The Harvard Observatory's human computers, seen here circa 1890, examined hundreds of thousands of photographic plates between the 1880s and the 1920s, classifying stars based on color, position and brightness.

### [THE AUTHOR]



**Martin Campbell-Kelly** is a professor in the department of computer science at the University of Warwick in England, where he specializes in the history of computing. He is author of *Computer: A History of the Information Machine* (along with William Aspray) and of *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*. He is editor of *The Works of Charles Babbage*.

ons for deployment. To aim artillery at a target of known range, soldiers had to set the vertical and horizontal angles (the elevation and azimuth) of the barrel so that the fired shell would follow the desired parabolic trajectory—soaring skyward before dropping onto the target. They selected the angles out of a firing table that contained numerous entries for various target distances and operational conditions.

Every entry in the firing table required the integration of an ordinary differential equation. A human computer would take two to three days to do each calculation by hand. The Differential Analyzer, in contrast, would need only about 20 minutes.

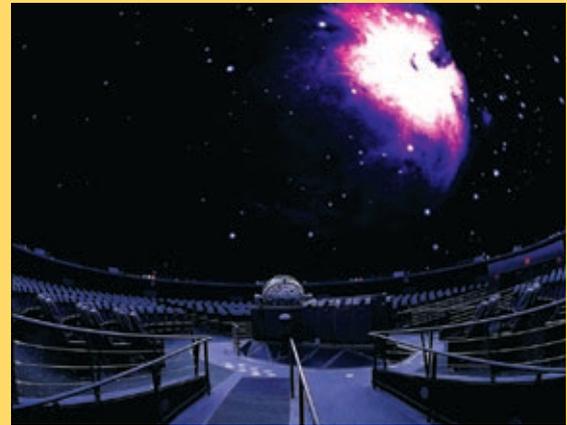
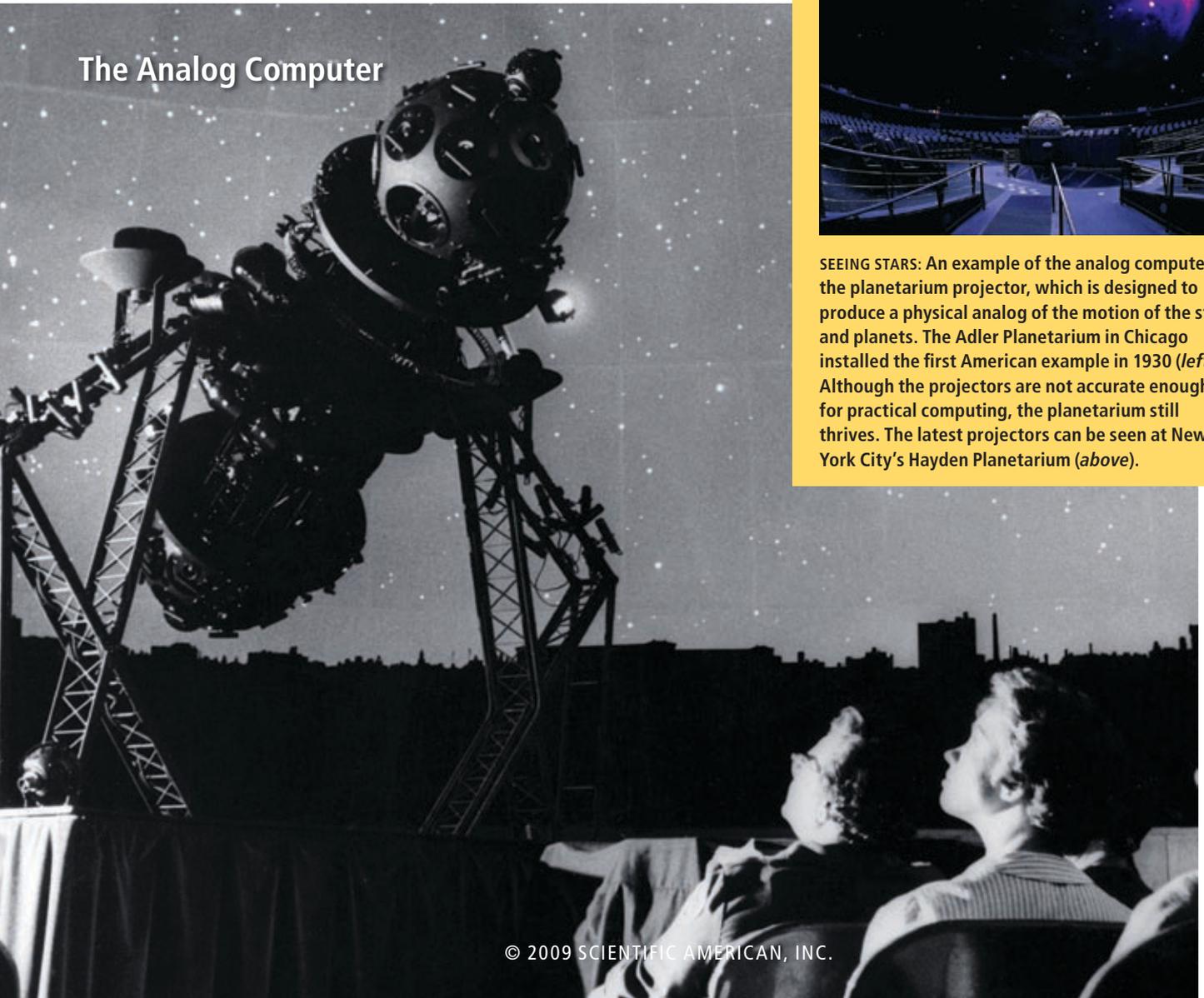
### Everything Is Change

On December 7, 1941, Japanese forces attacked the U.S. Navy base at Pearl Harbor. The U.S. was at war. Mobilization meant the army needed ever more firing tables, each of which con-

tained about 3,000 entries. Even with the Differential Analyzer, the backlog of calculations at Aberdeen was mounting.

Eighty miles up the road from Aberdeen, the Moore School of Electrical Engineering at the University of Pennsylvania had its own differential analyzer. In the spring of 1942 a 35-year-old instructor at the school named John W. Mauchly had an idea for how to speed up calculations: construct an “electronic computer” [sic] that would use vacuum tubes in place of the mechanical components. Mauchly, a theoretically-minded individual, found his complement in an energetic young researcher at the school named J. Presper (“Pres”) Eckert, who had already shown sparks of engineering genius.

## The Analog Computer



**SEEING STARS:** An example of the analog computer is the planetarium projector, which is designed to produce a physical analog of the motion of the stars and planets. The Adler Planetarium in Chicago installed the first American example in 1930 (left). Although the projectors are not accurate enough for practical computing, the planetarium still thrives. The latest projectors can be seen at New York City's Hayden Planetarium (above).

COURTESY OF THE ADLER PLANETARIUM (Zeiss projector);  
COURTESY OF D. FINNIN American Museum of Natural History (inset)

## The Digital Computer



A year after Mauchly made his original proposal, following various accidental and bureaucratic delays, it found its way to Lieutenant Herman Goldstine, a 30-year-old Ph.D. in mathematics from the University of Chicago who was the technical liaison officer between Aberdeen and the Moore School. Within days Goldstine got the go-ahead for the project. Construction of the ENIAC—for Electronic Numerical Integrator and Computer—began on April 9, 1943. It was Eckert's 23rd birthday.

Many engineers had serious doubts about whether the ENIAC would ever be successful. Conventional wisdom held that the life of a vacuum tube was about 3,000 hours, and the ENIAC's initial design called for 5,000 tubes. At that failure rate, the machine would not function for more than a few minutes before a broken tube put it out of action. Eckert, however, understood that the tubes tended to fail under the stress of being turned on or off. He knew it was for that reason radio stations never turned off their transmission tubes. If tubes were operated significantly below their rated voltage, they would last longer still. (The total number of tubes would grow to 18,000 by the time the machine was complete.)

Eckert and his team completed the ENIAC in two and a half years. The finished machine was an engineering tour de force, a 30-ton behemoth that consumed 150 kilowatts of power. The machine could perform 5,000 additions per second and compute a trajectory in less time than a shell took to reach its target. It was also a prime example of the role that serendipity often plays in invention: although the Moore School was not then a leading computing research facility, it happened to be in the right location at the right time with the right people.

Yet the ENIAC was finished in 1945, too late to help in the war effort. It was also limited in its capabilities. It could store only up to 20 numbers at a time. Programming the machine took days and required manipulating a patchwork of cables that resembled the inside of a busy telephone exchange. Moreover, the ENIAC was designed to solve ordinary differential equations. Some challenges—notably, the calculations required for the Manhattan Project—required the solution of partial differential equations.

John von Neumann was a consultant to the Manhattan Project when he learned of the ENIAC on a visit to Aberdeen in the summer of 1944. Born in 1903 into a wealthy Hungarian banking family, von Neumann was a mathe-

matical prodigy who tore through his education. By 23 he had become the youngest ever privatdozent (the approximate equivalent of an associate professor) at the University of Berlin. In 1930 he emigrated to the U.S., where he joined Albert Einstein and Kurt Gödel as one of first faculty members of the Institute for Advanced Study in Princeton, N.J. He became a naturalized U.S. citizen in 1937.

Von Neumann quickly recognized the power of electronic computation, and in the several months after his visit to Aberdeen, he joined in meetings with Eckert, Mauchly, Goldstine and Arthur Burks—another Moore School instructor—to hammer out the design of a successor machine, the Electronic Discrete Variable Automatic Computer, or EDVAC.

The EDVAC was a huge improvement over the ENIAC. Von Neumann introduced the ideas and nomenclature of Warren McCullough and Walter Pitts, neuroscientists who had developed a theory of the logical operations of the brain (this is where we get the term computer “memory”). Like von Neumann, McCullough and Pitts had been influenced by theoretical studies in the late 1930s by British mathematician Alan Turing, who established that a simple machine can be used to execute a huge variety of complex tasks. There was a collective shift in perception around this time from the computer as a mathematical instrument to a universal information-processing machine.

Von Neumann thought of the machine as having five core parts: *Memory* held not just numerical data but also the instructions for operation. An *arithmetic unit* performed calculations. An *input* “organ” enabled the transfer of

**POWER ON:** Computing entered the electronic age with the ENIAC, invented by J. Presper Eckert and John W. Mauchly of the Moore School of Electrical Engineering at the University of Pennsylvania. The ENIAC used vacuum tubes to hold numbers in storage and consumed 150 kilowatts of power, equivalent to more than 1,000 modern PCs.

**The Analytical Engine was almost like looking at a computer designed on another planet.**

# THE FUTURE OF COMPUTER ARCHITECTURE

The stored-program computer has formed the basis of computing technology since the 1950s. What may come next?

**QUANTUM:** The much touted quantum computer exploits the ability of a particle to be in many states at once. Quantum computations operate on all these states simultaneously.

**NEURAL NET:** These systems are formed from many simple processing nodes that connect to one another in unique ways. The system as a whole exhibits complex global behavior.

**LIVING:** Computers based on strands of DNA or RNA process data encoded in genetic material.

programs and data into memory, and an *output* organ recorded the results of computation. Finally, a *control unit* coordinated operations.

This layout, or architecture, makes it possible to change the computer's program without altering the physical structure of the machine. Moreover, a program could manipulate its own instructions. This feature would not only enable von Neumann to solve his partial differential equations, it would confer a powerful flexibility that forms the very heart of computer science.

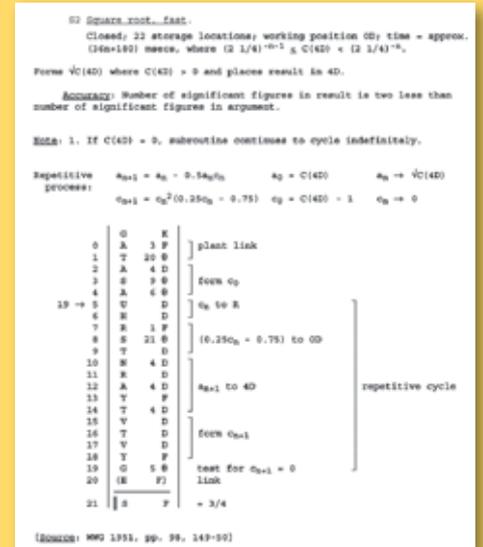
In June 1945 von Neumann wrote his classic *First Draft of a Report on the EDVAC* on behalf of the group. In spite of its unfinished status, it was rapidly circulated among the computing cognoscenti with two consequences. First, there never was a second draft. Second, von Neumann ended up with most of the credit.

## Machine Evolution

The subsequent 60-year diffusion of the computer within society is a long story that has to be told in another place. Perhaps the single most remarkable development was that the comput-

er—originally designed for mathematical calculations—turned out to be infinitely adaptable to different uses, from business data processing to personal computing to the construction of a global information network.

We can think of computer development as



**CHANGING PROGRAMS:** The first practical stored-program computer was the EDSAC, built at the University of Cambridge by Maurice Wilkes and William Renwick in 1949 (below). Early attempts to make a symbolic programming system (above) were a breakthrough in simplifying programming.

## The Stored-Program Computer



POPPEPROGETTY IMAGES (EDSAC); SOURCE: "THE EDSAC SIMULATOR PROGRAM DOCUMENTATION," BY THE DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF WARWICK (1987)

having taken place along three vectors—hardware, software and architecture. The improvements in hardware over the past 60 years are legendary. Bulky electronic tubes gave way in the late 1950s to “discrete” transistors—that is, single transistors individually soldered into place. In the mid-1960s microcircuits contained several transistors—then hundreds of transistors, then thousands of transistors—on a silicon “chip.” The microprocessor, developed in the early 1970s, held a complete computer processing unit on a chip. The microprocessor gave rise to the PC and now controls devices ranging from sprinkler systems to ballistic missiles.

The challenges of software were more subtle. In 1947 and 1948 von Neumann and Goldstine produced a series of reports called *Planning and Coding Problems for an Electronic Computing Instrument*. In these reports they set down dozens of routines for mathematical computation with the expectation that some lowly “coder” would be able to convert them into working programs. It was not to be. The process of writing programs and getting them to work was excruciatingly difficult. The first to make this discovery was Maurice Wilkes, the University of Cambridge computer scientist who had created EDSAC, the first practical stored-program computer [see box on opposite page]. In his *Memoirs*, Wilkes ruefully recalled the moment in 1949 when “the realization came over me with full force that a good part of the remainder of my life was going to be spent in finding errors in my own programs.”

He and others at Cambridge developed a method of writing computer instructions in a symbolic form that made the whole job easier and less error prone. The computer would take this symbolic language and then convert it into binary. IBM introduced the programming language Fortran in 1957, which greatly simplified the writing of scientific and mathematical programs. At Dartmouth College in 1964, educator John G. Kemeny and computer scientist Thomas E. Kurtz invented Basic, a simple but mighty programming language intended to democratize computing and bring it to the entire undergraduate population. With Basic even schoolkids—the young Bill Gates among them—could begin to write their own programs.

In contrast, computer architecture—that is, the logical arrangement of subsystems that make up a computer—has barely evolved. Nearly every machine in use today shares its basic architecture with the stored-program computer of



1945. The situation mirrors that of the gasoline-powered automobile—the years have seen many technical refinements and efficiency improvements in both, but the basic design is largely the same. And although it might be possible to design a radically better device, both have achieved what historians of technology call “closure.” Investments over the decades have produced such excellent gains that no one has had a compelling reason to invest in an alternative [see “Internal-Combustion Engine,” on page 97].

Yet there are multiple possibilities for radical evolution. In the 1980s interest ran high in so-called massively parallel machines, which contained thousands of computing elements operating simultaneously. This basic architecture is still used for computationally intensive tasks such as weather forecasting and atomic weapons research. Computer scientists have also looked to the human brain for inspiration. We now know that the brain contains specialized processing centers for different tasks, such as face recognition or speech understanding. Scientists are harnessing some of these ideas in “neural networks” for applications such as license plate identification and iris recognition.

More blue sky research is focused on building computers from living matter such as DNA [see “Bringing DNA Computers to Life,” by Ehud Shapiro and Yaakov Benenson; *SCIENTIFIC AMERICAN*, May 2006] and computers that harness the weirdness of the quantum world [see “The Limits of Quantum Computers,” by Scott Aaronson; *SCIENTIFIC AMERICAN*, March 2008]. No one knows what the computers of 50 years hence will look like. Perhaps their abilities will surpass even the powers of the minds that created them.

**CHILD’S PLAY:** Simple programming languages such as Basic allowed the power of programming to spread to the masses. A young Paul Allen (seated) and his friend Bill Gates worked on a Teletype terminal attached by a phone line to a mainframe computer that filled a room.

## ➔ MORE TO EXPLORE

**The Difference Engine: Charles Babbage and the Quest to Build the First Computer.** Doron Swade. Penguin, 2002.

**Computer: A History of the Information Machine.** Martin Campbell-Kelly and William Aspray. Westview Press, 2004.

**The Modern History of Computing.** Stanford Encyclopedia of Philosophy. <http://plato.stanford.edu/entries/computing-history>