History of computer science

The **history of computer science** began long before the modern discipline of computer science, usually appearing in forms like mathematics or physics. Developments in previous centuries alluded to the discipline that we now know as computer science.[1] This progression, from mechanical inventions and mathematical theories towards modern computer concepts and machines, led to the development of a major academic field, massive technological advancement across the Western world, and the basis of a massive worldwide trade and culture.[2]

Prelamary

The earliest known tool for use in computation was the abacus, developed in the period between 2700–2300 BCE in Sumer.[3] The Sumerians' abacus consisted of a table of successive columns which delimited the successive orders of magnitude of their sexagesimal number system.[4] Its original style of usage was by lines drawn in sand with pebbles. Abaci of a more modern design are still used as calculation tools today, such as the Chinese abacus.[5]

In the 5th century BC in ancient India, the grammarian Pāṇini formulated the grammar of Sanskrit in 3959 rules known as the Ashtadhyayi which was highly systematized and technical. Pāṇini used metarules, transformations and recursions.[6]

The Antikythera mechanism is believed to be an early mechanical analog computer.[7] It was designed to calculate astronomical positions. It was discovered in 1901 in the Antikythera wreck off the Greek island of Antikythera, between Kythera and Crete, and has been dated to circa 100 BC.[8]

Mechanical analog computer devices appeared again a thousand years later in the medieval Islamic world and were developed by Muslim astronomers, such as the mechanical geared astrolabe by Abū Rayḥān al-Bīrūnī,[9] and the torquetum by Jabir ibn Aflah.[10] According to Simon Singh, Muslim mathematicians also made important advances in cryptography, such as the development of cryptanalysis and frequency analysis by Alkindus.[11] Programmable machines were also invented by Muslim engineers, such as the automatic flute player by the Banū Mūsā brothers.[12]

During the Middle Ages, several European philosophers made attempts to produce analog computer devices. Influenced by the Arabs and Scholasticism, Majorcan philosopher Ramon Llull (1232–1315) devoted a great part of his life to defining and designing several logical machines that, by combining simple and undeniable truths by simple logical operations, produced by the machine by mechanical means, in such ways as to produce all possible knowledge.[13] Llull's machine was a very important development in the history of computer science.[14]

In 1492, Leonardo da Vinci developed cog-wheels he modeled in a mechanical calculator.[citation needed] Some consider this a late beginning of the history of computer hardware.[15]

Technological advances of similar computing machines were made around the same time. In 1623 Wilhelm Schickard invented a calculating machine.[16]

When John Napier discovered logarithms for computational purposes in the early 17th century,[citation needed] there followed a period of considerable progress by inventors and scientists in making calculating tools. In 1623 Wilhelm Schickard designed a calculating machine which he termed the Calculating Clock, of which a modern replica has been built, but abandoned the project when the prototype he had started building was destroyed by a fire in 1624.[17] Around 1640, Blaise Pascal, a leading French mathematician, constructed a mechanical adding device based on a design described by Greek mathematician Hero of Alexandria.[18] Then in 1672 Gottfried Wilhelm Leibniz invented the Stepped Reckoner which he completed in 1694.[19]

In 1837 Charles Babbage first described his Analytical Engine which is accepted as the first design for a modern computer. The analytical engine had expandable memory, an arithmetic unit, and logic processing capabilities that enabled it to interpret a programming language with loops and conditional branching. Although never built, the design has been studied extensively and is understood to be Turing equivalent. The analytical engine would have had a memory capacity of less than 1 kilobyte of memory and a clock speed of less than 10 Hertz.[20]

Considerable advancement in mathematics and electronics theory was required before the first modern computers could be designed.

Binary logic

In 1702, Gottfried Wilhelm Leibniz developed logic in a formal, mathematical sense with his writings on the binary numeral system. In his system, the ones and zeros also represent true and false values or on and off states. But it took more than a century before George Boole published his Boolean algebra in 1854 with a complete system that allowed computational processes to be mathematically modeled.[21]

By this time, the first mechanical devices driven by a binary pattern had been invented. The Industrial Revolution had driven forward the mechanization of many tasks, and this included weaving. Punched cards controlled Joseph Marie Jacquard's loom in 1801, where a hole punched in the card indicated a binary one and an unpunched spot indicated a binary zero. Jacquard's loom was far from being a computer, but it did illustrate that machines could be driven by binary systems.[21]

Emergence of a discipline

Charles Babbage and Ada Lovelace

Charles Babbage is often regarded as one of the first pioneers of computing. Beginning in the 1810s, Babbage had a vision of mechanically computing numbers and tables. Putting this into reality, Babbage designed a calculator to compute numbers up to 8 decimal points long. Continuing with the success of this idea, Babbage worked to develop a machine that could compute numbers with up to 20 decimal places. By the 1830s, Babbage had devised a plan to develop a machine that could use punched cards to perform arithmetical operations. The machine would store numbers in memory units, and there would be a form of sequential control. This means that one operation would be carried out before another in such a way that the machine would produce an answer and not fail. This machine was to be known as the "Analytical Engine", which was the first true representation of what is the modern computer.[22]

Ada Lovelace (Augusta Ada Byron) is credited as the pioneer of computer programming and is regarded as a mathematical genius. Lovelace began working with Charles Babbage as an assistant while Babbage was working on his "Analytical Engine", the first mechanical computer.[23] During her work with Babbage, Ada Lovelace became the designer of the first computer algorithm, which had the ability to compute Bernoulli numbers.[24] Moreover, Lovelace's work with Babbage resulted in her prediction of future computers to not only perform mathematical calculations, but also manipulate symbols, mathematical or not.[25] While she was never able to see the results of her work, as the "Analytical Engine" was not created in her lifetime, her efforts in later years, beginning in the 1840s, did not go unnoticed.[26]

Early post-industrial Revolution designs

Following Babbage, although at times unaware of his earlier work, others designed similar machines. Leonardo Torres Quevedo, in his work Essays on Automatics (1913), formalized the concept of a programmable mechanical computer, which he described in a work published in 1914.[27]

Two other inventors, Leonardo Torres Quevedo and Vannevar Bush, also furthered research in computing. Torres introduced the Electromechanical Arithmometer which was controlled by a read-only program and consisted of many gears.[28] In 1920, to celebrate the 100th anniversary of the invention of the arithmometer, Torres presented in Paris the Electromechanical Arithmometer, which allowed a user to input arithmetic problems through a keyboard, and computed and printed the results,[29] demonstrating the feasibility of an electromechanical analytical engine.[30] Bush's work in analog computing was quite influential, and in 1938 he published a paper describing the use of electrical circuits to do Boolean logic.[31]

Charles Sanders Peirce and electrical switching circuits

In an 1886 letter, Charles Sanders Peirce described how logical operations could be carried out by electrical switching circuits.[32] During 1880–81 he showed that NOR gates alone (or alternatively NAND gates alone) can be used to reproduce the functions of all the other logic gates, but this work on it was unpublished until 1933.[33] The first published proof was by Henry M. Sheffer in 1913, so the NAND logical operation is sometimes called Sheffer stroke; the logical NOR is sometimes called Peirce's arrow.[34] Consequently, these gates are sometimes called universal logic gates.[35]

Eventually, vacuum tubes replaced relays for logic operations. Lee De Forest's modification, in 1907, of the Fleming valve can be used as a logic gate. Ludwig Wittgenstein introduced a version of the 16-row truth table as proposition 5.101 of Tractatus Logico-Philosophicus (1921). Walther Bothe, inventor of the coincidence circuit, got part of the 1954 Nobel Prize in physics, for the first modern electronic AND gate in 1924. Konrad Zuse designed and built electromechanical logic gates for his computer Z1 (from 1935 to 1938).

Up to and during the 1930s, electrical engineers were able to build electronic circuits to solve mathematical and logic problems, but most did so in an ad hoc manner, lacking any theoretical rigor. This changed with switching circuit theory in the 1930s. From 1934 to 1936, Akira Nakashima, Claude Shannon, and Victor Shestakov published a series of papers showing that the two-valued Boolean algebra, which can describe the operation of switching circuits.[36][37][38][39]

This concept, of utilizing the properties of electrical switches to do logic, is the basic concept that underlies all electronic digital computers, its theoretical practical foundations laid in switching circuit theory. Shannon's thesis became the foundation of practical digital circuit design when it became widely known among the electrical engineering community during and after World War II.

Alan Turing and the Turing machine

Before the 1920s, computers (sometimes computors) were human clerks that performed computations. They were usually under the lead of a physicist. Many thousands of computers were employed in commerce, government, and research establishments. Many of these clerks who served as human computers were women.[40][41][42] Most were bright and well-educated graduates holding degrees in mathematics.[43] Some did astronomical calculations for calendars, others ballistics tables for the military.[43]

After the 1920s, the expression computing machine referred to any machine that performed the work of a human computer, especially those in accordance with effective methods of the Church-Turing thesis. The thesis states that a mathematical method is effective if it could be set out as a list of instructions able to be followed by a human clerk with paper and pencil, for as long as necessary, and without ingenuity or insight.[44]

Machines that computed with continuous values became known as the analog kind. They used machinery that represented continuous numeric quantities, like the angle of a shaft rotation or difference in electrical potential.[45]

Digital machinery, in contrast to analog, was able to render a state of a numeric value and store each digit separately. Digital machinery used difference engines or relays before the invention of faster memory devices.[46]

The phrase computing machine gradually gave way, after the late 1940s, to just computer as the onset of electronic digital machinery became common. These computers were able to perform the calculations that were performed by the previous human clerks.

Since the values stored by digital machines were not bound to physical properties like analog devices, a logical computer, based on digital equipment, was able to do anything that could be described "purely mechanical." The theoretical Turing Machine, created by Alan Turing, is a hypothetical device theorized in order to study the properties of such hardware.[47]

The mathematical foundations of modern computer science began to be laid by Kurt Gödel with his incompleteness theorem (1931). In this theorem, he showed that there were limits to what could be proved and disproved withi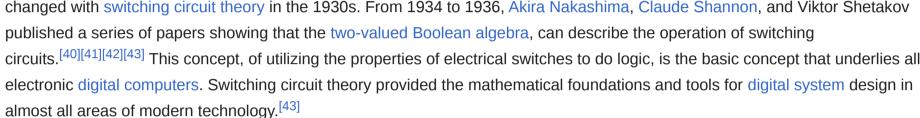n a formal system. This led to work by Gödel and others to define and describe these formal systems, including concepts such as mu-recursive functions and lambda-definable functions.[48]

In 1936 Alan Turing and Alonzo Church independently, and also together, introduced the formalization of an algorithm, with limits on what can be computed, and a "purely mechanical" model for computing.[49] This became the Church-Turing thesis, a hypothesis about the nature of mechanical calculation devices, such as electronic computers. The thesis states that any calculation that is possible can be performed by an algorithm running on a computer, provided that sufficient time and storage space are available.[49]

In 1936, Alan Turing also published his seminal work on the Turing machines, an abstract digital computing machine which is now simply referred to as the Universal Turing machine. This machine invented the principle of the modern computer and was the birthplace of the stored program concept that almost all modern day computers use.[50] These hypothetical machines were designed to formally determine, mathematically, what can be computed, taking into account limitations on computing ability. If a Turing machine can complete the task, it is considered Turing computable.[51]

The Los Alamos physicist Stanley Frankel, has described John von Neumann's view on the fundamental importance of Turing's 1936 paper, in a letter:[52]

> I know that in about 1943 or '44 von Neumann was well aware of the fundamental importance of Turing's paper of 1936 ... Von Neumann introduced me to that paper and at his urging I studied it with care. Many people have acclaimed von Neumann as the "father of the computer" (in a modern sense of the term) but I am sure that he would never have made that mistake himself. He might well be called the midwife, perhaps, but he firmly emphasized to me, and to others I am sure, that the fundamental conception is owing to Turing— in so far as not anticipated by Babbage ... Both Turing and von Neumann, of course, also made substantial contributions to the "reduction to practice" of these concepts but I would not regard these as comparable in importance with the introduction and explication of the concept of a computer able to store in its memory its program of activities and of modifying that program in the course of these activities.

Kathleen Booth and the first assembly language

Kathleen Booth wrote the first assembly language and designed the assembler and autocode for the Automatic Relay Calculator (ARC2) at Birkbeck, University of London.[53] She he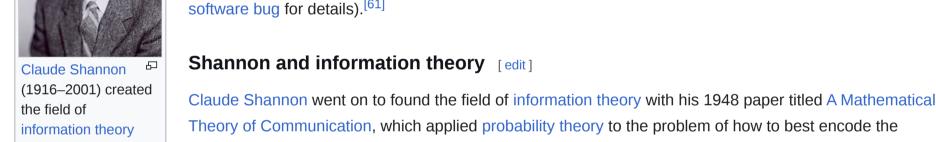lped design three different machines including the ARC (Automatic Relay Calculator), the SEC (Simple Electronic Computer), and APEXC.[53]

Early computer hardware

The world's first electronic digital computer, the Atanasoff–Berry computer, was built on the Iowa State campus from 1939 through 1942 by John V. Atanasoff, a professor of physics and mathematics, and Clifford Berry, an engineering graduate student.

In 1941, Konrad Zuse developed the world's first functional program-controlled computer, the Z3. In 1998, it was shown to be Turing-complete in principle.[54] Zuse also developed the S2 computing machine, considered the first process-control computer. He founded one of the earliest computer businesses in 1941, producing the Z4, which became the world's first commercial computer. In 1946, he designed the first high-level programming language, Plankalkül.[55]

In 1948, the Manchester Baby was completed; it was the first electronic digital computer that ran programs stored in its memory, like almost all modern computers.[56] The influence on Max Newman of Turing's seminal 1936 paper on the Turing Machines and of his logico-mathematical contributions to the project, were both crucial to the successful development of the Manchester Baby.[57]

In 1950, Britain's National Physical Laboratory completed Pilot ACE, a small scale programmable computer, based on Turing's philosophy. With an operating speed of 1 MHz, the Pilot Model ACE was for some time the fastest computer in the world.[57][58] Turing's design for ACE had much in common with today's RISC architectures and it called for a high-speed memory of roughly the same capacity as an early Macintosh computer, which was enormous by the standards of his day.[57] Had Turing's ACE been built as planned and in full, it would have been in a different league from the other early computers.[57]

The first actual computer bug was a moth. It was stuck in between the relays on the Harvard Mark II.[59] While the invention of the term 'bug' is often but erroneously attributed to Grace Hopper, a future rear admiral in the U.S. Navy, who supposedly logged the "bug" on September 9, 1947, most other accounts conflict at least with these details. According to these accounts, the actual date was September 9, 1947 when operators filed this 'incident' — along with the insect and the notation "First actual case of bug being found" (see software bug for details).[59]

Shannon and information theory

Claude Shannon went on to found the field of information theory with his 1948 paper titled A Mathematical Theory of Communication, which applied probability theory to the problem of how to best encode the information a sender wants to transmit. This work is one of the theoretical foundations for many areas of study, including data compression and cryptography.[60]

Wiener and cybernetics

From experiments with anti-aircraft systems that interpreted radar images to detect enemy planes, Norbert Wiener coined the term cybernetics from the Greek word for "steersman." He published "Cybernetics" in 1948, which influenced artificial intelligence. Wiener also compared computation, computing machinery, memory devices, and other cognitive similarities with his analysis of brain waves.[61]

John von Neumann and the von Neumann architecture

In 1946, a model for computer architecture was introduced and became known as Von Neumann architecture. Since 1950, the von Neumann model provided uniformity in subsequent computer designs. The von Neumann architecture was considered innovative as it introduced an idea of allowing machine instructions and data to share memory space.[62] The von Neumann model is composed of three major parts, the arithmetic logic unit (ALU), the memory, and the instruction processing unit (IPU). In von Neumann machine design, the IPU passes addresses to memory, and memory, in turn, is routed either back to the IPU if an instruction is being fetched or to the ALU if data is being fetched.[63]

Von Neumann's machine design uses a RISC (Reduced instruction set computing) architecture,[a] which means the instruction set uses a total of 21 instructions to perform all tasks. (This is in contrast to CISC, complex instruction set computing, instruction sets which have more instructions from which to choose.) With von Neumann architecture, main memory along with the accumulator (the register that holds the result of logical operations)[64] are the two memories that are addressed. Operations can be carried out as simple arithmetic (these are performed by the ALU and include addition, subtraction, multiplication and division), conditional branches (these are more commonly seen now as if statements or while loops; the branches serve as go to statements), and logical moves between the different components of the machine, i.e., a move from the accumulator to memory or vice versa. Von Neumann architecture accepts fractions and instructions as data types. Finally, as the von Neumann architecture is a simple one, its register management is also simple. The architecture uses a set of seven registers to manipulate and interpret fetched data and instructions. These registers include the "IR" (instruction register), "IBR" (instruction buffer register), "MQ" (multiplier quotient register), "MAR" (memory address register), and "MDR" (memory data register).[63] The architecture also uses a program counter ("PC") to keep track of where in the program the machine is.[63]

John McCarthy, Marvin Minsky and artificial intelligence

The term artificial intelligence was credited by John McCarthy to explain the research that they were doing for a proposal for the Dartmouth Summer Research. The naming of artificial intelligence also led to the birth of a new field in computer science.[65] On August 31, 1955, a research project was proposed consisting of John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. The official project began in 1956 that consisted of several significant parts they felt would help them better understand artificial intelligence's makeup.[66]

McCarthy and his colleagues' ideas behind automatic computers were that while a machine may be capable of completing a task, the same would not be confirmed to be completed until tested. Machines should also be capable of reasoning; this is because a machine is capable of interpreting information. Under this assumption McCarthy and his colleagues realized that the best way to manufacture a computer that displayed these properties was for the machine itself be the programmer. The knowledge to produce a program that demonstrated these properties was the main goal.[66]

The concept behind this was looking at how humans understand our own language and structure of how we form sentences, giving different meaning and rule sets and comparing them to a machine process. The way computers can understand is at a hardware level. This language is written in binary (1s and 0s). This has to be written in a specific format that gives the computer the rule set to run a particular piece of hardware.[66]

Minsky's process determined how these artificial neural networks could be arranged to have similar qualities to the human brain. However, he could only produce partial results and needed to further the research into this idea.[66]

McCarthy and Shannon's also bid these ideas was to develop a way to use complex problems to determine and measure the machine's efficiency through mathematical theory and computations.[66] However, they were only to receive partial test results.[66]

The idea behind machine learning is to look at developing methods that can be applied to problems by the machine itself. In this, they would try to find solutions to a problem in which the computer did not have answers. They wanted to create machines that could look at complex problems and be able to modify themselves to output feasible results. They applied this idea to imitate how the human brain works. This is why they came up with the term artificial intelligence to signify a machine that could predict the future; in their case it was the machine being able to predict the result of a game of chess. The human mind has the ability to look at the situation in a logical manner and make sense of it based on logic. The idea to make a machine act in this way is the concept behind artificial intelligence.[66]

Their goal though that research is the concept of a machine that could follow the program of a human and react to external stimulus in place of a human. They wanted to make a machine that could think in a logical and methodological way like a human by replacing the human with a machine.[66]

The idea of computational creativity is that the program or a machine can have human-like thinking and feelings. It is not clear however that a program could be created that mimics the original test set, and impose upon a mind to think in the same way as the human mind. However, the fact that artificial intelligence is a rapidly growing field and there are constant advances in the field, gives the idea of computing with the aid of machines to replicate a human being and their thinking processes a plausible theory.[66]

See also

- Computer Museum
- Computer science
- History of computing
- History of computing hardware
- History of software
- Timeline of algorithms
- Timeline of computing

References

[References section — numbered citations 1 through approximately 66, not fully legible at this resolution]

Sources

Further reading

[Bibliographic entries not legible at this resolution]