1 Beginnings of a revolution

> Computer science also differs from physics in that it is not actually a science. It does not study natural objects. Neither is it, as you might think, mathematics. Rather, computer science is about getting something to do something. . . .
>
> Richard Feynman[1]

What is computer science?

It is commonplace to say that we are in the midst of a computing revolution. Computers are impacting almost every aspect of our lives. And this is just the beginning. The Internet and the Web have revolutionized our access to information and to other people. We see computers not only providing intelligence to the safety and performance of such things as cars and airplanes, but also leading the way in mobile communications, with present-day smart phones having more computing power than leading-edge computers only a decade ago. This book tells the story how this all came about, from the early days of computers in the mid-1900s, to the Internet and the Web as we know it today, and where we will likely be in the future.

The academic field of study that encompasses these topics draws from multiple disciplines such as mathematics and electronics and is usually known as *computer science*. As Nobel Prize recipient, physicist Richard Feynman says in the quotation that introduces this chapter, computer science is not a science in the sense of physics, which is all about the study of natural systems; rather, it is more akin to engineering, since it concerns the study of man-made systems and ultimately is about getting computers to do useful things. Three early computing pioneers, Allen Newell, Alan Perlis, and Herbert Simon, were happy to use science to describe what they did, but put forward a similar definition to Feynman: computer science is the study of computers. As we shall see, computer science has much to do with the management of complexity, because modern-day computers contain many billions of active components. How can such complex systems be designed and built? By relying on the principles of *hierarchical abstraction* and *universality*, the two main themes that underlie our discussion of computers.

Hierarchical abstraction is the idea that you can break down the design of a computer into layers so that you can focus on one level at a time without having to worry about what is happening at the lower levels of the hierarchy. Feynman in his *Lectures on Computation* makes an analogy with geology and the

Fig. 1.1 The famous geological map of Great Britain devised by William "Strata" Smith (1769–1839). Smith was a canal and mining engineer who had observed the systematic layering of rocks in the mines. In 1815, he published the "map that changed the world" – the first large-scale geological map of Britain. Smith was first to formulate the superposition principle by which rocks are successively laid down on older layers. It is a similar layer-by-layer approach in computer science that allows us to design complex systems with hundreds of millions of components.

work of William Smith, the founder of stratigraphy – the branch of geology that studies rock layers and layering (Fig. 1.1). While the layering approach used in computer science was not inspired by geological layers, Feynman's analogy serves as a useful memory hook for explaining hierarchical layers of computer architecture by reminding us that we can examine and understand things at each level (Fig. 1.2). This is the key insight that makes computers comprehensible.

Universality is linked to the notion of a universal computer that was introduced by Alan Turing and others. Turing suggested a very simple model for a computer called a Universal Turing Machine. This uses instructions encoded on a paper tape divided into sections with a very simple set of rules that the machine is to follow as the instruction in each section is read. Such a machine would be horribly inefficient and slow at doing complex calculations; moreover, for any specific problem, one could design a much more efficient, special-purpose machine. Universality is the idea that, although these other computers may be faster, the Universal Turing Machine can do any calculation that they can do. This is known as the Church-Turing thesis and is one of the cornerstones of computer science. This truly remarkable conjecture implies that your laptop, although much, much slower than the fastest supercomputer, is in principle just as powerful – in the sense that the laptop can do any calculation that can be done by the supercomputer!

So how did we get to this powerful laptop? Although the idea of powerful computational machines dates to the early nineteenth century, the direct line to today's electronic computers can be traced to events during World War II (1939–1945).

A chance encounter

There are many detailed histories of the origins of computing, and it would take us too far from our goal to discuss this history in detail. Instead, we will concentrate only on the main strands, beginning with a chance meeting at a train station.

In 1943, during World War II, the U.S. Army had a problem. Their Ballistic Research Laboratory (BRL) in Aberdeen, Maryland, was falling badly behind in its calculations of firing tables for all the new guns that were being produced. Each new type of gun needed a set of tables for the gunner that showed the correct angle of fire for a shell to hit the desired target. These trajectory calculations were then being carried out by a machine designed by MIT Professor Vannevar Bush. This was the differential analyzer (Fig. 1.3). It was an analog device, like the slide rules that engineers once used before they were made obsolete by digital calculators, but built on a massive scale. The machine had many rotating disks and cylinders driven by electric motors and linked together with metal rods, and had to be manually set up to solve any specific differential equation problem. This setup process could take as long as two days. The machine was used to calculate the basic trajectory of the shell before the calculation was handed over to an army of human "computers" who manually calculated the effects on this trajectory of other variables, such as the wind speed and direction. By the summer of 1944, calculating

Fig. 1.2 This sponge cake is a further analogy of abstraction layers. It is most certainly more appealing to our senses than the rock layers of geological periods.

Fig. 1.3 Vannevar Bush's Differential Analyzer was a complicated analog computer that used rotating discs and wheels for computing integrals. The complete machine occupied a room and linked several integration units connected by metal rods and gears. The Differential Analyzer was used to solve ordinary differential equations to calculate the trajectories of shells at the U.S. Army Ballistics Research Laboratory in Aberdeen, Maryland.

these tables was taking far too long and the backlog was causing delays in gun development and production. The situation seemed hopeless since the number of requests for tables that BRL received each week was now more than twice its maximum output. And this was after BRL had doubled its capacity by arranging to use a second differential analyzer located in the Moore School of Electrical Engineering at the University of Pennsylvania in Philadelphia. Herman Goldstine was the young army lieutenant in charge of the computing substation at the Moore School. And this was why he happened to be on the platform in Aberdeen catching a train back to Philadelphia on an evening in August 1944.

It was in March 1943 that Goldstine had first heard of a possible solution to BRL's problems. He was talking to a mechanic at the Moore School and learned of a proposal by an assistant professor, John Mauchly (B.1.1), to build an electronic calculator capable of much faster speeds than the differential analyzer. Mauchly was a physicist and was originally interested in meteorology. After trying to develop a weather prediction model he soon realized that without some sort of automatic calculating machine this task was impossible. Mauchly therefore developed the idea of building a fast electronic computer using vacuum tubes.

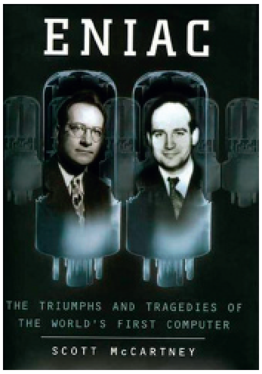
Goldstine was a mathematician by training, not an engineer, and so was not aware of the generally accepted wisdom that building a large-scale computer with many thousands of vacuum tubes was considered impossible because of the tubes' intrinsic unreliability. After talking with Mauchly, Goldstine asked him to submit a full proposal for such a vacuum-tube machine to BRL for funding. Things moved fast. Mauchly, together with the smartest graduate of the school, J. Presper Eckert, gave a presentation on their new proposal in Aberdeen less than a month later. They got their money – initially $150,000 – and Project PX started on June 1, 1943. The machine was called the ENIAC, usually taken to stand for the Electronic Numerical Integrator And Computer.

It was while he was waiting for his train back to Philadelphia that Goldstine caught sight of a man he recognized. This was the famous mathematician John von Neumann (B.1.2), whom Goldstine had heard lecture on several occasions in his research as a mathematician before the war. As he later wrote:

> It was therefore with considerable temerity that I approached this world-famous figure, introduced myself and started talking. Fortunately for me von Neumann was a warm, friendly fellow who did his best to make people feel relaxed in his presence. The conversation soon turned to my work. When it became clear to von Neumann that I was concerned with the development of an electronic computer capable of 333 multiplications per second, the whole atmosphere of our conversation changed from one of relaxed good humor to one more like an oral examination for a doctor's degree in mathematics.[2]

Soon after that meeting, Goldstine went with von Neumann to the Moore School so that von Neumann could see the ENIAC (Fig. 1.4) and talk with Eckert and Mauchly. Goldstine remembers Eckert's reaction to the impending visit:

> He [Eckert] said that he could tell whether von Neumann was really a genius by his first question. If this was about the logical structure of the machine,



B.1.1 John Mauchly (1907–80) and Presper Eckert (1919–95) were the designers of ENIAC. With John von Neumann, they went on to propose the EDVAC, a design for a stored-program computer, but unfortunately their future efforts were complicated by legal wrangling over intellectual property and patents. As a result, they left the Moore School at the University of Pennsylvania and set up a company to build the UNIVAC, the first successful commercial computer in the United States.

> he would believe in von Neumann, otherwise not. Of course this was von Neumann's first query.[3]

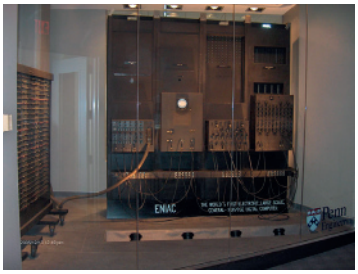


Fig. 1.4 A section of the original ENIAC machine on display at the University of Pennsylvania.

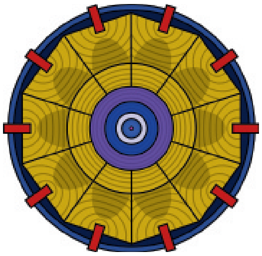


Fig. 1.5 A schematic diagram of the spherical implosion lens required to start the nuclear reaction in a plutonium bomb. John von Neumann's search for an automatic device that would speed up the complex calculations needed to model the lens led to his interest in ENIAC.

The reason why von Neumann was so interested in the ENIAC was because of his work for the Manhattan atom bomb project at Los Alamos, New Mexico. The physicists at Los Alamos had a bottleneck in their schedule to produce a plutonium bomb. This was due to the complex calculations needed to model the spherical implosive lens for the bomb (Fig. 1.5). The lens was formed by accurately positioned explosives that produced a spherical compression wave. The wave would then compress the plutonium at the center of the sphere to criticality and thereby start the nuclear chain reaction. Von Neumann had asked Bush's Office of Scientific Research and Development (OSRD) for suggestions as to how this calculational bottleneck could be removed. He was advised to look at three automatic calculator projects that OSRD was funding that might deliver the increased computing power he needed. By the time he met Goldstine, von Neumann had concluded that none of the suggested projects, which included the Mark I, an electromechanical computer created by IBM and Howard Aiken at Harvard, would be of any help. The OSRD had made no mention of the Army-funded ENIAC project, since this was regarded by Bush and others as just a waste of money. The ENIAC team were therefore glad to welcome the famous von Neumann into their camp, and they had regular discussions over the next few months.

The ENIAC was completed in November 1945, too late to help the war effort. It was eight feet high, eighty feet long, and weighed thirty tons. It contained approximately 17,500 vacuum tubes, 70,000 resistors, 10,000 capacitors, 1,500 relays, and 6,000 manual switches. It consumed 174 kilowatts of power – enough to power several thousand laptops. Amazingly, only fifty years later, all of this monster amount of hardware could be implemented on a single chip (Fig. 1.6). Fortunately, the vacuum tubes turned out to be far more reliable than

B.1.2 John von Neumann (1903–57) was born in Budapest in the family of a wealthy banker. After graduating with a PhD in mathematics from Budapest ELTE and a diploma in chemical engineering from Zurich ETH, he won a scholarship in Gottingen and worked with David Hilbert on his ambitious program on the "axiomatization" of mathematics. In 1933, von Neumann was offered an academic position at the Institute for Advanced Study in Princeton, and was one of the institute's first four professors.

Von Neumann's extraordinary talent for mathematics and languages was evident from early in his childhood. At university, his teacher George Polya at the ETH in Zurich said of him:

> He is the only student of mine I was ever intimidated by. He was so quick. There was a seminar for advanced students in Zurich that I was teaching and von Neumann was in the class. I came to a certain theorem, and I said it is not proved and it may be difficult. Von Neumann did not say anything but after five minutes he raised his hand. When I called on him he went to the blackboard and proceeded to write down the proof. After that I was afraid of von Neumann.[B1]

Von Neumann was a genuine polymath who made pioneering contributions to game theory, quantum mechanics, and computing. He also hosted legendary cocktail parties, but his driving skills apparently left something to be desired:

> Von Neumann was an aggressive and apparently reckless driver. He supposedly totaled a car every year or so. An intersection in Princeton was nicknamed "Von Neumann Corner" for all the auto accidents he had there.[B2]

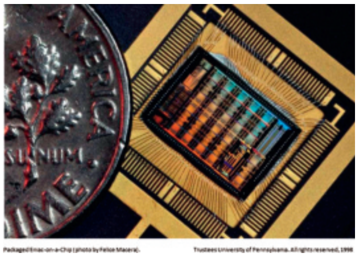


Fig. 1.6 The ENIAC on a chip. This chip was designed to mark the fiftieth anniversary of the ENIAC project by a group of students at the University of Pennsylvania. This 0.5 cm^2 chip can do the same computations as the original 30-ton computer in 1946. No other technology in the course of human history has achieved this pace of development.

anyone had expected. The calculational speed of the ENIAC was impressive – it was more than a thousand times faster than Aiken's Mark I machine. On ten-digit numbers, the machine could calculate more than five thousand additions or three hundred multiplications per second! However, although this was very much faster than the differential analyzer and the Mark I in terms of its basic operations, it still took about two days to set up the ENIAC to solve a specific problem – and this was after the operators had written a program specifying the correct sequence of operations.

Writing an ENIAC program required the programmer to have almost as much knowledge of the machine as its designers did (Fig. 1.7). The program was implemented by setting the ENIAC's switches to carry out the specific instructions and by plugging in cables to arrange for these instructions to be executed in the correct order. The six women who did most of the programming for the ENIAC were finally inducted into the Women in Technology International Hall of Fame in 1997 (Fig. 1.8).

The first problem to be performed by the ENIAC was suggested by von Neumann. The problem arose from his work at Los Alamos and involved the complex calculations necessary to evaluate a design for Edward Teller's proposed hydrogen bomb. The results revealed serious flaws in the design. Norris Bradbury, Director of the Los Alamos Laboratory, wrote a letter to the Moore School saying, "The complexity of these problems is so great that it would have been impossible to arrive at any solution without the aid of ENIAC."[4]

Von Neumann and the stored-program computer

After the ENIAC design was finalized and the machine was being built, Eckert and Mauchly had time to think about how they could design a better computer using new memory storage technologies. It had become clear to them that the ENIAC needed the ability to store programs. This would enable programmers to avoid the lengthy setup time. Eckert and Mauchly probably came up with this idea for a *stored-program computer* sometime in late 1943 or early 1944. Unfortunately for them, they never got around to explicitly writing down their ideas in a specific design document for their next-generation computer. There are only some hints of their thinking in their progress reports on the construction of the ENIAC, but there now seems little doubt that they deserve at least to share the credit for the idea of the stored-program computer. When von Neumann first arrived at the Moore School in September 1944, he was briefed by Eckert and Mauchly about their ideas for a new machine they called EDVAC – Electronic Discrete Variable Computer. According to Mauchly's account, they told von Neumann the following:

> We started with our basic ideas: there would be only one storage device (with addressable locations) for the entire EDVAC, and this would hold both data and instructions. All necessary arithmetic operations would be performed in just one arithmetic unit (unlike the ENIAC). Of course, there would be devices to handle input and output, and these would be subject to the control module just as the other modules were.[5]

Fig. 1.7 U.S. Army ENIAC poster. The ENIAC was advertised as a work opportunity for mathematicians and puzzle solvers.

In the months that followed, the three of them refined their ideas for the EDVAC, which eventually resulted in von Neumann writing a paper, titled the

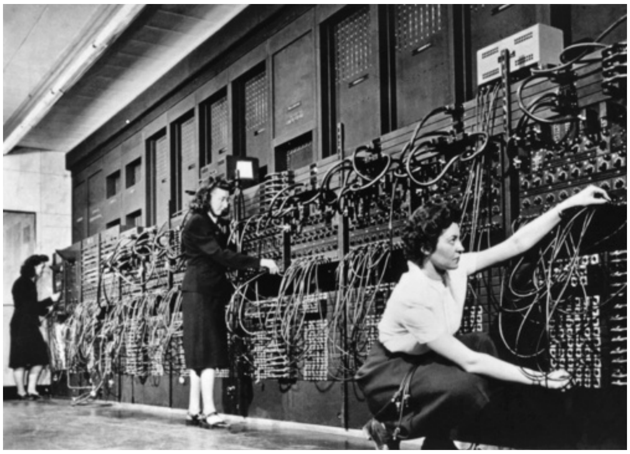


Fig. 1.8 The first programmers of ENIAC were women. In those days, programming meant setting all switches and rewiring the computer, a tedious operation that often took days to complete.

"First Draft of a Report on the EDVAC." Although von Neumann had left blank spaces on his draft for the names of co-authors, unfortunately for Ekert and Mauchly, Goldstine went ahead and released the paper listing von Neumann as the sole author. The report contained the first description of the logical structure of a stored-program computer and this is now widely known as the von Neumann architecture (Fig. 1.9).

The first great abstraction in the report was to distinguish between the computer hardware and software. On the hardware side, instead of going into detail about the specific hardware technology used to build the machine, von Neumann described the overall structure of the computer in terms of the basic logical functions that it was required to perform. The actual hardware that performed these functions could be implemented in a variety of technologies – electromechanical switches, vacuum tubes, transistors, or (nowadays) modern silicon chips. All these different technologies could deliver the same computational capabilities, albeit with different performance. In this way, the problem of how the logical components are put together in a specific order to solve a particular problem has now been separated from concerns about the detailed hardware of the machine. This splitting of responsibilities for the hardware design and for the programming of the machine was the beginning of two entirely new engineering disciplines: computer architecture and software engineering.

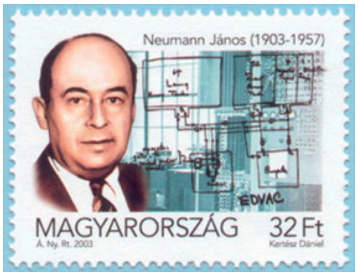


Fig. 1.9 A Hungarian postage stamp that honors John von Neumann, complete with the mathematician's likeness and a sketch of his computer architecture.

For the hardware of the machine, von Neumann identified five functional units: the central arithmetic unit (CA), the central control unit (CC), the memory (M), the input (I), and the output (O) (Fig. 1.10). The CA unit carried out all the arithmetic and logical operations, and the CC unit organized the sequence of operations to be executed. The CC is the *conductor*, since it coordinates the operation of all components by fetching the instructions and data from the memory and providing clock and control signals. The CA's task is to perform the required calculations. The memory was assumed to store both programs and data in a way that allowed access to either program instructions or data. The I/O units could read and write instructions or data into and out of the computer memory directly. Finally, unlike the ENIAC, which had used decimal arithmetic, von Neumann recommended that the EDVAC use binary arithmetic

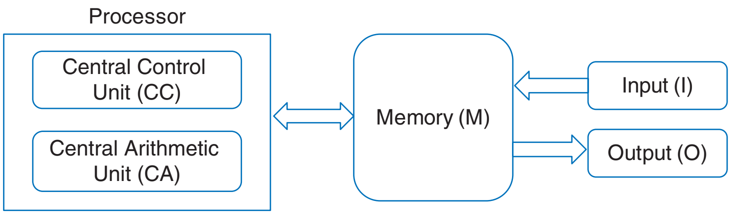
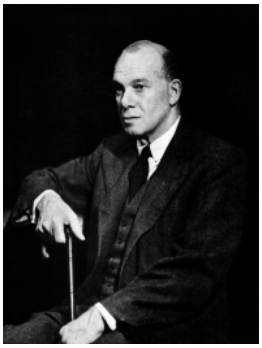


Fig. 1.10 The von Neumann Architecture. The main building blocks of all computers are the input, output, memory, and processor. The input (typically now a keyboard or a mouse) feeds data into the computer. This information is encoded by binary numbers and stored in the memory. The processor then fetches the information, decodes it, and performs the required calculations. The results are put back in the memory, where they can be read by the output device (typically a monitor, printer, or even a loudspeaker). The processor consists of two components: the Central Control Unit (CC) and the Central Arithmetic Unit (CA), now known as the Arithmetical and Logical Unit (ALU).

for its operations. As we shall see in Chapter 2, binary, base-2 arithmetic is much better suited to efficient and simple electronic implementations of arithmetic and logic operations.

How does this von Neumann architecture relate to Turing's ideas about universality? Before the war, Turing had spent time in Princeton and von Neumann was well aware of the groundbreaking paper on theoretical computing machines he had completed as a student in Cambridge, UK. The memory, input, and output of von Neumann's abstract architecture are logically equivalent to the tape of a Universal Turing Machine, and the arithmetic and central control units are equivalent to the read/write component of Turing's logical machine. This means that no different computer design can do any different calculations than a machine built according to von Neumann's architecture. Instead of coming up with new architectures, computer engineers could spend their time optimizing and improving the performance of the von Neumann design. In fact, as we will see later, there are ways of improving on his design by eliminating the so-called von Neumann bottleneck – in which all instructions are read and executed serially, one after another – by using multiple processors and designing *parallel computers*.



B.1.3 Leslie Comrie (1893–1950) was an astronomer and an expert on numerical calculations. He visited the Moore School in 1946 and brought the first copy of the EDVAC report back to Britain.

The global EDVAC diaspora

There were thirty-two people on the original mailing list for the "Report on the EDVAC" but news of the report soon spread far and wide. With World War II having come to an end, scientists were once again able to travel internationally, and by early 1946 the Moore School had already had several visitors from Britain. The first visitor from the United Kingdom to the Moore School was a New Zealander named Leslie Comrie (B.1.3). Comrie had a longtime interest in astronomy and scientific computation, and during the war he had led a team of scientists to computerize such things as bombing tables for the Allied Air Force. Remarkably, after his visit to see the ENIAC, Comrie was allowed to take a copy of the EDVAC report back to England. Back in England, he went to visit Maurice Wilkes (see Timeline) in Cambridge. Wilkes was a mathematical physicist who had returned from war service and was trying to

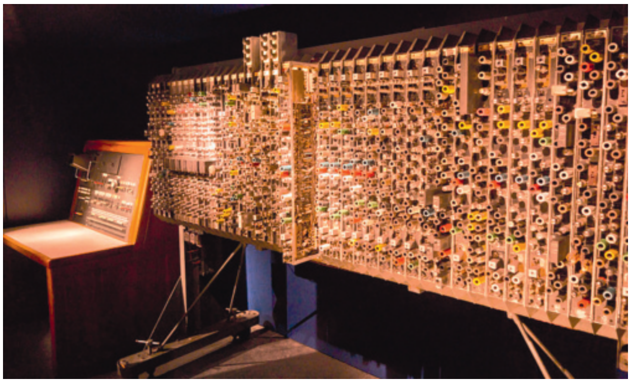
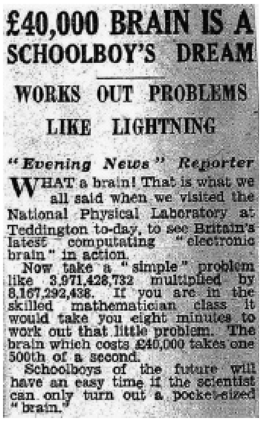


Fig. 1.11 The Pilot ACE was a computer with a distinctive flavor. Turing's design was much more detailed than that contained in von Neumann's EDVAC report published only three months earlier. Pilot ACE had many innovative features, such as three address instructions, variable-length block transfer, and bit-level manipulation, but it was difficult to program. This is one of the reasons why this unique design had little impact on the architecture of computers.

establish a viable computing laboratory in Cambridge. Wilkes recalls in his memoirs:

> In the middle of May 1946 I had a visit from L.J. Comrie who was just back from a trip to the United States. He put in my hands a document written by J. von Neumann on behalf of the group at the Moore School and entitled "Draft Report on the EDVAC." Comrie, who was spending the night at St. John's College, obligingly let me keep it until the next morning. Now, I would have been able to take a Xerox copy, but there were then no office copiers in existence and so I sat up late into the night reading the report. In it, clearly laid out, were the principles on which the development of the modern digital computer was to be based: the stored program with the same store for numbers and instructions, the serial execution of instructions, and the use of binary switching circuits for computation and control. I recognized this at once as the real thing, and from that time on never had any doubt as to the way computer development would go.[6]

Another early visitor to the Moore School was J. R. Womersley from the U.K. National Physical Laboratory. Womersley had worked with differential analyzers and was duly impressed by the performance of the ENIAC. As a result of this visit, Womersley set about organizing a computing project at his laboratory and hired Turing to lead the team. Turing read von Neumann's report and then designed his own plan for a stored-program computer called ACE –Automatic Computing Engine (Figs. 1.11 and 1.12), where his use of the word *engine* was a deliberate homage to Charles Babbage. The ACE design report describes the concept for the machine in the following words:



£40,000 BRAIN IS A SCHOOLBOY'S DREAM

WORKS OUT PROBLEMS LIKE LIGHTNING

"Evening News" Reporter

WHAT a brain! That is what we all said when we visited the National Physical Laboratory at Teddington to-day, to see Britain's latest computating "electronic brain" in action.

Now take a "simple" problem like 3,971,428,732 multiplied by 8,167,292,438. If you are in the skilled mathematician class it would take you eight minutes to work out that little problem. The brain which costs £40,000 takes one 500th of a second.

Schoolboys of the future will have an easy time if the scientist can only turn out a pocket-sized "brain."

Fig. 1.12 The *London Evening News* from November 28, 1950, reporting the speed of the Pilot ACE computer.

> It is intended that the setting up of the machine for new problems shall be virtually only a matter of paper work. Besides the paper work nothing will have to be done except to prepare a pack of Hollerith cards in accordance with this paper work, and to pass them through a card reader connected to the machine. There will positively be no internal alterations to be made even if we wish suddenly to switch from calculating the energy levels of the neon atom to the enumeration of groups of order 720. It may appear puzzling that


Fig. 1.13 The Moore School of Electrical Engineering at the University of Pennsylvania, where the ENIAC was born.

> this can be done. How can one expect a machine to do all this multitudinous variety of things? The answer is that we should consider the machine as doing something quite simple, namely carrying out orders given to it in a standard form which it is able to understand.[7]

This is not the last computing project to underestimate the difficulties associated with the "paper work" or, as we would now say, "programming the machine"!

In 1946, at the instigation of the new dean of the Moore School, Howard Pender, the Army Ordnance Department, and the U.S. Office of Naval Research sponsored a summer school on stored-program computing at the Moore School (Fig. 1.13). There were thirty to forty invitation-only participants mainly from American companies, universities, and government agencies. Alone among the wartime allies, Britain was invited to participate in the summer school. The Moore School Lectures on Computing took place over eight weeks in July and August, and besides Eckert and Mauchly, Aiken and von Neumann made guest appearances as lecturers. The first part of the course was mainly concerned with numerical mathematics and details of the ENIAC. It was only near the end of the course that security clearance was obtained that enabled the instructors to show the participants some details of the EDVAC design. Wilkes had received an invitation from Dean Pender and, despite funding and visa problems, decided it was worth going since he thought he was "not going to lose very much in consequence of having arrived late."[8] After attending the last two weeks of the school, Wilkes had time to visit Harvard and MIT before he left the United States. At Harvard he saw Howard Aiken's Mark I and II electromechanical computers, and at MIT he saw a new version of Bush's differential analyzer. He left the United States more convinced than ever that the future was not going to follow such "dinosaurs" but instead follow the route laid out by the EDVAC report for stored-program computers. On his return to Cambridge in England, Wilkes started a project to build the Electronic Delay Storage Automatic Calculator – usually shortened to EDSAC, in conscious homage to its EDVAC heritage.

The EDSAC computer became operational in 1949. In these early days of computing, a major problem was the development of suitable memory devices to store the binary data. Eckert had had the idea of using tubes filled up with mercury to store sound waves traveling back and forth to represent the bits of data, and Wilkes was able to successfully build such mercury delay line memory for the EDSAC. A variant on Wilkes's design for the EDSAC was developed into a commercial computer called Lyons Electronic Office, or LEO. It was successfully used for running business calculations for the network of Lyon's Corner Houses and Tea Shops. Wilkes later introduced the idea of microprogramming, which enabled complicated operations to be implemented in software rather than hardware. This idea significantly reduced the hardware complexity and became one of the key principles of computer design.

Meanwhile, back in the United States, Eckert and Mauchly had resigned from the Moore School after an argument over patent rights with the university and were struggling to get funding to build a commercial computer.

Fig. 1.14 Tom Kilburn and Freddie Williams with the "Baby" computer in Manchester. The machine had only seven instructions and had 32 × 32 bits of main memory implemented using a cathode ray tube.

After many difficulties, they ultimately succeeded in designing and building the famous UNIVAC (UNIVersal Automatic Computer) machine. With the war ended, von Neumann returned to Princeton and wasted no time getting funds to build an EDVAC architecture computer for the Institute for Advanced Study (IAS). He quickly recruited Goldstine and Arthur Burks from the EDVAC team and a talented engineer, Julian Bigelow, to help him design the IAS machine (see Timeline). In 1947, with Goldstine, von Neumann wrote the first textbook on software engineering called *Planning and Coding Problems for an Electronic Computing Instrument*.

While commercial interest in computers was beginning to develop in the United States, it was actually two teams in the United Kingdom that first demonstrated the viability of the stored-program computer. At Manchester, Freddie Williams and Tom Kilburn had followed the path outlined by von Neumann and in June 1948 they had a prototype machine they called Baby (see Timeline and Fig. 1.14). This ran the first stored program on an electronic computer on 21 June 1948. This success was followed in May 1949 by Wilkes's EDSAC machine in Cambridge – which was undoubtedly the first stored-program computer with any significant computational power.

Key concepts

- Computation can be automated
- Layers and abstractions
- The stored program principle
- Separation of storage and processing
- Von Neumann architecture

Cartoon illustrating the requirement for calculating shell trajectories.

Some early history of computing

An idea long in the making

While the origins of the modern electronic computer can be traced back to EDVAC in the 1940s, the idea of powerful computational machines goes back much further, to the early nineteenth century and an Englishman named Charles Babbage.

Charles Babbage and the Difference Engine

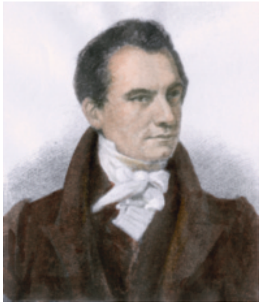
The first government-funded computer project to overrun its budget was Charles Babbage's attempt to construct his Difference Engine in 1823. The project had its origins in the problem of errors in the mathematical tables of the *navigator's bible*, the British Nautical Almanac. These errors, either from mistakes in calculation or from copying and typesetting, plagued all such tables and were popularly supposed to be the cause of numerous shipwrecks. One study of a random selection of forty volumes of tables found three thousand errors listed on correction or errata sheets. Some sheets were even correction sheets for earlier corrections!

Charles Babbage (B.1.4) was a mathematician and a student at Cambridge University in 1812 when he first had the idea of using a machine to calculate mathematical tables. He wrote about the moment in his autobiography:

> One evening I was sitting in the rooms of the Analytical Society, at Cambridge, my head leaning forward on the table in a kind of dreamy mood, with a table of logarithms lying open before me. Another member, coming into the room, and seeing me half asleep called out "Well, Babbage, what are you dreaming about?" to which I replied, "I am thinking that all these tables (pointing to the logarithms) might be calculated by machinery."[9]

Some years later Babbage was checking astronomical tables with his astronomer friend John Herschel. They each had a pile of papers in front of them containing the results for the tables as calculated by "computers." In those days, computers were not machines but people who had been given a precise arithmetical procedure to do the routine calculations by hand. The two piles contained the same set of calculations, each done by different computers but both should be identical. By comparing the results line by line Babbage and Herschel found a number of errors and the whole process was so slow and tedious that Babbage finally exclaimed "I wish to God these calculations had been executed by steam."[10]

As a result of his experience, Babbage spent the next few years designing what he called the Difference Engine – a mechanical machine that was able to calculate astronomical and navigational tables using a mathematical process called the method of constant differences. Correct calculations were only part of the problem however, since the copying and typesetting of the results were equally error prone. In order to eliminate these errors, Babbage designed his machine to record the results on metal plates that could be used directly for printing. By 1822 he had built a small working prototype and made a proposal to the Royal Society that a large,



B.1.4 Charles Babbage (1792–1871) was the son of a wealthy banker. He studied mathematics at Cambridge and was the leader of a radical group of students that overthrew the negative legacy of Isaac Newton's approach to calculus on mathematics in England by introducing new notation and mathematical techniques from France and Germany. Babbage is now known for his pioneering work in computing, but he was also a prolific inventor. Among other things, he invented the ophthalmoscope, a cowcatcher, the seismograph, and a submarine propelled by compressed air. However, Babbage's computing engines were never completed, and he died a disappointed man.

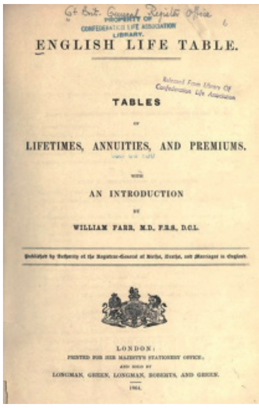


Fig. 1.15 The front page of *English Life Table* from 1864, portions of which were composed on a machine inspired by Babbage's Difference Engine. Unfortunately the machine lacked some of Babbage's error-protection mechanisms and this severely hampered its usefulness.

full-scale Difference Engine be built. As a result of his proposal, Babbage was encouraged to seek funding from the U.K. government, and he eventually received what was then an unprecedented offer of £1,500 in financial support. To give some idea of how this amount compares in today's prices, Doron Swade in his book on the Difference Engine states that "A gentleman in 1814, for example, could expect to support his wife and a few children in modest comfort on an income of £300 a year."[11] So in 1822, this represented a very significant investment by the government.

The full-scale engine would need thousands of precisely engineered gears and axles. Babbage therefore had to begin the project by spending a great deal of time with his skilled engineering draughtsman, Joseph Clement, devising and designing better machine tools capable of producing hundreds of identical parts. It was unfortunate for Babbage that he was building his machine at a time when manufacturing technology in Britain was in transition between individual craft manufacture and mass-production methods. A symptom of this was that there were no standards for even simple components like screws. One of Clement's engineers, Joseph Whitworth, later played a major role in the adoption of the standard Whitworth screw thread by the U.K. manufacturing industry.

By 1832, almost ten years and £10,000 after the project began, Babbage and Clement had built a small part of the machine, complete except for the printing mechanism. It worked perfectly and was able to deliver calculations to six-digit accuracy. Alas, at this point the project was brought to a standstill by a dispute between Babbage and Clement and work on the machine was never resumed. By the time of the formal cancellation of the project in 1842, the total cost to the U.K. government was more than £17,000; by comparison, a new steam engine built by Robert Stephenson for shipping to the United States in 1831 had cost less than £800.

The first article giving any substantial details of Babbage's design for his Difference Engine appeared in 1834, written by a colorful character called Dionysius Lardner and published under the title "Babbage's Calculating Engine" in the *Edinburgh Review*. After reading this lengthy and somewhat eulogistic article, George Scheutz, a Stockholm printer, publisher, and journalist, took up the cause of mechanized calculations. Without having access to full details of Babbage's mechanisms for the various operations, Scheutz invented his own. What is more, by the summer of 1843, he and his son Edvard had actually built a working machine, including the printing mechanism. Although they were originally concerned about Babbage's reaction to their efforts, the truth is that Babbage was delighted and set about helping them market their machine. In 1857, after winning a gold medal at the 1855 Great Exposition in Paris, one machine was sold for £1,000 to Dr. Benjamin Gould, the director of the Dudley Observatory in Albany, New York, as part of a flagship project to make the Albany observatory the "American Greenwich." Gould used the machine to calculate a set of tables for the orbit of the planet Mars. Alas, in 1859 Gould was fired and Scheutz's Tabulating Machine eventually was donated to the Smithsonian Museum.

Another machine was purchased by the United Kingdom's General Register Office. William Farr, the chief statistician, wanted to use the machine to automate the production of the tables of life expectancy, annuities, insurance premiums, and interest payments that he needed for the 1864 *English Life Table* (Fig. 1.15). The machine was built by Donkin & Company in London and was used for the English Life Table project. However, because the machine did not have all of Babbage's careful error-protection mechanisms, it proved to need constant care and attention. At the end of the project, only twenty-eight of the six hundred pages of printed tables in the *English Life Table* were entirely composed by the machine and 216 were partially

Fig. 1.16 Postage stamp issued to mark the bicentenary of Babbage's birth.

composed. The conclusion was that the machine had failed to deliver any significant benefits or cost savings. A sad postscript to this story is that both Scheutz and his son ended their lives bankrupt, a condition at least partly caused by their overenthusiasm for their calculating engines.

There is a positive postscript to this story. Babbage's son, Henry, inherited most of the unused parts for the Difference Engine that had been manufactured by Clements. Although many of the parts went for scrap, Henry saved enough parts to assemble six small demonstration machines. He sent these to several universities including Cambridge, University College London, and Manchester in the United Kingdom and Harvard in the United States. In the late 1930s, Howard Aiken, who with IBM pioneered the development of the Harvard Mark I, one of the early electromechanical computers, discovered the small demonstration engine sent to Harvard. He later said that he "felt that Babbage was addressing him personally from the past."[12]

It was not until 1991 that Doron Swade and a team at the Science Museum in the United Kingdom unveiled a working model of a full-scale Difference Engine built following Babbage's designs (Fig. 1.16). This demonstration showed conclusively that such machines could have been built in Babbage's day, albeit with a huge amount of engineering effort (see Fig. 1.17).

The Analytical Engine

Babbage was not a master of tact. Instead of finishing his Difference Engine, he unwisely suggested that the government abandon work on the still incomplete original machine and build a much more powerful and versatile machine he called the Analytical Engine. The Difference Engine was essentially a special-purpose calculator, and Babbage had realized that he could design a much more general-purpose machine capable of performing any arithmetical or logic operation. He conceived the idea for this vastly more powerful and flexible machine between 1834 and 1836 and kept tinkering and improving his design until the day he died. This Analytical Engine was never built and was therefore only a thought experiment. Nevertheless, its design

Fig. 1.17 In 1991, to mark the bicentenary of Babbage's birth, Doron Swade and his colleagues at the London Science Museum unveiled the Difference Engine II, a working model constructed according to Babbage's original designs. The computing historians and engineers went to great lengths to preserve the authenticity, using the original drawings, materials, and precision of manufacturing available in Babbage's time. They considered this work as the continuation of Babbage's project, but almost 150 years later. Difference Engine II contains about eight thousand cogs and weighs about 4.5 tons. It is operated by a crank handle, as can be seen in the accompanying photo of Swade cranking the machine. The design included numerous failsafe features, such as mechanical parity checking that prevented errors occurring even when some cogs get deranged due to vibrations. The cogwheels were manufactured so that they could fracture in a controlled way; this is an equivalent of a mechanical fuse. It must be said, however, that the machine was overdimensioned. It can calculate to an accuracy of forty-four binary digits. This looks excessive, especially if we consider that many of the machines we use today calculate accurately only to thirty-two binary digits. The machine can also calculate up to seventh-order polynomials, today we usually use third-order polynomials. It is not clear why Babbage thought he needed this level of accuracy; a simpler machine would have saved him many cogs and certainly would have made the construction much easier. Babbage saw the computer as an integral part of the calculation process, which he envisaged as a factory that produces numbers.

Fig. 1.18 A photograph of Jacquard's Loom showing the punched cards encoding the instructions for producing intricate patterns. The program is a sequence of cards with holes in carefully specified positions. The order of the cards and the positions of these holes determine when the needles should be lifted or lowered to produce the desired pattern.

B.1.5 Joseph-Marie Jacquard (1752–1834) (left) and Philippe de la Salle (1723–1804) pictured on a mural in Lyon (Mur des Lyonnais). Philippe de la Salle was a celebrated designer, who made his name in the silk industry. Jacquard's use of punched cards to provide the instructions for his automated loom inspired Babbage, who proposed using punched cards to program his Analytical Engine.

captured many of the principles to be found in today's computers. In particular, Babbage's design separated the section of the machine where the various arithmetical operations were performed from the area where all the numbers were kept before and after processing. Babbage named these two areas of his engine using terms borrowed from the textile industry: the *mill* for the calculations, which would now be called the central processing unit or CPU, and the *store* for storing the data, which would now be called computer memory. This separation of concerns was a fundamental feature of von Neumann's famous report that first laid out the principles of modern computer organization.

Another key innovation of Babbage's design was that the instructions for the machine – or program as we would now call them – were to be supplied on punched cards. Babbage got the idea of using punched cards to instruct the computer from an automatic loom (Fig. 1.18) invented in France by Joseph-Marie Jacquard (B.1.5). The cards were strung together to form a kind of tape and then read as they moved through a mechanical device that could sense the pattern of holes on the cards. These looms could produce amazingly complex images and patterns. At his famous evening dinner parties in London, Babbage used to show off a very intricate silk portrait of Jacquard that had been produced by a program of about ten thousand cards.

Babbage produced more than six thousand pages of notes on his design for the Analytical Engine as well as several hundred engineering drawings and charts indicating precisely how the machine was to operate. However, he did not publish any scientific papers on the machine and the public only learned about his new ambitious vision through a presentation that Babbage gave in 1840 to Italian scientists in Turin. The report of the meeting was written up by a remarkable young engineer called Luigi Menabrea – who later went on to become a general in the Italian army and then prime minister of Italy.

Ada Lovelace

It is at this point in the story that we meet Augusta Ada, Countess of Lovelace (B.1.6), the only legitimate daughter of the Romantic poet, Lord Byron. Lovelace first met Babbage at one of his popular evening entertainments in 1833 when she was seventeen. Less than two weeks later, she and her mother were given a personal demonstration of his small prototype version of his computing engine. Unusually for women of the time, at the insistence of her father, Ada had had some mathematical training. After this first meeting with Babbage, Ada got married and had children but in 1839 she wrote to Babbage asking him to recommend a mathematics tutor for her. Babbage recommended

Augustus De Morgan, a well-known mathematician who had made significant contributions to both algebra and logic. Ada had a very high opinion of her own abilities and wrote to Babbage that "the more I study, the more insatiable do I feel my genius for it to be."[13] Her opinion of her talent is supported, in part at least, by a letter written by De Morgan to her mother. In the letter, De Morgan suggested that Ada's abilities could lead her to become "an original mathematical investigator, perhaps of first-rate eminence."[14]

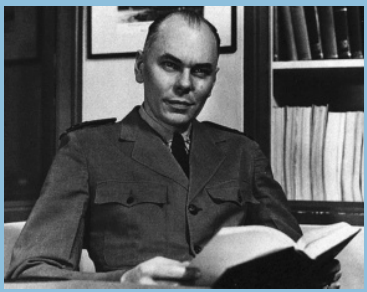
At the suggestion of a mutual friend, the scientist Charles Wheatstone, Lovelace translated Menabrea's paper for publication in English. Babbage then suggested that she add some notes of her own to the paper. He took a great interest in her work and gave her the material and examples he had used in his talk in Turin and helped her by annotating drafts of her notes. Babbage also wrote a completely new example for her: the calculation of the Bernoulli numbers (a complex sequence of rational numbers) using the Analytical Engine. Although Ada did not originate this example, she clearly understood the procedure well enough to point out a mistake in Babbage's calculation. She both amplified Babbage's ideas and expressed them in her own forthright manner, as is evident from these two examples from her notes:

> The distinctive characteristic of the Analytical Engine, and that which has rendered it possible to endow mechanism with such extensive faculties as bid fair to make this engine the executive right-hand of abstract algebra, is the introduction into it of the principle which Jacquard devised for regulating, by means of punched cards, the most complicated patterns in the fabrication of brocaded stuffs. It is in this that the distinction between the two engines lies. Nothing of the sort exists in the Difference Engine. We may say most aptly that the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves. . . .
>
> Many persons … imagine that because the business of the Engine is to give its results in numerical notation the nature of its processes must consequently be arithmetical and numerical, rather than algebraical and analytical. This is an error. The engine can arrange and combine its numerical quantities exactly as if they were letters or any other general symbols; and in fact it might bring out its results in algebraic notation, were provisions made accordingly.[15]

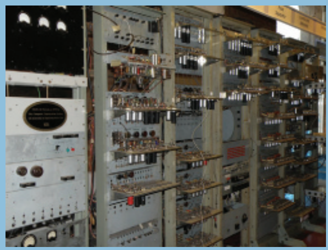
Babbage certainly had not published or developed the idea of using his machine for algebra in any detail. One remark from Lovelace is also often quoted in debates about artificial intelligence and computers: "The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform."[16] We will look later at the question of whether computers are capable of recognizable intelligence.

B.1.6 A portrait of Ada Lovelace (1815–52) drawn by a robotic artist. Her father, the Romantic poet Lord Byron, was instrumental in ensuring that she was educated in mathematics. She was the first to write in English about the potential capabilities of Babbage's Analytical Engine and is considered by some to be the first "computer programmer." She was certainly the first to emphasize that the machine could manipulate symbols as well as perform numerical calculations. She also wrote that perhaps one day machines would even be able to write poetry.

TL.1.1. John Vincent Atanasoff (1903–95) with Clifford Berry; they constructed ABC using vacuum tubes.



TL.1.2. Howard Aiken (1900–73), the constant clicking of relays created a sound as if the "room was full of knitting ladies."



TL.1.3. The first stored program computer. The memory was constructed from Cathode Ray Tubes.

ABC
1936

MARK-I
1944

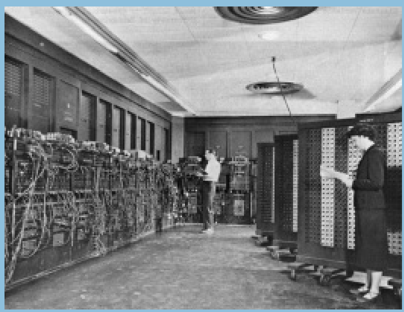
Manchester Baby
1948

Z1
1934

ENIAC
1945

EDSAC
1949

TL.1.4 Konrad Zuse tinkering with his Z-computer.



TL.1.5 Rewiring the ENIAC was a challenging task.

TL.1.6 Maurice Wilkes checking the valves of the EDSAC computer.

TL.1.7 The first stored-program computer constructed in Australia used a mercury line memory and could generate music.

TL.1.8 LEO, a successful business computer, used by a chain of Lyon's tea shops.

TL.1.9 The first Soviet computer was built in a monastic hostel near Kiev.

CSIR Mark I
1949

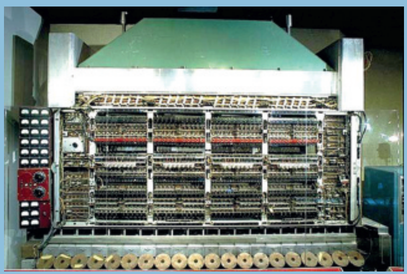
LEO
1951

MESM
1951

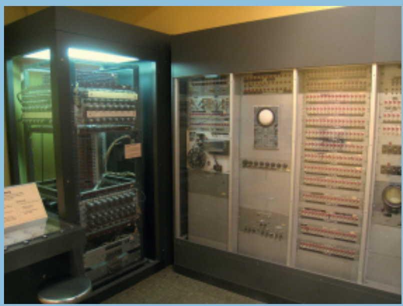
IAS
1952

Whirlwind
1951

UNIVAC
1952



TL.1.10 The IAS machine was a prototype for many computers because the design was not patented and was widely disseminated. The programs running on the computer were mainly calculations for the H-bomb, but also biological simulations.



TL.1.11 The first flight simulator computer was used for training bomber crews.

TL.1.12 Presper Ekert (center) demonstrating the UNIVAC to CBS reporter Walter Cronkite. The machine was used to predict the results of the 1952 U.S. election, but even the programmers did not believe their (correct) prediction, made after only 7% of the vote was in: a landslide win for Eisenhower instead of the close election predicted by the pollsters.

Code breakers and bread makers

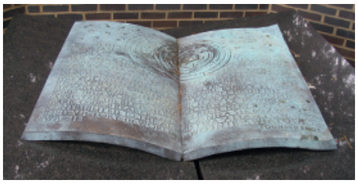


Fig. 1.19 Memorial to Polish code breakers at Bletchley Park. Their contribution was critical to the development of the Bombe machines used to break the Enigma codes.

No history of the early days of computing would be complete without recounting the pioneering work of the British cryptologists at Bletchley Park and the development of the first computer dedicated to business use.

Bletchley Park, Enigma, and Colossus

During World War II, British mathematicians and scientists had been looking to automated machines for their top-secret work on code breaking at Bletchley Park. Both Turing and his Cambridge mentor Max Newman (B.1.7) were intimately involved in designing and building automated machines to assist in decrypting secret German military communications. Turing was involved in deciphering messages encrypted using the famous Enigma machine. With a head start given to them by Polish Intelligence (Fig. 1.19), Turing helped develop electromechanical machines, known as *bombes*, which were used to determine the settings of the Enigma machine. These machines were operational as early as 1940 and contributed greatly to protecting convoys from U-boats in the North Atlantic.

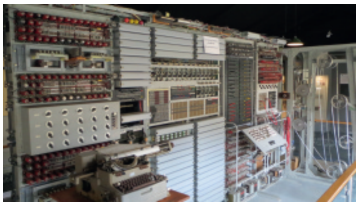
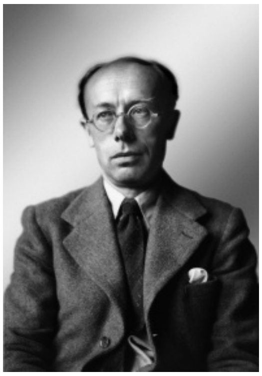


Fig. 1.20 A photograph of the Colossus computer, the code-breaking machine that nobody knew existed until many years after the war. It was designed and built by Tommy Flowers, an engineer at the British Post Office in 1943.

The German High Command in Berlin used an even more complex cipher machine called Lorenz. Newman's team built a machine – called Heath Robinson after a popular cartoonist who drew eccentric machines – that showed it was possible to make a device to break the Lorenz codes. This was followed by the ULTRA project to build an all-electronic version of Heath Robinson called Colossus (Fig. 1.20). Although this machine was certainly not a general-purpose computer, it had 1,500 vacuum tubes as well as tape readers with optical sensors capable of processing five thousand teleprinter characters a second. The machine was designed and built by Tommy Flowers (B.1.8), an engineer at the U.K. Post Office's Dollis Hill research laboratory in London, and became operational in December 1943, more than two years before the ENIAC. One of the great achievements of Colossus was reassuring the Allied generals, Eisenhower and Montgomery, that Hitler had believed the deception that the D-Day invasion fleet would come from Dover. The immense contribution of code breakers was recognized by Winston Churchill when he talked about "the Geese that laid the golden eggs but never cackled."[17]



B.1.7 Max Newman (1897–1984) was a brilliant Cambridge, U.K., mathematician and code breaker. It was Newman's lectures in Cambridge that inspired Alan Turing to invent his famous Turing Machine. Newman was at Bletchley Park during World War II and his team was working on the messages encrypted by the Lorenz cipher machine. They built a machine – called Heath Robinson – to break the Lorenz code, and this was later automated as the Colossus computer.

The main task for the code breakers was to read the text from a paper tape and to work out the possible settings of the twelve rotors of the encrypting device. Colossus was first demonstrated in December 1943

B.1.8. The name of Tommy Flowers (1905–98) is virtually unknown to most students of computing. His immense contribution to computing and code breaking during the war has only recently emerged from the obsessive secrecy imposed on the code-breaking activities at Bletchley Park after World War II. Flowers built an electronic code-breaking machine called Colossus, which was capable of breaking the so-called Lorenz cipher used by the German high command. Instead of using electromechanical devices as in the bombes used for breaking the Enigma codes, Flowers decided to use vacuum tubes. This idea initially met with some resistance because it was generally thought that tubes would not be sufficiently reliable. Colossus contained about one and a half thousand vacuum tubes and was the world's first special-purpose electronic computer. Flowers described the heat generated by the computer with the following words: "Ah, the warmth at two a.m. on a damp, cold English winter!"[B3]

and provided invaluable information for preparing the Normandy landing. The automated code-breaking devices such as Colossus and the bombes made a significant contribution to shortening the war (Fig. 1.21).

At the end of the war, Winston Churchill gave orders that most of the ten Colossus machines should be destroyed. Flowers personally burned the blueprints in a furnace at the Dollis Hill laboratory. The reason for this destruction was so that the British government could promote and sell Enigma-like machines to other governments yet still retain the ability to decipher messages sent using the machines! Two Colossus machines were in use at the UK Government Communication Headquarters (GCHQ) in Cheltenham until the late 1950s. With the coming of digital communications, the need for such secrecy about the wartime activities at Bletchley Park became unnecessary, and information about Colossus began to emerge in the 1970s. A secret 1945 report on the decrypting of the Lorenz signals was declassified in 2000 and contains the following description of working with Colossus:

> It is regretted that it is not possible to give an adequate idea of the fascination of a Colossus at work; its sheer bulk and apparent complexity; the fantastic speed of thin paper tape round the glittering pulleys; … the wizardry of purely mechanical decoding letter by letter (one novice thought she was being hoaxed); the uncanny action of the typewriter in printing the correct scores without and beyond human aid. . . ."[18]

One clear result of this U.K. obsession for secrecy about its achievements in computer development during the war years was that all subsequent computer developments, even in the United Kingdom, were based on von Neumann's EDVAC design for stored-program computers.

Fig. 1.21. "We Also Served" – a memorial to code breakers at Bletchley Park. On the back of the memorial is a quote from Winston Churchill, written in Morse code: "My Most Secret Source."

LEO: The first business computer

A curious footnote to the EDSAC is the development of Lyons Electronic Office, or LEO, the world's first computer specifically designed for business applications rather than for numerical calculations. The unlikely business was that of J. Lyons & Co., which ran a nationwide chain of Lyons Tea Shops as well as Lyons Corner Houses (Fig. 1.22). These offered English high teas and cream cakes in London and featured uniformed waitresses called "Nippies." The catering business required an army of clerks to ensure that the correct quantity of baked goods was delivered fresh every day and to process the associated receipts and invoices. It seems obvious to us now that such jobs can be computerized, but at the time it required real vision to recognize that a computer originally designed to calculate trajectories of shells could be useful for nonscientific business applications.

Fig. 1.22 Lyons operated a network of tea shops and "corner houses" throughout the United Kingdom and, surprisingly, pioneered the use of computers for business calculations.

B.1.9 John Pinkerton (1919–97), one of the first computer engineers, pictured in front of the LEO, the first business computer that he designed and built as a modified version of the EDSAC. The machine was built for J. Lyons & Co. to automate the record keeping for the production and delivery of their baked goods to their famous Lyons Tea Shops. LEO, Lyons Electronic Office, went into operation in 1951.

The LEO project was the vision of John Simmons, a mathematician in the Organization and Methods Department of Lyons who was an enthusiastic advocate for automation with computers. After sending a team to the United States in May 1947, Simmons realized that, with Maurice Wilkes on their doorstep in Cambridge, he had a local university partner who could help his company find a solution to its problems. In May 1949, after the EDSAC was clearly shown to work, Simmons was given the go-ahead to build LEO. The engineer appointed to design the machine was John Pinkerton (B.1.9), arguably the first industrial computer engineer. He assembled a small very talented team and ensured that their modifications of the EDSAC design took account of the unusual user requirements. Their type of business jobs was very different from scientific calculations, which typically had very little input and output and ran for a long time between operations. The LEO machine had multiple input and output channels and much more memory than the EDSAC. In addition, because reliability was a critical concern for running day-to-day business applications, Pinkerton designed the machine with 28 interchangeable vacuum tube units so that defective units could be swapped out without delay. The machine was ready in late 1951, and LEO took over "bakery valuations" from the clerical staff in November. The machine calculated the value of the bakery output in terms of bread, cakes, and pies from the bakery input of materials, labor, and power costs. It used the factory costs with the prices and profit margins to calculate the value of the products distributed to the teashops, grocers, and restaurants. LEO also calculated the value of products kept in stock. The LEO Computers Company was formed in 1954 and delivered upgrades of the machine until the early 1960s, when the company merged with English Electric. In 1968 this company formed the foundation for a new British computer company called International Computers Ltd., which operated profitably for several decades.

Other Beginnings

Computer development was not limited to the United States and the United Kingdom; other countries, including Germany, the Soviet Union, and Australia, also pioneered the development of digital electronic computers.

Konrad Zuse, the Z series, and Plankalkül

In Germany, Konrad Zuse (B.1.10) is now widely acknowledged as one of the founding fathers of computing. He worked on his designs in isolation during the difficult times leading up to the war years. In 1941, his first operational electromechanical computer, the Z3, contained some architectural features that, independently of Zuse's work, have become key principles for computer designers. Unlike the ENIAC, Zuse's machine used binary encoding for both data and instructions, significantly reducing the complexity of the design. The Z3 was the first automatic, program-controlled, relay computer, predating Howard Aiken's Mark I machine. By 1943, Zuse was constructing a new Z4 computer. At the height of the wartime bombing of Germany, Zuse managed to transport the Z4 from Berlin and hide it in a stable in a small village in the Bavarian Alps. Zuse also developed a programming language called Plankalkül (plan calculus). In this language he introduced the concept of assignment and loops that are now seen as key components of programming languages. After the war, in 1949, Zuse founded a computer company, Zuse KG; he sold his first Z4 to ETH in Zurich in 1950. His company continued to develop new machines; he sold 56 of his Z22 vacuum tube machines for industrial and university research. The company was bought by Siemens in 1967.

B.1.10 Konrad Zuse (1910–95) independently designed and constructed computers during World War II. Until recently, his pioneering work was not widely known although IBM had taken an option on his patents in 1946. His programming language, Plankalkül, was never implemented in Zuse's lifetime. A team from the Free University of Berlin produced a compiler for the language in 2001.

Sergei Lebedev, MESM, and BESM

Sergei Alekseyevich Lebedev (B.1.11) was one of the pioneers of Soviet computing. Under his leadership after the war, a secret electronic laboratory was established in the outskirts of Kiev, where he and his team started to build the first Soviet computers. By December 1951, they had a functioning machine, and this marked the beginning of indigenous Soviet computers. They produced computers that ranged from large mainframe computers of the classes BESM, URAL, and Elbrus to smaller machines such as MIR and MESM. These names are largely unknown outside of Russia, but in scientific and engineering circles behind the Iron Curtain they were held in great respect. The BESM computers formed the backbone of Soviet computing; about 350 were produced. BESM-1 was built in 1953, and the last of this series, BESM-6, in 1966. However, in 1967 a political decision was taken to copy IBM machines. This was the end of indigenous Soviet computing and a bitter disappointment for many of the Soviet computer pioneers.

B.1.11 Sergei Alekseyevich Lebedev (1902–74) was the founder of the Soviet computer industry. A keen alpinist as well as a brilliant engineer, he climbed Europe's highest peak, Mount Elbrus. In 1996 he was posthumously awarded the Charles Babbage medal by the IEEE society. His name in Cyrillic script is written Сергей Алексеевич Лебедев.

Few specifics are known about the Soviet machines, since most of the documents were never published. There is a general perception that computing in the Soviet Union in the 1950s and 1960s was far less developed than in the West. However, it is hard to imagine that the Soviets would have been able to achieve the spectacular results in space exploration, defense, and technology without possessing some serious computing capacity. Doron Swade, a senior curator of the London Science Museum, traveled to Siberia in 1992 to procure a Soviet BESM-6 computer for his museum's collection. In an interview for BBC Radio 4, he was asked about Lebedev's contribution and the MESM computer:

> Was MESM original? I would say almost completely yes. Was its performance comparable? Certainly. Was BESM's performance comparable? I'd say BESM by that stage was being outperformed by the equivalent generation in the [United] States. But as a workhorse, as an influential machine in the plenty of Russian computer science in terms of its utility and its usefulness to the space program, to military research and scientific research it is probably, arguably the most influential machine in the history of modern computing.[19]

Trevor Pearcey and the CSIR Mark I

Trevor Pearcey (B.1.12) was born in the United Kingdom and had worked on applying advanced mathematics to radar development during World War II. In 1945, he emigrated to Australia and visited Harvard and MIT on the way; he saw both Aiken's Mark I and Bush's Differential Analyzer in operation. By 1946 Pearcey was working at the Division of Radiophysics of the Australian Council for Scientific and Industrial Research (CSIR) located at the University of Sydney. He understood the limitations of the machines he had seen in the United States and saw the potential for using vacuum tubes to create a high-speed digital computer. By the end of 1947, Pearcey, working on the theory, and Maston Beard, an electrical engineering graduate from Sydney working on the hardware, had defined their design. Although Pearcey visited the United Kingdom near the end of 1948 and saw the Manchester Baby and the Cambridge EDSAC, he saw no reason to change his original design. He later asserted that the CSIR Mark I "was completely 'home-grown' some 10,000 miles distant from the mainstream development in the UK and USA."[20] As with all the early computers, the development of computer memory technology was one of the major challenges. It was left to engineer Reg Ryan on the Australian team to design the memory system for the CSIR Mark I using mercury delay lines. The machine operated at 1 kilohertz and its delay line memory could store 768 words, each 20 bits long. By the end of 1949, their computer was able to run some basic mathematical operations and could genuinely claim to be one of the first operational stored-program computers. By 1951, CSIR had changed its name to the Commonwealth Scientific and Industrial Research Organisation, and the computer became the CSIRO Mark I. At Australia's first conference on Automatic Computing Machines in August 1951, the Mark I gave the first demonstration of computer-generated music by playing the popular wartime song "Colonel Bogey." In 1954, the CSIRO project was officially ended, and the machine was transferred to the University of Melbourne in 1955. The university's new Computation Laboratory was opened in 1956 with the CSIRO machine as its workhorse, rechristened CSIRAC. The machine ran for the next eight years, with only about 10 percent of its running time taken up for maintenance.

B.1.12 Trevor Pearcey (1919–98) was born in London and graduated from Imperial College with a degree in physics and mathematics. After working on radar systems during the war, he emigrated to Australia and was responsible for designing and building the CSIR Mark I at the University of Sydney. This was one of the world's first computers to use vacuum tubes.