ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

College of Engineering

Department of software engineering

course:Computer Architecture and Organization

| Group Members | ID No. |
|---|---|
| 1.Rahmet Abdella | ETS 1161/16 |
| 2.Sara Redwan | ETS 1241/16 |
| 3.Paulos Asmlash | ETS 1149/16 |
| 4.Samul Lire | ETS 1230 /16 |
| 5.Salim Ahmed | ETS 1213/16 |

Submitted to:Mr. Aderaw

Submission Date:Dec,3 2025GC

# Table of content

# Introduction

The first-generation computers were developed during the late 1930s to the mid-1950s and marked the beginning of electronic computing. Their invention was driven largely by the challenges of World War II and the need to perform calculations far faster than humans could. For example, the U.S. Army required accurate artillery firing tables, which involved solving thousands of complex equations. Human calculators using mechanical devices could take weeks to complete a single calculation, creating a critical need for faster, automated machines.

These early computers, such as ENIAC, EDVAC, and UNIVAC I, relied on vacuum tubes as their main electronic components. They were extremely large, often occupying entire rooms, consumed huge amounts of electricity, and required elaborate cooling systems due to the heat generated by thousands of vacuum tubes. Programming was done manually using switches, cables, and patchboards, and memory was very limited compared to modern standards.

Despite these challenges, first-generation computers were revolutionary. They could perform thousands of calculations per second, far surpassing human capabilities. They also introduced the stored-program concept, which allowed instructions to be stored in memory alongside data, making the machines more flexible. These computers laid the foundation for modern computing, transforming calculations and data processing and setting the stage for subsequent generations of faster, smaller, and more reliable computers.

# 1. Structure of First-Generation Computers

The structure of first-generation computers (built between 1940 and 1956) was based on vacuum tube technology and consisted of several interconnected components that enabled electronic computation. These computers included a central processing unit (CPU), which contained the arithmetic and logic unit (ALU) and control unit (CU), primary and secondary memory for storing data and instructions, and input/output devices such as punched cards and printers. The machines relied on supporting components like resistors, capacitors, diodes, and extensive wiring to manage signals and circuits. We will be seeing them in detail.

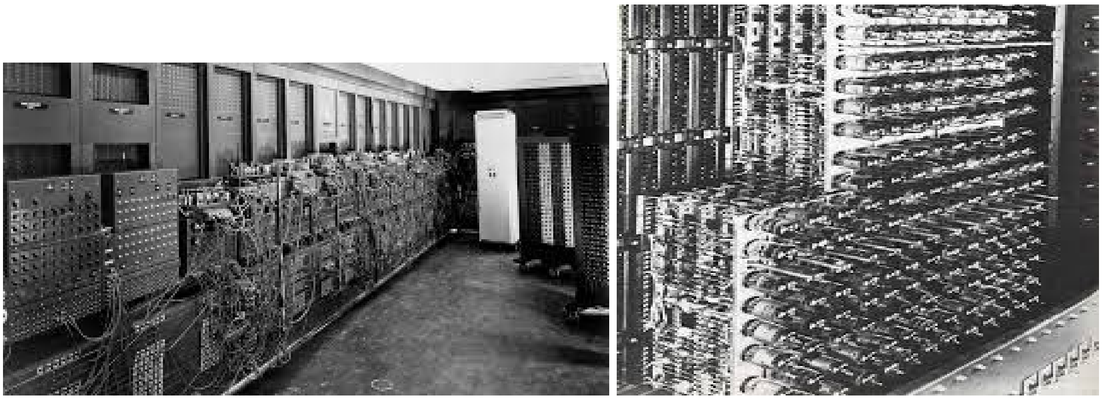


fig 1.1 first gen computers and vacuum tubes

### 1.1 Electronic Components

### 1.1.1 Vacuum Tubes

These were the basic building blocks, functioning as switches and amplifiers to control electrical signals. Each tube could represent a binary 0 or 1, but thousands were required for even simple operations.

**How Vacuum Tubes Worked ?**

Vacuum tubes were glass cylinders with electrodes that allowed electric current to flow. They acted as electronic switches and amplifiers, managing the flow of electricity in circuits. The basic type used in computers was the triode, invented by Lee De Forest in 1906. It consisted of three parts:

- Cathode: Heated to emit electrons.

- Anode (Plate): Attracts electrons to create current.
- Control Grid: Regulates the flow of electrons between cathode and anode.

By adjusting the voltage on the control grid, the tube could switch between conducting and non-conducting states, representing binary 1s and 0s. This switching ability allowed vacuum tubes to perform logical operations and store temporary data, forming the basis of digital computation.

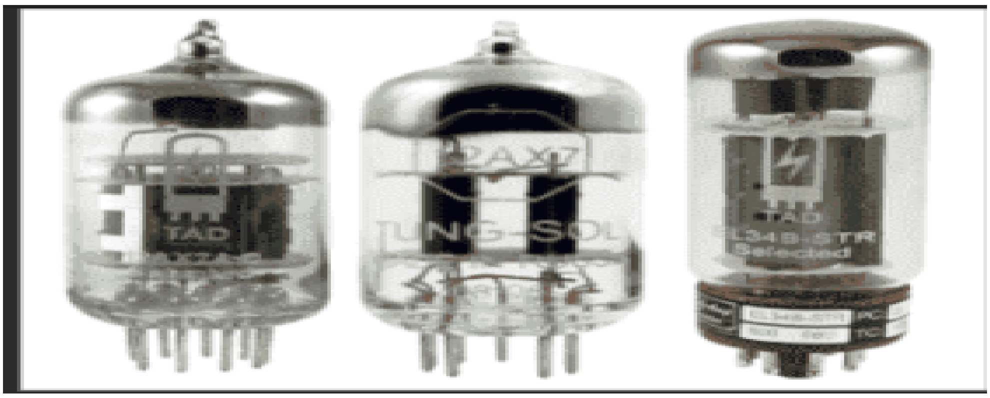
Role in First Generation Computers

Vacuum tubes were widely used in the logic circuits, memory units, and arithmetic processors of early computers. Notable examples include:

- ENIAC (1946): Used over 17,000 vacuum tubes and took up an entire room.
- UNIVAC I (1951): The first commercial computer, also based on vacuum tube technology.

**Limitations of Vacuum Tubes**

- **Size and Heat:** Tubes were large and produced a lot of heat, requiring complex cooling systems.
- **Reliability**: Tubes were fragile and had short lifespans. A single tube failure could crash the entire system.
- **Maintenance:** Frequent replacements and repairs were necessary, demanding constant technical attention.



**Vaccume Tube**

### 1.1.2 Supporting Components:

- **Resistors** : controlled the flow of current in circuits and helped set proper voltage levels for vacuum tubes
- **Capacitors** :stored small amounts of electric charge temporarily, smoothing voltage fluctuations and supporting timing functions in memory and logic circuits.
- **Diodes**:Used to allow current to flow in only one direction, diodes were essential in early logic circuits, such as creating AND and OR gates.
- **Inductors and Transformers**:These components were used in power supply circuits to regulate voltages, step up or step down electrical current, and isolate different parts of the system.
- **Switches and Relays:**Manual switches or electromechanical relays were used to configure circuits, load programs, and control the flow of electricity in peripheral devices.

## 1.2 Central Processing Unit (CPU)

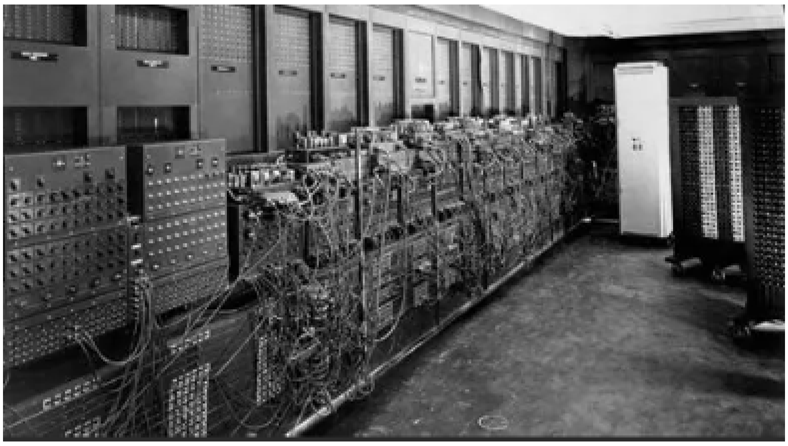
The Central Processing Unit (CPU) was the “brain” of first-generation computers, responsible for processing data and controlling the flow of instructions. It consisted of two main components: the Arithmetic and Logic Unit (ALU) and the Control Unit (CU).

The ALU performed all the basic arithmetic operations, such as addition, subtraction, multiplication, and division, as well as logical operations like comparisons. These operations were carried out using networks of vacuum tubes, which acted as high-speed electronic switches representing binary digits (0 and 1).

The Control Unit coordinated all activities of the computer by fetching instructions from memory, decoding them, and sending signals to the ALU, memory, and input/output devices. In first-generation computers, the control unit was hardwired, meaning its logic circuits were physically wired to execute instructions sequentially.

In addition, the CPU included registers, which were small, high-speed storage locations that temporarily held data, intermediate results, and addresses during processing. The CPU worked closely with memory and input/output units to execute programs stored in binary machine language, performing operations one instruction at a time.

Although primitive compared to modern CPUs, the first-generation CPU enabled automatic electronic computation, making it significantly faster and more reliable than earlier mechanical calculators. Its design formed the foundation of all subsequent computer generations.

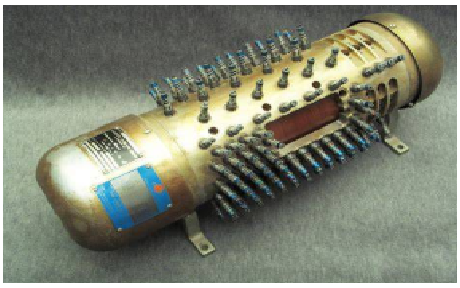


### 1.3 Memory Components

Memory in first-generation computers was used to store data, instructions, and intermediate results during program execution. It was divided into primary (main) memory and secondary (auxiliary) memory.

<u>Primary memory</u> temporarily held the instructions and data that the CPU needed while performing calculations. Early computers used devices like magnetic drums, mercury delay lines, and Williams tubes for this purpose.

- **Magnetic drums :**were rotating cylinders coated with a magnetic material that stored data in the form of magnetic patterns.
- **Mercury delay lines**: stored data as pulses of sound traveling through tubes of mercury.
- **Williams tubes :**used the surface of cathode ray tubes to hold data as charged dots.



**Magnetic Drums**

Primary memory was small in size and could store only a limited number of instructions and data at a time.

- **Secondary memory** was used for long-term storage of programs and data. This included punched cards, paper tapes, and magnetic tapes, which allowed users to input programs into the computer and store output results. Unlike modern computers, first-generation computers had no random-access memory (RAM) as we know it today, so access to data was often sequential and relatively slow.

## 1.4 Input/Output Devices

Input and output devices in first-generation computers were responsible for feeding data and instructions into the computer and displaying or storing the results of computations.

Input devices primarily included punched card readers and paper tape readers. Users would prepare programs and data by punching holes into cards or tapes, with each hole representing a binary digit (0 or 1). The computer read these cards or tapes to fetch instructions and input data into memory for processing. Some early machines, like the ENIAC, even required manual switches or plugboards to enter instructions directly, which made programming more labor-intensive.

Output devices included line printers, punched card punches, and teleprinters. After the computer completed a calculation, results were either printed on paper or punched into cards for further use or analysis. The output was generally limited to textual or numeric data, and there was no visual display like modern screens. Some machines also used cathode ray tubes (CRTs) for temporary visual display of results.

Together, input and output devices allowed first-generation computers to interact with users, read programs and data, and produce results in a form that could be interpreted and utilized. While slow and cumbersome by modern standards, these devices were essential for making electronic computation practical during the first generation of computing.

# 2. Organization of First-Generation Computers

The organization of first-generation computers refers to the arrangement and interaction of their internal components to perform computation. These computers were designed based on the Von Neumann architecture, where the CPU, memory, and input/output units were connected to allow the flow of data and instructions in a sequential manner. The control unit directed the operation of the arithmetic and logic unit (ALU), memory, and I/O devices, ensuring that instructions were executed in the correct order. Memory was organized to store both data and program instructions, while input and output devices provided communication between the computer and the user. This structured organization enabled first-generation computers to process information automatically and electronically, marking a significant advance over mechanical calculating machines.

## 2.1 Von Neumann Architecture

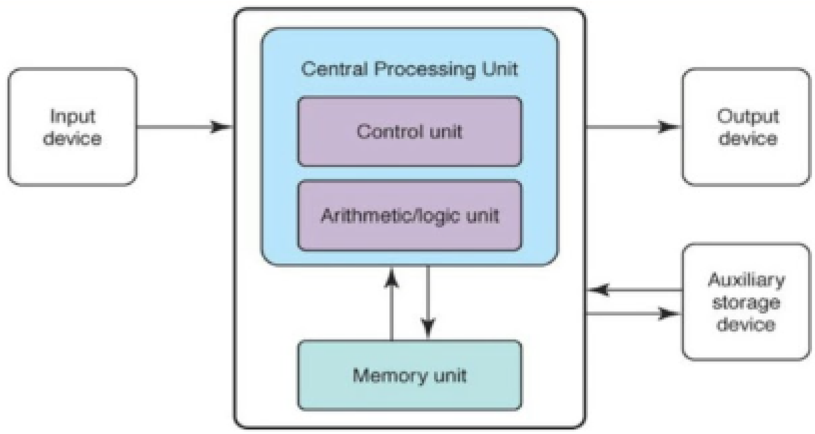
The Von Neumann architecture, proposed by John von Neumann in 1945, is the foundational design for most first-generation computers. It describes a system in which a computer's memory, central processing unit (CPU), and input/output devices are organized to allow stored-program operation. In this architecture, both program instructions and data are stored in the same memory, enabling the CPU to fetch, decode, and execute instructions sequentially.

The CPU itself is divided into the arithmetic and logic unit (ALU), which performs mathematical and logical operations, and the control unit (CU), which directs the flow of data between memory, the ALU, and input/output devices. Input devices supply programs and data to memory, while output devices display or record the results. The system bus connects all major components, allowing information to move between them.

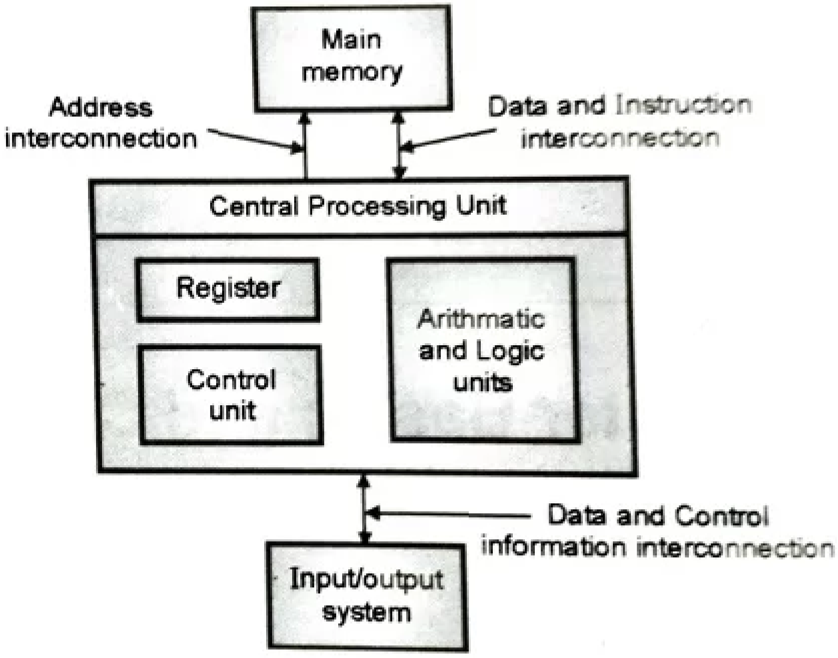
The Von Neumann architecture introduced the concept of automatic electronic computation, where programs could be stored and executed without manual rewiring or physical configuration changes. This design principle was used in early computers such as ENIAC (later modified) and UNIVAC I, enabling them to perform calculations efficiently and reliably compared to mechanical calculators.

Generally von numan architecture introduced:

- A single memory for both data and instructions.
- Sequential execution of instructions (one at a time).
- Use of a control unit to coordinate memory, ALU, and I/O.




**Von Neumann Architecture**




## 2.2 System Units and Data Flow

The system consisted of:

**1. Input Unit**: The input unit is responsible for taking data and instructions from the user and converting them into a form that the computer can understand. Devices like keyboard, mouse, scanner, and microphone serve as input tools. Without the input unit, the computer would not know what tasks to perform.

**2. Memory Unit:** The memory unit stores both data and instructions temporarily or permanently. It includes primary memory (RAM, which is fast but temporary) and secondary storage (like hard drives, which are slower but permanent). The memory unit ensures that the CPU has quick access to the information it needs to process tasks efficiently

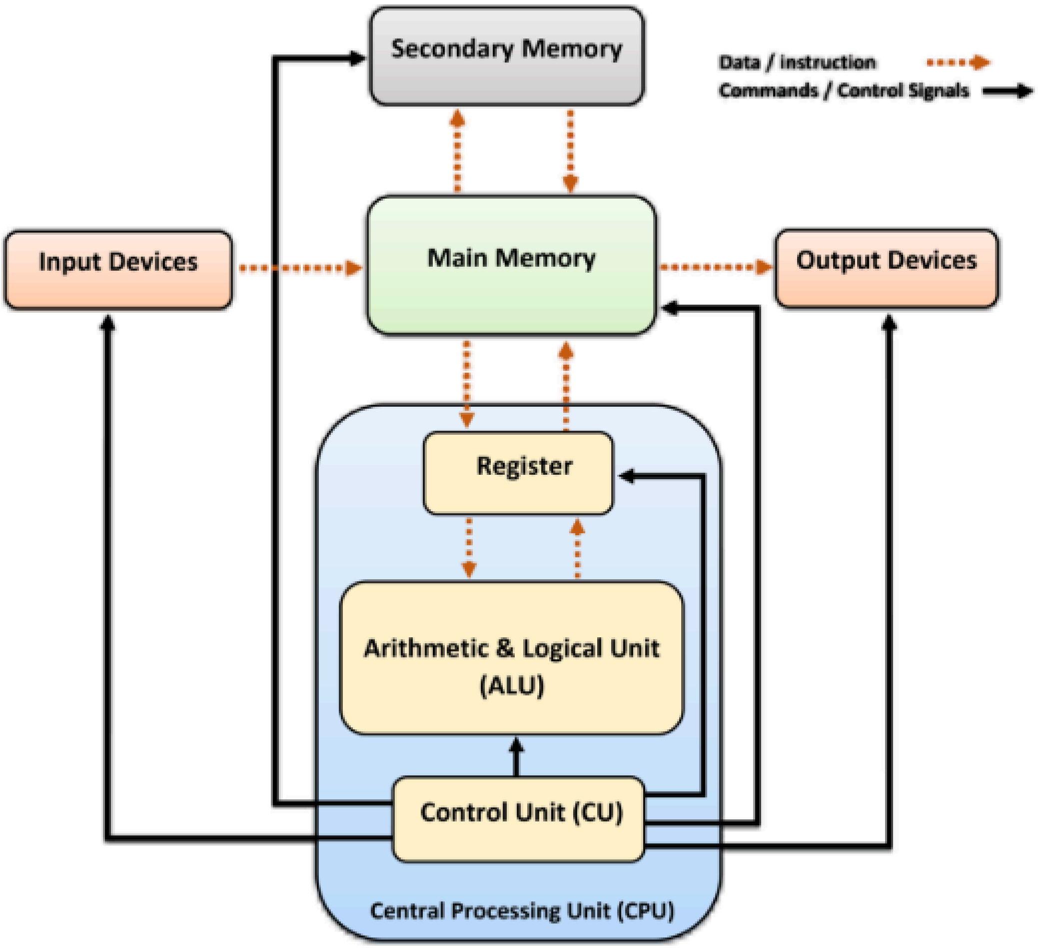
**3. CPU (Central Processing Unit):** The CPU is the brain of the computer, responsible for processing data and executing instructions. It has two main parts:

- **ALU (Arithmetic Logic Unit):** Performs all arithmetic calculations (like addition and multiplication) and logical operations (like comparisons).

- **CU (Control Unit):** Directs the flow of data between the CPU, memory, input, and output units, ensuring instructions are executed in the correct order.

**4. Output Unit:**The output unit delivers the results of the computer's processing to the user in a readable form. Examples include monitors, printers, and speakers. The output unit allows humans to interact with the computer and use the information it produces.

Data moved through a common bus system, meaning all transfers shared the same communication lines between memory and the CPU.




**Block diagram of a computer system showing input, memory, CPU (ALU + CU), output and system bus illustrating data flow**

## 2.3 Memory Organization

Memory organization in the first generation of computers, which spans roughly from 1940 to 1956, was one of the most defining technological challenges of early computing. Computers such as ENIAC, EDVAC, and UNIVAC I marked a transition from mechanical computation to fully electronic processing, and one of the most crucial components of their architecture was the way memory was structured and used. Since there were no integrated circuits, no semiconductor memory, and not even a unified concept of random access memory in its modern sense, the early machines relied on a mixture of physical storage technologies that were dramatically different from each other in speed, capacity, and method of operation. This early memory design directly affected the speed, reliability, and overall capability of the computer and influenced the way programmers wrote and structured their programs.

One of the most revolutionary ideas that appeared during this generation was the stored program architecture, also known as the Von Neumann Architecture. Before this idea, computers such as ENIAC were programmed manually through the use of cables, switches, and plugboards. Every change in procedure required physically rewiring the machine. With the stored program concept introduced in the EDVAC design, both the instructions that made up a program and the data that program manipulated were stored together in the same memory unit. This concept became the foundation of all later computers. It meant the computer could load instructions electronically, modify its own program during execution, treat instructions as data, and replace entire programs without touching the hardware. This required a new kind of memory organization capable of holding both data and instructions in an integrated structure.

The memory of first generation computers was organized into several layers. The first and primary layer was the main memory. This was where the active program and its data were stored during execution. Two primary technologies were used to implement this part of memory. The first was mercury delay line memory. In this system, data was stored as acoustic pulses traveling through a long tube filled with mercury. An electrical signal was converted into a sound wave using a quartz crystal, sent through the mercury, detected at the other end, amplified, and fed back. The bits circulated continuously inside the tube. While effective for the time, this technology had the significant limitation of serial access. The computer had to wait for the desired bit to reach the output end of the tube before it could be read. This introduced variable access delays and slowed overall processing. The second main memory technology of the period was the Williams tube, which used a cathode ray tube to store bits as electrically charged spots on the phosphor surface. A metal plate inside the tube detected changes in charge when the electron beam scanned over the stored data. This system worked as one of the earliest forms of random access memory because the machine could go directly to any memory location. It was faster than delay lines but required continuous refreshing and was sensitive to electrical disturbances. The capacity of primary memory was extremely small, often only a few hundred to a few thousand words.

The second layer of memory was secondary storage, used to hold data and programs that did not fit in the limited main memory. The dominant form of secondary memory was magnetic tape. It offered much larger storage capacity but could only be accessed sequentially. Machines such as UNIVAC I used magnetic tape drives to load programs into main memory when needed. Once the program finished executing, the results were written back to another tape for long-term storage. Because tape had to be physically wound forward or backward to reach a specific location, access was slow, but the large capacity made it essential for storing entire data sets and programs.

The third and smallest layer of memory was located inside the central processing unit. This included a handful of special purpose registers that were built directly using vacuum tubes. These registers were the fastest memory in the computer. They included the accumulator, which stored intermediate arithmetic results, the instruction register, which held the current instruction, and the program counter, which held the memory address of the next instruction to be executed. Although few in number, these registers played a crucial role in the execution cycle. To read or write main memory, the computer used a memory address register to store the target address and a memory data register to hold the value being transferred. These internal memory structures created an organized communication pathway between the CPU and the rest of memory.

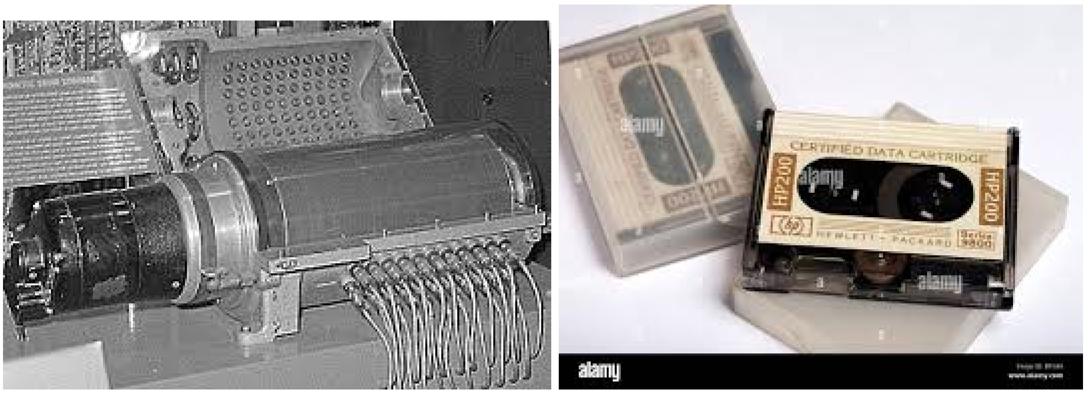
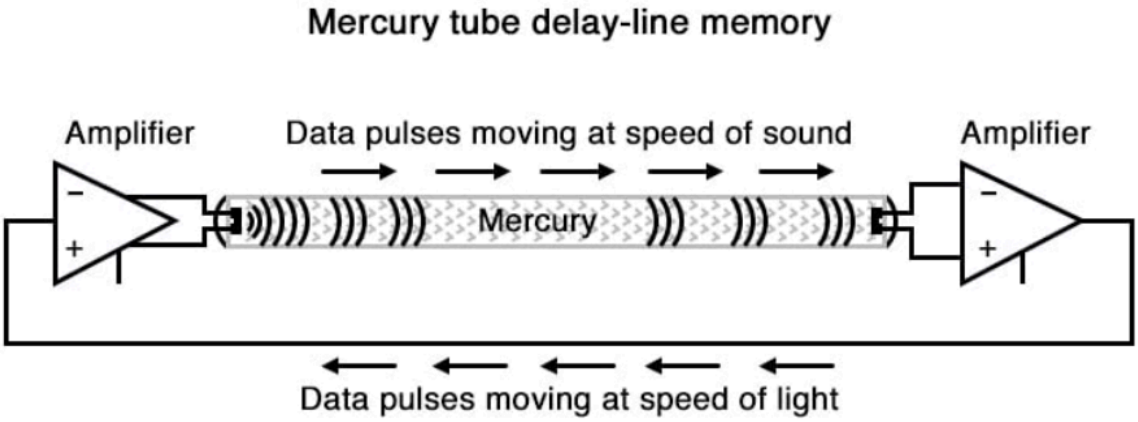



Fig  Mgnetic drum and Magnetic tapes

The physical and logical nature of memory in this era produced several consequences. Many early machines were word oriented rather than byte oriented, meaning the memory was addressed in terms of long words that contained several digits or bits. Machines that used delay lines often operated serially, processing one bit at a time because the memory itself was serial. This contrasted with later parallel systems that processed entire words at once. The physical behavior of memory influenced programming practices. For example, with delay line memory, programmers had to carefully arrange instructions in a sequence that minimized waiting time. This practice was known as optimum coding. Likewise, Williams tubes required constant refreshing, which demanded careful electrical management.

Several challenges emerged from these early memory systems. Size limitations were severe. With only a few kilobytes of memory available, programmers were forced to write extremely compact and efficient code. Large programs had to be divided into smaller chunks that were loaded one at a time, creating a primitive early form of overlay management. There was also a significant speed mismatch between the CPU and memory. Vacuum tube circuits could operate relatively quickly, but delay line memory was slow. As a result, the CPU often sat idle waiting for memory, creating the earliest version of what would later be named the Von Neumann bottleneck. Reliability was another issue. Vacuum tubes burned out frequently, mercury delay line memory was sensitive to temperature, and the charged spots on Williams tubes could fade or be disrupted. This meant early computers required constant maintenance to function properly.




### 2.4 Interconnection and Communication

Signal pathways were hard-wired through switchboards and cables. Since interconnections were fixed, changing the logic or adding new instructions required physical modification.

## 3. Function of First generation computer

The first-generation computers, developed during the 1940s and 1950s, marked a major milestone in computing history. Machines such as the ENIAC (Electronic Numerical Integrator and Computer) and UNIVAC I (Universal Automatic Computer I) were primarily designed to perform numerical calculations at speeds far beyond human capability. They could execute arithmetic operations addition, subtraction, multiplication, and division with remarkable accuracy, making them invaluable tools for scientific research, engineering projects, and military applications. For example, the military used them for ballistic trajectory calculations, which involved processing complex mathematical equations much faster than humans could.

Beyond numerical computation, first-generation computers were also capable of data processing, which made them useful for businesses and government organizations. They could handle large volumes of numerical information, which was essential for tasks such as payroll, census analysis, and inventory management. However, the input and output processes were slow and cumbersome, relying on punch cards, paper tapes, and printed reports, which limited efficiency.

Some of the key functions and capabilities of first-generation computers included:

- **Numerical calculations:** Fast and accurate execution of arithmetic operations, reducing human effort in complex calculations.

- **Data processing:** Handling large datasets for tasks like payroll, census statistics, and inventory control.

- **Sequential program execution:** Programs ran automatically in order once set up, though changing programs required **manual rewiring**.

- **Temporary memory storage:** Using magnetic drums or delay line memory to store data and instructions temporarily.

- **Basic logical operations:** Ability to perform comparisons and simple decision-making within programs.

Despite these capabilities, first-generation computers had notable limitations. Memory capacity was extremely small, input/output operations were slow, and programming was cumbersome due to the need for manual setup. Nevertheless, these computers revolutionized computation, automating repetitive and complex tasks for scientific, industrial, and military applications. Their development laid the foundation for the more advanced and flexible computers of the second generation and beyond.


**first commercial electronic computer**

## 4. Operation of First-Generation Computers

### 4.1 Program Input and Loading

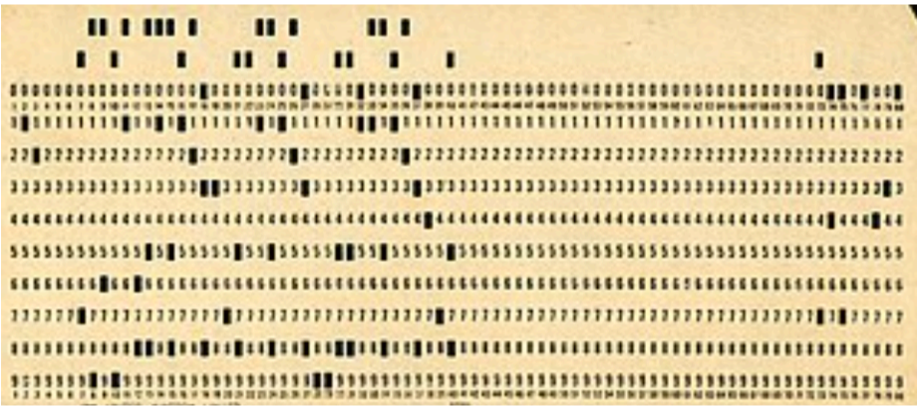
Instructions were entered into first-generation computers primarily through punched cards or paper tapes, which represented programs in machine language (binary code: 0s and 1s). Early machines, like the ENIAC, initially required manual rewiring and setting switches to input a program, which made programming slow and inflexible. Later first-generation computers, such as UNIVAC, could read instructions from magnetic drums or punched media, allowing the CPU to automatically fetch and execute instructions sequentially. Each instruction told the computer which operation to perform, what data to use, and where to store the result.

#### 4.1.1 Punch Cards

Punch cards (also called Hollerith cards) were stiff paper cards with rectangular holes punched in predefined positions to represent data or instructions in binary form. Each column on the card could represent a number, letter, or special symbol. Computers read these cards using a card reader, which detected the presence or absence of holes in each position.

1. Early programmers would encode instructions (machine language) as a sequence of holes on multiple cards. Each card could represent a single instruction or a set of data.
2. Numbers, text, or other data were punched into cards and fed into the computer sequentially.
3. The computer read the cards one at a time, executed the instructions, and performed calculations.
4. Results were printed on paper or sometimes punched back onto new cards for further use.

Punch cards allowed batch processing, where a large set of instructions and data could be executed without human intervention once loaded. However, they were time-consuming to prepare and could be easily damaged or misordered.



**punched cards**, **card readers**, **paper-tape readers**, or **magnetic-tape reels**

### 4.1.2 Paper Tapes

Paper tapes were long strips of paper with holes punched along their length to encode instructions or data in a serial manner. Each line of holes represented a single character, number, or command. Paper tapes were read by a tape reader, which sensed the holes electrically and converted them into signals the computer could understand.

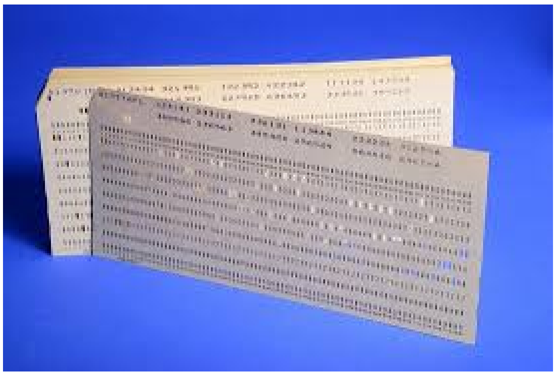
1. Programs were encoded on paper tape in machine language or sometimes in an early form of assembly language.
2. Paper tapes could store both instructions and data, which were fed into the computer sequentially.
3. Unlike punch cards, paper tape allowed continuous streaming of data, which could be fed into the computer without stopping, making it useful for repetitive or long-running calculations.

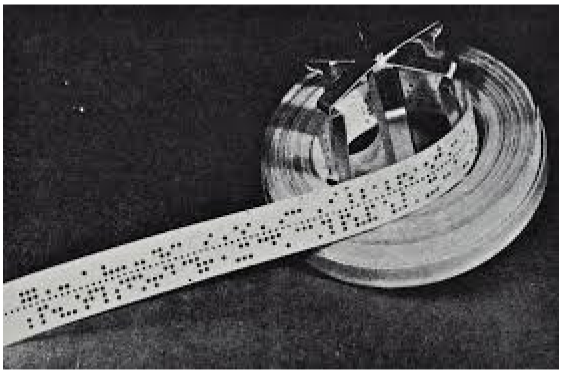
Paper tapes were more flexible than punch cards because they could be rolled into spools for continuous reading and reused, but they were also delicate and could tear easily.

These punched cards or tapes were then loaded into the computer using an input device such as a card reader or tape reader. The reader converted the holes into electrical signals that represented binary data.

These binary signals were then sent to the main memory, which could be a magnetic drum, delay line, or later, electrostatic memory. The memory stored the instructions as electrical charges or magnetic patterns in specific locations called memory addresses. Each address corresponded to one instruction or one piece of data.

When the computer started running the program, the control unit (a part of the CPU) sent signals to the memory to fetch the next instruction. The memory then transferred the stored binary instruction back to the control unit through electronic circuits. This process is  known as the fetch stage of the fetch-decode-execute cycle.





Punchcards b.paper tapes

## 4.2 Instruction Execution Cycle

Once the instruction was fetched from memory in a first-generation computer, it was then executed through a series of carefully timed electronic operations controlled by the control unit and carried out by the arithmetic logic unit (ALU). This process followed the same basic pattern used in modern computers, known as the **fetch-decode-execute cycle**, but it was entirely implemented using vacuum tubes and hardwired circuits.

After the control unit fetched the instruction from memory, it decoded the binary pattern to determine what operation needed to be performed. The instruction was made up of two main parts  the operation code (opcode) and the operand. The opcode told the computer *what to do* (for example, add, subtract, or load data), while the operand specified *where to find the data* in memory. The control unit interpreted these bits using electronic logic circuits built with vacuum tubes.

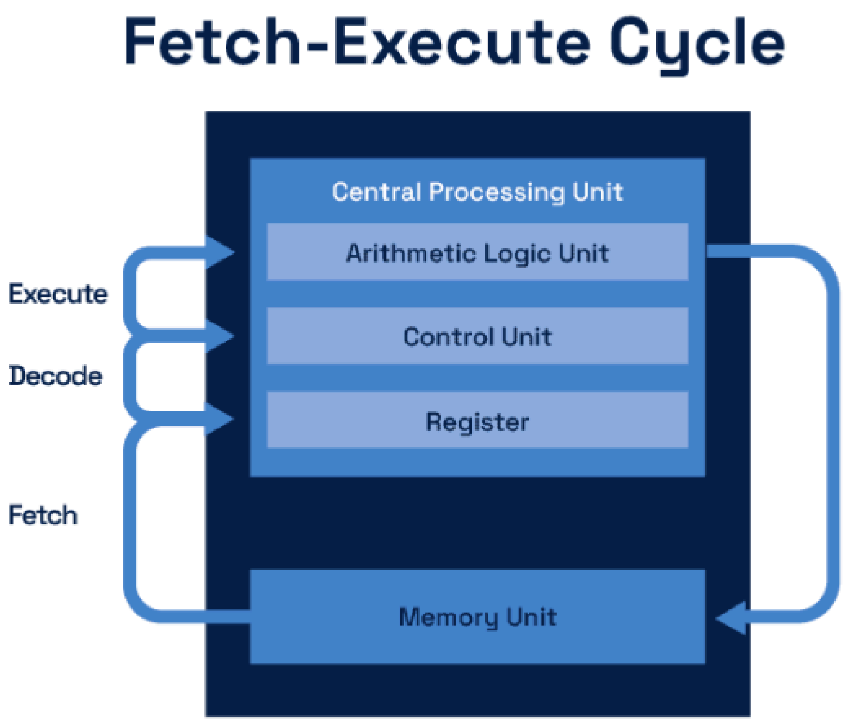
Next came the execution phase. During this stage, the control unit sent the necessary control signals to the ALU or other parts of the system. For example, if the instruction was an addition, the ALU would retrieve the numbers from memory or registers and perform the calculation. The result was then sent back to memory or stored in a temporary register for later use. Each of these operations required a sequence of electrical pulses that turned the vacuum tubes on and off in precise timing, representing the flow of binary data.

Once the instruction was executed, the control unit moved to the next instruction by increasing the program counter, which kept track of the memory address of the next command. This cycle—fetch, decode, and execute—repeated continuously until the program finished running or an input/output operation interrupted it.

After the arithmetic or logical operation was completed by the arithmetic logic unit (ALU), the control unit sent electronic control signals to guide where the result should go. The result was temporarily held in a register, usually called the accumulator, which was a small, fast storage location inside the CPU. From there, the data was transferred into the main memory so that it could be used by later instructions or displayed as output.

The memory itself in first-generation computers was typically made of magnetic drums, delay line memory, or electrostatic storage. Each memory cell or position had a unique address, which the control unit used to locate exactly where to store the result. The binary data (a pattern of 0s and 1s) representing the result was converted into electrical or magnetic signals. These signals either created or removed magnetic fields on a magnetic drum's surface or formed standing sound waves in a delay line. In this way, the result was stored as a physical change in the memory device.

Once the storing process was complete, the computer was ready to fetch the next instruction. The program counter would move to the address of the following command, and the fetch-decode-execute cycle would start again.




### 4.3 Output Generation

After the store stage, when the processed data was saved in memory, the control unit sent signals to the output devices to present the results. These devices were connected to the computer through electronic circuits that carried the binary results as electrical pulses. The pulses represented numbers or characters, and the output devices translated those patterns into printed text, punched holes, or visual indicators.

The most common output devices in first-generation computers were printers, punch card machines, paper tape punches, and display panels with rows of lights. For example, a line printer could print the output results onto paper. Each number or character was represented by a specific combination of electrical signals that activated mechanical print heads. In the case of a punch card or paper tape, the computer's results were recorded as a series of holes, which could later be read or processed by another machine.

Some systems also used indicator lights or oscilloscopes for immediate monitoring. The lights on the computer's front panel would turn on or off to represent binary values, allowing operators to see the contents of certain registers or memory locations. Although this was not very convenient for reading large results, it was useful for debugging and testing the machine's operation.

# 5 Performance and Characteristics

### 1. Speed and Processing Power

First-generation computers could perform only a few thousand calculations per second, which was considered impressive at the time. For example, the ENIAC could perform about 5,000 additions per second and about 300 multiplications per second. However, each operation took milliseconds, and the machines took time to read, process, and output data. The overall speed was limited by the switching time of vacuum tubes and the slow input/output devices such as punch card readers and printers.

## 2. Memory and Storage Capacity

Memory in first-generation computers was extremely small and expensive. These computers used magnetic drums, delay line memory, or electrostatic storage to hold data and instructions. Storage capacities were measured in kilobytes, and access time was very slow often in milliseconds. Because memory was so limited, only small programs could be run at one time.

### 3. Reliability and Maintenance

Vacuum tubes were large, produced a lot of heat, and burned out frequently. A single machine could contain thousands of tubes, and if one failed, the computer stopped working. This made early computers unreliable and required constant maintenance. Cooling systems were necessary to prevent overheating, which made the machines even larger and more power-hungry.

### 4. Power Consumption

First-generation computers consumed a huge amount of electrical power. For example, ENIAC used about 150 kilowatts of electricity, enough to power several homes. Most of this power was used to heat the vacuum tubes and operate electromechanical components.

### 5. Programming and Efficiency

Programming was done in machine language or sometimes assembly language, using binary code or numeric symbols. This made writing, debugging, and modifying programs extremely difficult and time-consuming. There were no operating systems, compilers, or high-level languages, so every instruction had to be manually coded and entered through punch cards or switches.

### 6. Size and Portability

First-generation computers were enormous often filling an entire room. Machines like ENIAC and UNIVAC I weighed tons and used thousands of vacuum tubes. Moving or upgrading them was nearly impossible, and installing them required specialized environments with controlled temperature and humidity.

## 6.Limitations

1. **Use of Vacuum Tubes**

The biggest limitation of first-generation computers was their reliance on vacuum tubes. Although these tubes made electronic computation possible, they were large, fragile, and generated a lot of heat. Because thousands of tubes were used in a single computer, they frequently burned out, causing the machines to stop working. This made early computers highly unreliable and required constant maintenance.

2. **Enormous Size**

First-generation computers were very large and bulky, often taking up entire rooms. For example, the ENIAC weighed about 30 tons and used over 18,000 vacuum tubes. The huge size also meant that these machines needed a lot of space and special cooling systems to keep them from overheating. This made them difficult to install and operate.

3. **High Power Consumption**

These computers consumed massive amounts of electricity because of the vacuum tubes and cooling fans. The power requirements were so high that they were expensive to run and could cause frequent electrical problems. For instance, ENIAC used about 150 kilowatts of power, enough to power several homes.

**4. Limited Speed and Storage**

Although they were faster than humans, the speed of first-generation computers was very slow compared to later generations. They could only perform a few thousand calculations per second. Their memory and storage capacity were also very small, usually measured in just a few kilobytes, and used magnetic drums or delay line memory, which were slow to access.

**5. Difficult Programming**

Programming first-generation computers was extremely difficult because they could only understand machine language (binary code made up of 0s and 1s). Programmers had to write instructions manually, which made developing software time-consuming and error-prone. There were no operating systems, compilers, or high-level programming languages to simplify the process.

**6. Limited Versatility**

These computers were built for specific tasks, such as scientific calculations, military work, or census processing. They were not general-purpose machines like today's computers. Changing a program or performing a new task often required manual rewiring or reconfiguring hardware, which was slow and inefficient.

**7. Very Expensive**

First-generation computers were extremely costly to build and operate. Only large organizations, governments, or research institutions could afford them. The high cost of components like vacuum tubes, along with the electricity and maintenance requirements, made these computers impractical for general use.

**8. Limited Input/Output Devices**

The input and output methods were also slow and mechanical. Input was given through punch cards or paper tape, and output was obtained through printers, paper tape punches, or indicator lights. These methods made data entry and result retrieval time-consuming and prone to human error.

## 7.Legacy

The first-generation computers were the pioneers of the digital age, introducing the world to **automatic electronic computation**. They replaced slow, manual calculations with machines capable of performing **complex arithmetic and data processing tasks** much faster than humans. Though these machines were large, costly, and limited in memory and speed, they demonstrated the **enormous potential of computers** and inspired confidence in their future.

Their impact went beyond technology they transformed how **science, industry, and government** approached problem-solving. Military calculations, business payrolls, census processing, and scientific research all became faster and more accurate thanks to these early machines. The first-generation computers **laid the groundwork for the rapid evolution of computing**, influencing the development of smaller, faster, and more flexible machines in later generations.

In essence, their legacy is not just in the hardware they created, but in the **vision they sparked**: that electronic computers could revolutionize human work, making tasks that once took months or years possible in hours or minutes.

## 8.Conclusion

First-generation computers, despite their many limitations, represented a major milestone in the evolution of computing technology. They introduced the world to electronic computation, replacing slower mechanical and electromechanical devices, and demonstrated that complex calculations could be performed automatically and at speeds far beyond human capability. These computers were large, consumed enormous amounts of power, and required constant maintenance due to the frequent failure of vacuum tubes, but they nonetheless proved the practicality and potential of electronic machines.

Importantly, first-generation computers established fundamental concepts that remain central to computing today. The stored-program architecture, where both instructions and data are stored in memory, was first implemented in these machines and became the foundation for all modern computers. They also highlighted the need for more efficient programming methods, leading to the development of assembly languages and, eventually, high-level programming languages. Input and output methods, though primitive—using punch cards, paper tapes, and printers—demonstrated the importance of interfacing machines with human users, setting the stage for more advanced I/O systems in later generations.

The legacy of first-generation computers extends beyond technology. They were used in scientific research, military operations, and large-scale administrative tasks, proving the value of computers in solving real-world problems. Moreover, the challenges they presented, such as size, heat, power consumption, and unreliability, inspired engineers and scientists to innovate, leading to the invention of transistors and the creation of smaller, faster, and more reliable second-generation computers.

In summary, while first-generation computers were slow, bulky, and expensive, they laid the foundation for modern computing. They transformed theoretical ideas into practical machines, demonstrated the power of automated electronic calculation, and set in motion a series of technological advancements that have shaped the digital world we live in today. Their development was not just a technical achievement but a pivotal moment in human history, marking the birth of the computer age.

# Reference


1. **Haigh, T.**, *ENIAC in Action*, MIT Press, 2016.
2. **Ceruzzi, P.**, *A History of Modern Computing*, MIT Press, 2003.
3. **Wikipedia Contributors**, “First-generation computer,” *Wikipedia, The Free Encyclopedia*.
4. **Britannica, The Editors of Encyclopaedia**, “ENIAC,” *Encyclopedia Britannica*, 2023.
5. **Computer History Museum**, “Timeline of Computer History,” *computerhistory.org*.
6. **Williams, M.R.**, *A History of Computing Technology*, IEEE Press, 1997.