**Chapter One**

**Introduction to Computer Systems**

1. Historical Background

In this section, we would like to provide a historical background on the evolution of cornerstone ideas in the computing industry. We should emphasize at the outset that the effort to build computers has not originated at one single place. There is every reason for us to believe that attempts to build the first computer existed in different geographically distributed places.

In the sections that follow, we divide the evolution of computers into generations, each generation being defined by the technology used to build the machine. We have provided approximate dates for each generation for reference purposes only. You will find little agreement among experts as to the exact starting and ending times of each technological epoch.

* **Generation Zero: Mechanical Calculating Machines (1642–1945).** Prior to the 1500s, a typical European businessperson used an abacus for calculations and recorded the result of his ciphering in Roman numerals. After the decimal numbering system finally replaced Roman numerals, a number of people invented devices to make decimal calculations even faster and more accurate. Wilhelm Schickard (1592–1635) has been credited with the invention of the first mechanical calculator, the Calculating Clock (exact date unknown). This device was able to add and subtract numbers containing as many as six digits. In 1642, Blaise Pascal (1623–1662) developed a mechanical calculator called the Pascaline to help his father with his tax work. The Pascaline could do addition with carry and subtraction. It was probably the first mechanical adding device actually used for a practical purpose. In fact, the Pascaline was so well conceived that its basic design was still being used at the beginning of the twentieth century.
* **The First Generation: Vacuum Tube Computers (1945–1953).** Although Babbage is often called the “father of computing,” his machines were mechanical, not electrical or electronic. In the 1930s, Konrad Zuse (1910–1995) picked up where Babbage left off, adding electrical technology and other improvements to Babbage’s design. Zuse’s computer, the Z1, used electromechanical relays instead of Babbage’s hand-cranked gears. The Z1 was programmable and had a memory, an arithmetic unit, and a control unit. Because money and resources were scarce in wartime Germany, Zuse used discarded movie film instead of punched cards for input. Although his machine was designed to use vacuum tubes, Zuse, who was building his machine on his own, could not afford the tubes. Thus, the Z1 correctly belongs in the first generation, although it had no tubes.



U.S. Army, 1946

***H.W \ What Is a Vacuum Tube?***

* **The Second Generation: Transistorized Computers (1954–1965).** The vacuum tube technology of the first generation was not very dependable. In fact, some ENIAC detractors believed that the system would never run because the tubes would burn out faster than they could be replaced. Although system reliability wasn’t as bad as the doomsayers predicted, vacuum tube systems often experienced more downtime than uptime**.** In 1948, three researchers with Bell Laboratories—John Bardeen, Walter Brattain, and William Shockley—invented the transistor. This new technology not only revolutionized devices such as televisions and radios, but also pushed the computer industry into a new generation. Because transistors consume less power than vacuum tubes, are smaller, and work more reliably, the circuitry in computers consequently became smaller and more reliable. Despite using transistors, computers of this generation were still bulky and quite costly. Typically, only universities, governments, and large businesses could justify the expense. Nevertheless, a plethora of computer makers emerged in this generation; IBM, Digital Equipment Corporation (DEC), and Univac (now Unisys) dominated the industry. IBM marketed the 7094 for scientific applications and the 1401 for business applications. DEC was busy manufacturing the PDP-1. A company founded (but soon sold) by Mauchly and Eckert built the Univac systems. The most successful Unisys systems of this generation belonged to its 1100 series. Another company, Control Data Corporation (CDC), under the super- vision of Seymour Cray, built the CDC 6600, the world’s first supercomputer. The $10 million CDC 6600 could perform 10 million instructions per second, used 60-bit words, and had an astounding 128 kilowords of main memory.

***H.W/ What Is a Transistor?***

* **The Third Generation: Integrated Circuit Computers (1965–1980).** The real explosion in computer use came with the integrated circuit generation. Jack Kilby invented the integrated circuit (IC) or microchip, made of germanium. Six months later, Robert Noyce (who had also been working on integrated circuit design) created a similar device using silicon instead of germanium. This is the silicon chip upon which the computer industry was built. Early ICs allowed dozens of transistors to exist on a single silicon chip that was smaller than a single “discrete component” transistor. Computers became faster, smaller, and cheaper, bringing huge gains in processing power. The IBM System/360 family of computers was among the first commercially available systems to be built entirely of solid-state components. The 360-product line was also IBM’s first offering where all of the machines in the family were compatible, meaning they all used the same assembly language. Users of smaller machines could upgrade to larger systems without rewriting all of their software. This was a revolutionary new concept at the time.

***H.W/ What Is IC?***

* **The Fourth Generation: VLSI Computers (1980–????).** In the third generation of electronic evolution, multiple transistors were integrated onto one chip. As manufacturing techniques and chip technologies advanced, increasing numbers of transistors were packed onto one chip. There are now various levels of integration: SSI (small scale integration), in which there are 10 to 100 components per chip; MSI (medium scale integration), in which there are 100 to 1,000 components per chip; LSI (large scale integration), in which there are 1,000 to 10,000 components per chip; and finally, VLSI (very large-scale integration), in which there are more than 10,000 components per chip. This last level, VLSI, marks the beginning of the fourth generation of computers.

***H.W/ What Is VLSI?***

1. The Main Components of a Computer

Although it is difficult to distinguish between the ideas belonging to computer organization and those ideas belonging to computer architecture, it is impossible to say where hardware issues end and software issues begin. Computer scientists design algorithms that usually are implemented as programs written in some computer language, such as Java or C. But what makes the algorithm run? Another algorithm, of course! And another algorithm runs that algorithm, and so on until you get down to the machine level, which can be thought of as an algorithm implemented as an electronic device. Thus, modern computers are actually implementations of algorithms that execute other algorithms. This chain of nested algorithms leads us to the following principle:

**Principle of Equivalence of Hardware and Software**:

*Anything that can be done with software can also be done with hardware, and anything that can be done with hardware can also be done with software*.

A special-purpose computer can be designed to perform any task, such as word processing, budget analysis, or playing a friendly game of Tetris. Accordingly, programs can be written to carry out the functions of special-purpose computers, such as the embedded systems situated in your car or microwave. There are times when a simple embedded system gives us much better performance than a complicated computer program, and there are times when a program is the preferred approach. The Principle of Equivalence of Hardware and Software tells us that we have a choice. Our knowledge of computer organization and architecture will help us to make the best choice.

We begin our discussion of computer hardware by looking at the components necessary to build a computing system. At the most basic level, a computer is a device consisting of three pieces:

1. A processor to interpret and execute programs

2. A memory to store both data and programs

3. A mechanism for transferring data to and from the outside world

**Intel processors list**

|  |  |  |
| --- | --- | --- |
| **Processor** | **Production Date** | **No. of Cores** |
| 4040 | Nov. 15,1971 | 1 |
| 8008 | April 1972 | 1 |
| 8080 | April 1974 | 1 |
| 8085 | March 1976 | 1 |
| 8086 | June 8, 1978 | 1 |
| 8088 | June 1979 | 1 |
| 80286 | Feb. 1982 | 1 |
| 80386 | 1985 – 1990 | 1 |
| 80486 | 1989 – 1992 | 1 |
| [Intel Pentium](https://en.wikipedia.org/wiki/Intel_Pentium) | 1993 – 1999 | 1 |
| [Intel Pentium Pro](https://en.wikipedia.org/wiki/Intel_Pentium_Pro) | 1995 – 1998 | 1 |
| [Intel Pentium II](https://en.wikipedia.org/wiki/Intel_Pentium_II) | |  |  | | --- | --- | |  | 1997 - 1999 | | 1 |
| [Intel Pentium III](https://en.wikipedia.org/wiki/Intel_Pentium_III) | 1999 – 2003 | 1 |
| [Pentium 4](https://en.wikipedia.org/wiki/Pentium_4) | 2000 – 2008 | 1 |
| [Intel Atom](https://en.wikipedia.org/wiki/Intel_Atom) | 2008 - 2009 (as Centrino Atom)  2008–present (as Atom) | 1, 2 or 4 |
| [Intel Celeron](https://en.wikipedia.org/wiki/Intel_Celeron) | 1998–present | 1, 2 or 4 |
| [Intel Pentium Dual-Core](https://en.wikipedia.org/wiki/Intel_Pentium_Dual-Core) | 2006 – 2009 | 2 |
| [Intel Core 2](https://en.wikipedia.org/wiki/Intel_Core_2) | 2006 – 2011 | 1, 2 or 4 |
| [Intel Core i3](https://en.wikipedia.org/wiki/Intel_Core#Core_i3) | 2010–present | 2 /w hyperthreading |
| [Intel Core i5](https://en.wikipedia.org/wiki/Intel_Core_i5) | 2009–present | 2 /w hyperthreading, 4 |
| [Intel Core i7](https://en.wikipedia.org/wiki/Intel_Core_i7) | 2008–present | 4, 4 /w hyperthreading |
| [Intel Core i9](https://en.wikipedia.org/wiki/Intel_Core_i9) | Q3 2017-present | 10-18 (with hyperthreading) |

1. The von Neumann Model

The invention of stored program computers has been ascribed to a mathematician, John von Neumann, who was a contemporary of Mauchley and Eckert. Stored-program computers have become known as von Neumann Architecture systems. Today’s version of the stored-program machine architecture satisfies at least the following characteristics:

* + Consists of three hardware systems: A central processing unit (CPU) with a control unit, an arithmetic logic unit (ALU), registers (small storage areas), and a program counter; a main-memory system, which holds programs that control the computer’s operation; and an I/O system.
  + Capacity to carry out sequential instruction processing
  + Contains a single path, either physically or logically, between the main memory system and the control unit of the CPU, forcing alternation of instruction and execution cycles. This single path is often referred to as the von Neumann bottleneck.

Figure 1 shows how these features work together in modern computer systems. This architecture runs programs in what is known as the von Neumann execution cycle (*also called the fetch-decode-execute cycle*), which describes how the machine works. One iteration of the cycle is as follows:

1. The control unit fetches the next program instruction from the memory, using the program counter to determine where the instruction is located.
2. The instruction is decoded into a language the ALU can understand.
3. Any data operands required to execute the instruction are fetched from memory and placed into registers within the CPU.
4. The ALU executes the instruction and places the results in registers or memory.

**Central Processing Unit**

Program Counter

Registers

Arithmetic Logic

Unit

Control

Unit

Input/Output System

**Main Memory**

Program Counter

Main Memory

**Figure1: The von Neumann Architecture**

The ideas present in the von Neumann architecture have been extended so that programs and data stored in a slow-to-access storage medium, such as a hard disk, can be copied to a fast-access, volatile storage medium such as RAM prior to execution. This architecture has also been streamlined into what is currently called the system bus model. The data bus moves data from main memory to the CPU registers (and vice versa). The address bus holds the address of the data that the data bus is currently accessing. The control bus carries the necessary control signals that specify how the information transfer is to take place.

Other enhancements to the von Neumann architecture include using index registers for addressing, adding floating point data, using interrupts and asynchronous I/O, adding virtual memory, and adding general registers.

1. Non-von Neumann Model

Until recently, almost all general-purpose computers followed the von Neumann design. However, the von Neumann bottleneck continues to baffle engineers looking for ways to build fast systems that are inexpensive and compatible with the vast body of commercially available software. Engineers who are not constrained by the need to maintain compatibility with von Neumann systems are free to use many different models of computing. A number of different subfields fall into the non-von Neumann category, including neural networks (using ideas from models of the brain as a computing paradigm), genetic algorithms (exploiting ideas from biology and DNA evolution), quantum computation, and parallel computers. Of these, parallel computing is currently the most popular.

These improvements include adding specialized buses, floating-point units, and cache memories, to name only a few. But enormous improvements in computational power require departure from the classic von Neumann architecture.