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 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.

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.

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.

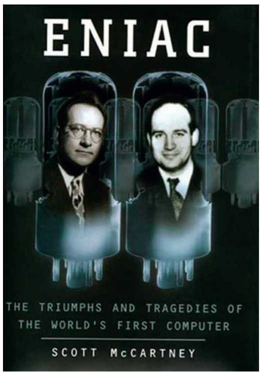
Goldstine was a mathematician by training, not an engineer, and so was not aware of the generally accepted wisdom that building a large-scale computer with many thousands of vacuum tubes was considered impossible because of the tubes' intrinsic unreliability. After talking with Mauchly, Goldstine asked 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.

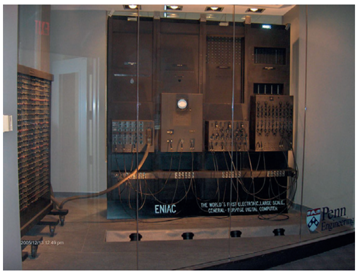


Fig. 1.4 A section of the original ENIAC machine on display at the University of Pennsylvania.

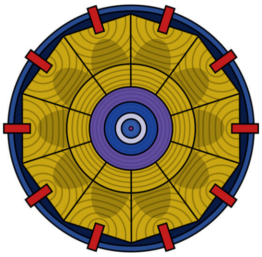


Fig. 1.5 A schematic diagram of the spherical implosion lens required to start the nuclear reaction in a plutonium bomb. John von Neumann's search for an automatic device that would speed up the complex calculations needed to model the lens led to his interest in ENIAC.

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

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

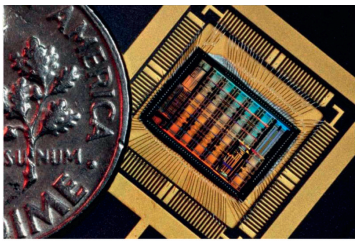
B.1.2 John von Neumann (1903–57) was born in Budapest in the family of a wealthy banker. After graduating with a PhD in mathematics from Budapest ELTE and a diploma in chemical engineering from Zurich ETH, he won a scholarship in Gottingen and worked with David Hilbert on his ambitious program on the "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]



Packaged Eniac-on-a-Chip (photo by Felice Macera).

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

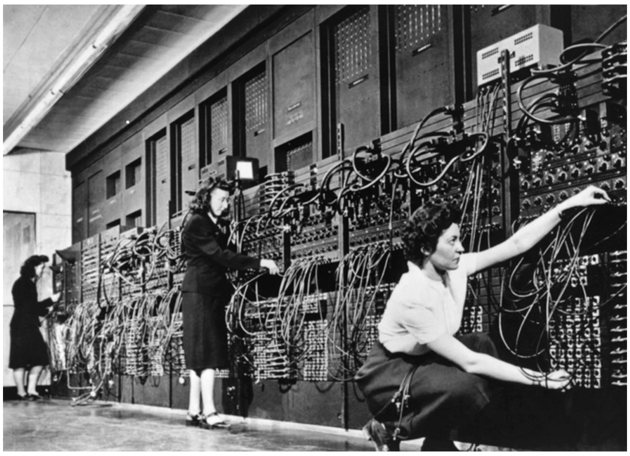


Fig. 1.8 The first programmers of ENIAC were women. In those days, programming meant setting all switches and rewiring the computer, a tedious operation that often took days to complete.

"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.

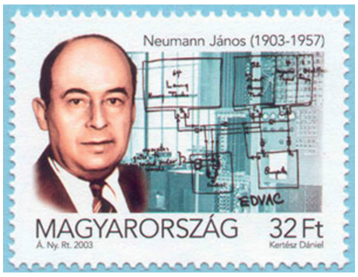


Fig. 1.9 A Hungarian postage stamp that honors John von Neumann, complete with the mathematician's likeness and a sketch of his computer architecture.

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

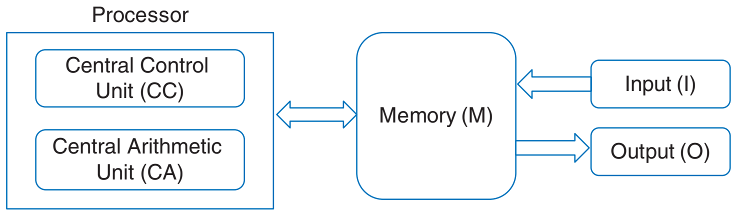
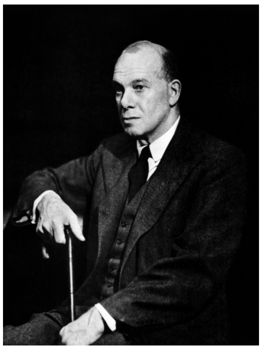


Fig. 1.10 The von Neumann Architecture. The main building blocks of all computers are the input, output, memory, and processor. The input (typically now a keyboard or a mouse) feeds data into the computer. This information is encoded by binary numbers and stored in the memory. The processor then fetches 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 long-time 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

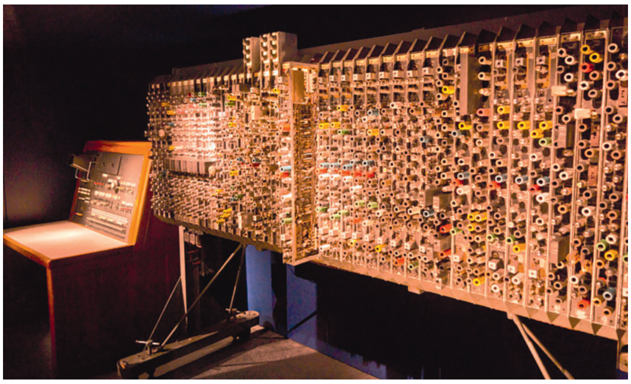
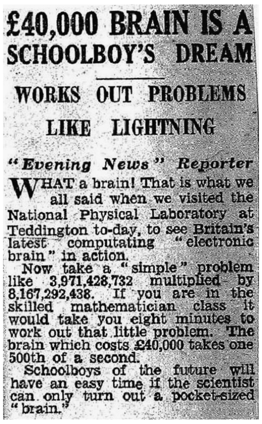


Fig. 1.11 The Pilot ACE was a computer with a distinctive flavor. Turing's design was much more detailed than that contained in von Neumann's EDVAC report published only three months earlier. Pilot ACE had many innovative features, such as three address instructions, variable-length block transfer, and 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

> 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"!

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

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.