

Chapter 1

Introduction to Computer Science

Dr. Adel ABDELLI

adel.abdelli@univ-biskra.dz

Contents

- a. Definitions of Computer Science
- b. Evolution of Computer Science and Computers
- c. Principles of How a Computer Functions
- d. Hardware Components of a Computer
- e. Operating Systems
- f. Systems for Encoding Information

1. Definition of Computer Science:

Computer science, in simple terms, is the study of computers and computational systems. It involves understanding how computers work, how they process information, and how we can use them to solve problems and create new technologies [1].

What is a computer?

A computer is an electronic device that accepts data as input, performs calculations and operations on that data, displays output results, and can store the data or results as needed. It combines both hardware and software resources that work together interactively to provide various functionalities to the user [1].

The hardware refers to the physical components of a computer such as the central processing unit (CPU), memory storage devices, monitor, keyboard, and other physical parts.

The software consists of programs and coded instructions that are required by the hardware in order to function properly. Software tells the hardware how to process input, perform calculations, store and retrieve data, display visuals, and communicate.

Together, the integrated hardware and software allow computers to rapidly accept vast amounts of digital information as input, perform complex operations on that data, store and retrieve files efficiently, and output results in useful formats. This makes computers invaluable tools that augment human capabilities in countless applications across all domains of modern life [1].



Figure 1. Hardware and software examples

2. Evolution of Computer Science and Computers

Computer science is a relatively new field that has grown tremendously since its beginnings in the mid-20th century. The earliest computers were extremely large and expensive machines designed primarily for government and military use. Over time, rapid advances were made as researchers sought to improve computational power, storage capacity, input/output capabilities, and overall usability of digital systems.

Here is a list of computers from 20th century to 21st century:

2.1. Abacus

One of the earliest known calculating devices was the abacus, which was used by ancient civilizations such as in China as early as 2400 BCE. The abacus functions through the manual sliding of beads along rods or slots. To perform calculations, the beads are moved to represent values according to their place and quantity. Unlike modern machines, the abacus relies solely on physical manipulation by hand rather than electronic components or software programs. As a result, it is typically a smaller, handheld size that can vary in exact dimensions and building materials across different historical contexts and cultures. Despite requiring manual operation rather than automatic processing, the abacus demonstrated the utility and importance of tools that could augment human capabilities for mathematical calculations before the development of more advanced electronic computers [3].

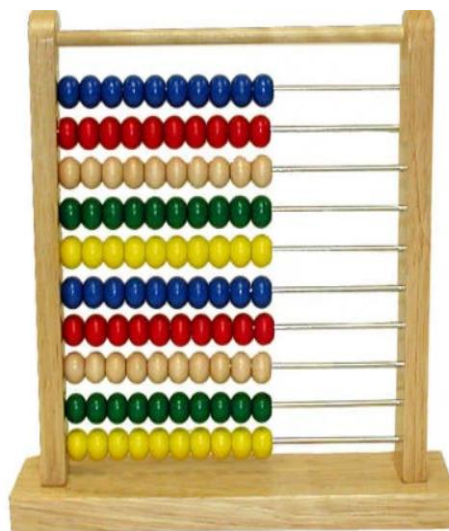


Figure 2. Abacus

2.2 Pascaline

One early mechanical calculating machine was the Pascaline, invented in 1642 by the renowned French mathematician Blaise Pascal. Weighing approximately 2.5 kilograms and with external dimensions of about 30 centimeters by 15 centimeters, the Pascaline utilized a series of interlocking gear wheels to perform basic arithmetic functions such as addition and subtraction. As one of the first non-abacus tools for calculation, it demonstrated the potential of machines to take over repetitive mathematical tasks through mechanical means rather than manual bead sliding and place-value representation. By incorporating gears that could be positioned and turned, the Pascaline helped lay the conceptual foundations for future programmable calculators and computers that operated not through direct physical manipulation but via the automated processing capabilities of engineered parts working in sequence. While a comparatively simple invention, it showed the dawn of calculation innovation through mechanical computation rather than solely mental or manual methods [4].



Figure 3. Blaise Pascale



Figure 4. Pascaline

2.3 Difference Engine:

Another landmark early calculating machine was Charles Babbage's Difference Engine, for which he began creating detailed designs in 1822. Though never fully completed during Babbage's lifetime, a functioning version was finally built in 1991 based on his plans and drawings. Weighing approximately 2.6 tons and occupying a space of around 2.4 meters by 2 meters by 0.6 meters, the Difference Engine represented an enormous mechanical calculator far larger than previous inventions. Through a system of interconnected cogwheels and gears, it was capable of tabulating and

printing polynomial functions and other repetitive calculations - establishing the concept of a programmable machine before the concepts of algorithms and computers were fully developed. While enormous in size compared to today's pocket-sized devices, the Difference Engine exemplified the growing potential of using engineered mechanisms rather than human effort alone to efficiently perform complex calculations at scale. It helped advance the capabilities of machines from basic arithmetic to more extensive mathematical workflows [5].



Figure 5. Charles Babbage

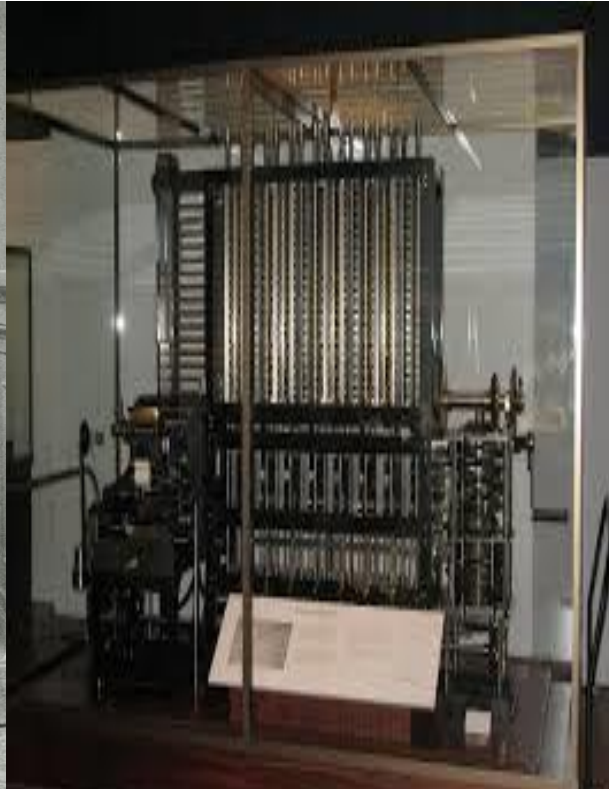


Figure 6. Difference Engine

2.4 ENIAC

A milestone in the development of electronic, programmable computers was ENIAC (Electronic Numerical Integrator and Computer), which was completed in 1946. Created by engineers John Mauchly and J. Presper Eckert at the University of Pennsylvania, ENIAC represented a colossal advancement over previous mechanical and electro-mechanical calculators. Weighing approximately 27 tons and occupying a floor space of around 167 square meters, the mammoth machine demonstrated for the first time the potential of fully electronic, hardware-based programmable computation. Through its use of over 18,000 vacuum tubes and 1,500 relays to perform mathematical and logical operations, ENIAC paved the way for stored-program digital computers by allowing complex routines to be coded and run

automatically without re-wiring. While tremendously large and energy-intensive compared to modern devices, ENIAC established the viability of building general-purpose computing technology that surpassed the functions of dedicated mechanical calculators through electronic logic controlled by software instead of unchangeable external hardware configurations. This proved a crucial step toward the digital revolution to come [6].



Figure 7. ENIAC Machine



Figure 8. John Mauchly and J. Presper

2.5 UNIVAC I

An important early commercial computer was the UNIVAC I, which was revealed to the public in 1951. Manufactured by Remington Rand, the UNIVAC I represented some key innovations that moved computing technology beyond specialized military and research applications into broader business and commercial sectors for the first time. At approximately 13 tons and occupying 35.5 square meters of floor area, the UNIVAC I built upon earlier electronic computer designs while introducing magnetic tape as the main memory and storage medium, significantly increasing data capacity. This established the means for routine data recording, processing, and report generation that opened a host of new business and administrative uses beyond scientific calculation. While still relatively enormous compared to later personal machines, the UNIVAC I demonstrated computing's commercial viability and paved the way for the mainframe computer industry. It played an important role in establishing the foundational technologies, especially magnetic storage, that enabled computers to take on diverse roles beyond research and begin automating industry processes [7].



Figure 9. Univac I

2.6 IBM 701:

One of the earliest scientific computers produced commercially was the IBM 701, released in 1952. As with many of the computing machines that preceded it, the IBM 701 was an immense system that required substantial floor space and infrastructure. Weighing several tons, the 701 could only be reasonably housed within an entire large room rather than typical workstation desk space. Despite its gigantic size compared to later microchips and personal devices, the IBM 701 represented an important milestone as it showed how specialized computation technology could serve scientists and engineers through programs developed specifically for mathematical, scientific, and engineering problems. While bulky and expensive to purchase and maintain, the 701 helped establish the IBM corporation as a leading force in early commercial computing and demonstrated a new generation of hardware designed for domains beyond just calculation, moving towards simulation, modeling, and number-crunching applications ahead of its time [8].



Figure 10. IBM 701

2.7 PDP 1

An important transition in the size and affordability of computers was the PDP-1, introduced in 1959 by Digital Equipment Corporation (DEC). Considered the first true minicomputer, the PDP-1 broke new ground by being significantly smaller and less expensive than earlier mainframe-scale machines, yet still offering the interactive processing capabilities of a contemporary mainframe computer. Weighing approximately 900 kilograms and occupying a floor space of around 4.9 square meters, the PDP-1 demonstrated that full-featured computer systems no longer required the enormity of previous room-filling predecessors. It helped usher in a new generation of specialized systems designed for efficiency, though still paling in comparison to the microchips of today. The PDP-1 established DEC as an industry leader and provided computing solutions for science and engineering at universities and research facilities with more modest budgets. It helped make computers smaller, more accessible tools beyond just corporate and government domains [9].



Figure 11. PDP 1

2.8 IBM System/360

A breakthrough that helped establish the modern model of modular component-based computing systems was the IBM System/360 line, introduced in 1964. As the first family of computers designed to be scalable, upgradeable and compatible with each other, the System/360 models represented a major advance over one-off proprietary designs like predecessors. The various System/360 CPU models could support a common base of peripheral devices and run the same operating system software regardless of processor speed or memory capacity. While the weight and physical footprint still varied significantly depending on the specific model—often requiring considerable floor space comparable to previous

room-filling mainframes—the System/360 established the concept of uniformly developing technologies into integrated families of processors, devices and software. This innovation paved the way for common standards and infrastructure that have persisted across generations of business computing—a crucial precursor to today's highly modular, interoperable computing ecosystems [10].

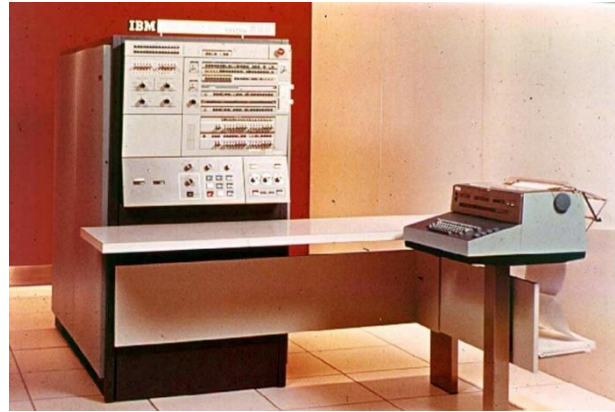


Figure 12. IBM system/360

2.9 Altair 8800

A major milestone in ushering in the personal computer era was the Altair 8800, released in 1975 by MITS. As the first commercially successful mini-computer kit available for hobbyist builders, the Altair 8800 is considered the spark that ignited the fledgling personal computer industry. Weighing approximately 15 kilograms and measuring 44cm by 17cm by 30cm, the Altair model used a Digital Equipment Corporation CPU board paired with RAM and I/O interfaces. Its small size, relative affordability of a few hundred dollars, and availability as a kit for individual assembly began opening access beyond industrial and academic circles. Altair helped motivate developers like Bill Gates and Paul Allen to launch Microsoft and deliver the popular BASIC programming language that made computers more approachable for non-experts. While limited in capability compared to later commercial PCs, the Altair 8800 played an influential role in catalyzing the home computing revolution by demonstrating viable do-it-yourself platforms [11].



Figure 13. Altair 8800

2.10 Apple II

A hugely influential early personal computer was the Apple II, introduced in 1977. Manufactured by the startup company Apple Computer under the leadership of Steve Wozniak and Steve Jobs, the Apple II was one of the first microcomputers to achieve significant commercial success. Measuring approximately 28 centimeters by 46 centimeters by 14 centimeters and weighing around 5.4 kilograms, the Apple II represented a major reduction in size compared to previous single-user machines. Its mass-market orientation helped spearhead a new generation of affordable, accessible home computers through integrations like color graphics and a BASIC programming language. With thousands of units sold in the first year, the Apple II demonstrated the mainstream viability of desktop-sized personal computing for tasks beyond hobbyist engineering. It helped establish Apple as a pioneer of the personal computer era and paved the way for the widespread adoption of microcomputers as work, educational and entertainment tools for average consumers in the decades ahead [12].



Figure 14. Apple II

2.11 IBM PC (1981)

A pivotal moment in establishing personal computing globally was the release of the IBM PC in 1981. Developed by IBM with major components from Intel, the IBM PC helped define the modern landscape of the personal computer industry. Measuring approximately 50 centimeters by 41 centimeters by 14 centimeters and weighing around 11 kilograms without a monitor, the IBM PC proved consumer-friendly in size and cost while surpassing other platforms in capabilities. Its open architecture allowed independent manufacturers to develop compatible accessories and software, rapidly expanding the ecosystem. The influence of Big Blue's entrance legitimized PCs as business machines beyond home experimentation. The IBM PC's design became the de facto industry standard, cementing Intel's dominance and Microsoft's DOS and Windows as the predominant operating system. While incremental in features compared to predecessors, the IBM PC spawned a multi-billion-dollar market and firmly established personal computers as a mainstream technology that transformed global productivity and society [13].



Figure 15. IBM PC (1981)

2.12 Apple Macintosh (1984)

A revolutionary development in personal computing usability was Apple's Macintosh introduction in 1984. Boasting a mere 7.5 kilogram weight and occupying a minimal footprint of approximately 34.5 centimeters by 24.4 centimeters by 27.7 centimeters, the Macintosh built upon the Apple II legacy with its new graphical user interface driven by a mouse pointer rather than esoteric command lines. In what was also a major marketing event, the Macintosh became the first successful commercialization of an easy-to-use graphical desktop environment. Its development under the guidance of Jef

Raskin and Bill Atkinson brought a user-friendly paradigm that helped usher in mainstream adoption worldwide by those without technical expertise. While limited in software compatibility and relatively expensive for the time, the Macintosh established Apple as an innovator once more and demonstrated the immense potential of intuitive visual computing interfaces that have become ubiquitous today. It transformed expectations for personal devices entirely [14].



Figure 16. Apple Macintosh (1984)

2.13 Modern Laptops and Smartphones (21st century):

While the earliest personal computers filled entire rooms, the dawn of the 21st century saw mobile devices shrink technology into ever more portable form factors. Laptop computers gradually became lighter and more compact, making intensive computing possible anywhere within a lightweight clamshell design. Meanwhile, smartphones merged ever more functions into handheld devices, starting with simple voice and text capabilities before adding powerful multimedia, internet, and app capabilities that eclipsed older personal devices. Modern laptops and smartphones demonstrate just how far technology has progressed, with incredibly powerful systems that fit comfortably in a briefcase or pocket. They reflect the culmination of decades of relentless miniaturization that have delivered general-purpose personal computing and constant connectivity to serve all facets of modern life. Today's slim, wireless devices capable of endless apps and functions would seem like science fiction to early pioneers who introduced computing to solve special-purpose tasks in machine rooms. The mobility revolution continues to push boundaries and redefine expectations of what personal technology can achieve.



Huawei
Mate X



Lenovo Yoga 360



Gamer PC

Figure 17. Modern Computers

3. Principles of How a Computer Functions

A computer operates through a series of well-defined steps that involve various internal components and processes. At the center of it all is the central processing unit (CPU), often described as the "brain" of the computer. The CPU coordinates with different types of memory such as RAM and cache to fetch instructions and data as needed for various tasks.

When a user or external device provides input, this can come in the form of commands, data or signals. The input is first processed by the control unit, which decodes instructions and directs the overall flow of operations. It ensures the CPU carries out calculations and logic in the correct sequence.

Central to processing is the arithmetic logic unit (ALU), a critical component of the CPU. The ALU performs basic arithmetic like addition and subtraction as well as logical operations like comparisons. It does so by manipulating input data as guided by the instructions obtained from memory.

Once the CPU completes its processing, any outputs are directed to different potential destinations. This can include monitors for displaying visual outputs, speakers for sound, as well as storage devices for saving files long-term. Temporary outputs are first stored in memory for fast access before being committed to permanent storage disks.

By cycling through input, processing, output and storage using its purpose-built components, the CPU can simulate complex programs and multi-task different workloads simultaneously. This provides the foundation for computers to serve as powerful and versatile machines we interact with daily.

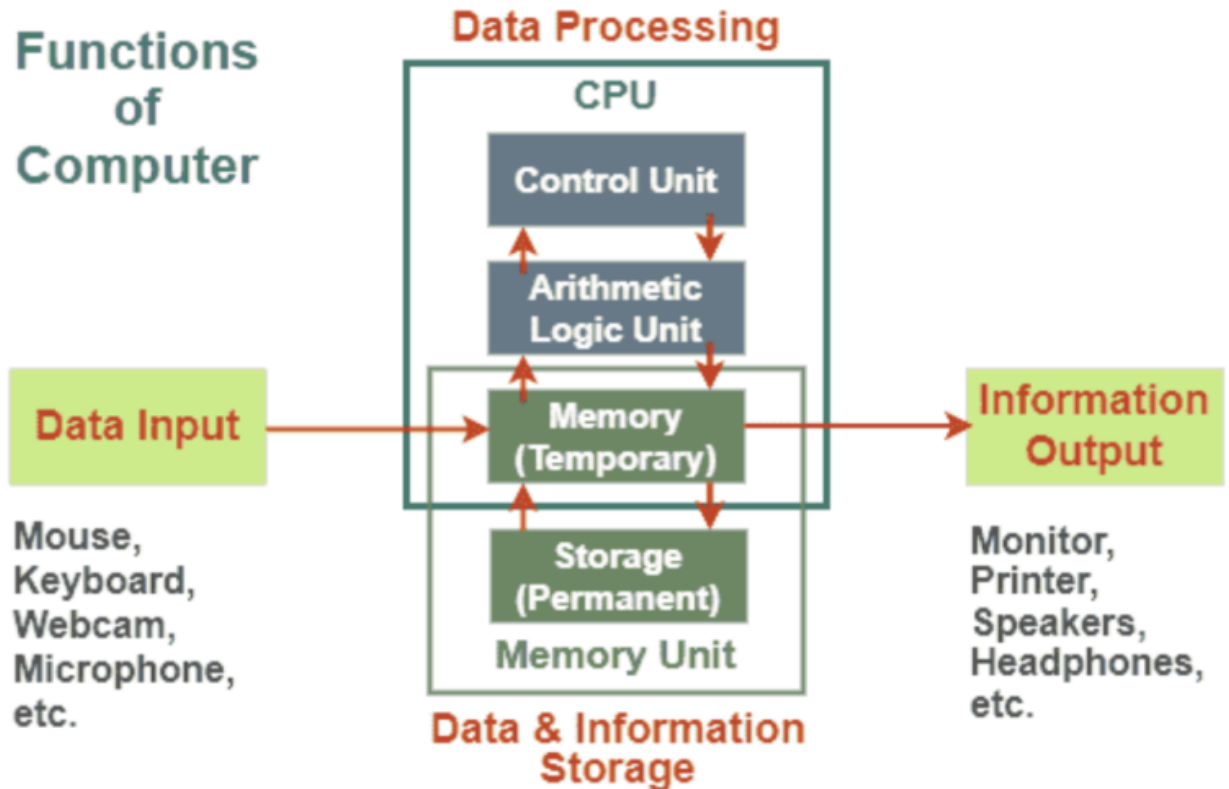


Figure 18. Functionality of a Computer [2]

➤ So, we have:

Input: The process begins when a user or peripheral device, such as a keyboard, mouse, or sensor, provides input to the computer. This input can be in the form of data, commands, or signals.

Storage (Short-Term): The input is temporarily stored in the computer's memory, which can be RAM (Random Access Memory) or cache memory. RAM is used for short-term storage and holds the data that the computer is currently using or processing.

Processing: The central processing unit (CPU), often considered the "brain" of the computer, performs calculations and executes instructions. It fetches

data from memory, processes it according to instructions from software, and then stores the results back in memory.

Control Unit: The control unit coordinates and manages the operations of the CPU. It fetches instructions from memory, decodes them, and directs the execution of those instructions.

Arithmetic Logic Unit (ALU): The ALU is part of the CPU responsible for performing mathematical calculations and logical operations. It can add, subtract, multiply, divide, compare values, and perform other fundamental arithmetic and logical tasks.

Output: After processing, the computer may produce output in various forms, such as text, images, sound, or control signals. This output can be directed to a monitor, printer, speakers, or other output devices.

Storage (Long-Term): Data that needs to be preserved beyond the current session is stored on long-term storage devices, such as hard drives, solid-state drives (SSDs), or external storage media. These devices store data persistently even when the computer is turned off.

4. Hardware Components of a Computer

Computer hardware can be classified into four main categories based on their function:

I. Input Devices:

Input devices are used to enter data into the computer. Common examples include keyboards and mice. Input devices translate human-readable data into forms the computer can understand for processing.

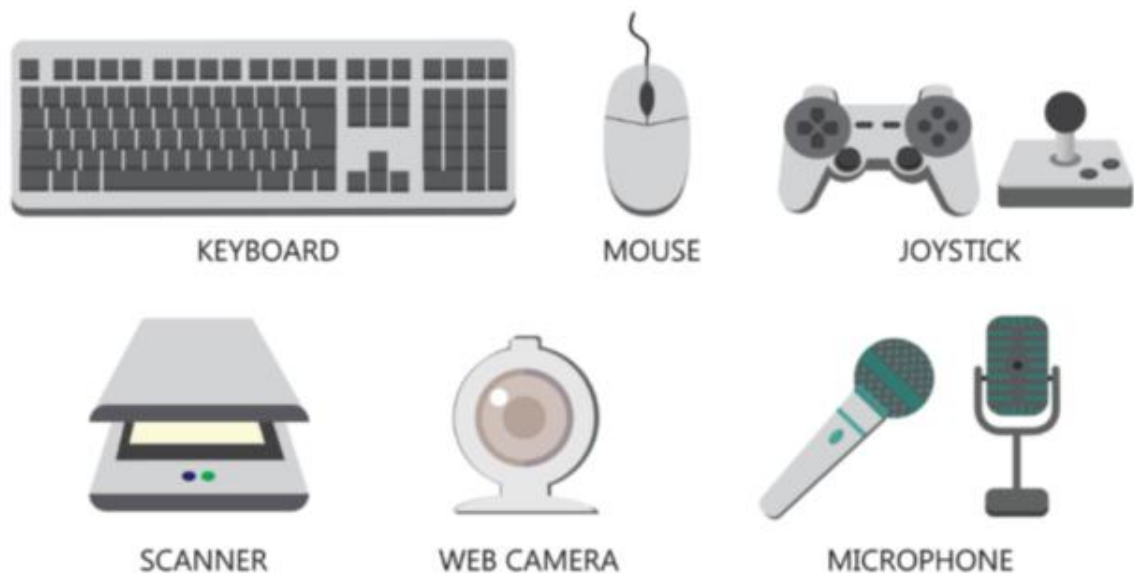
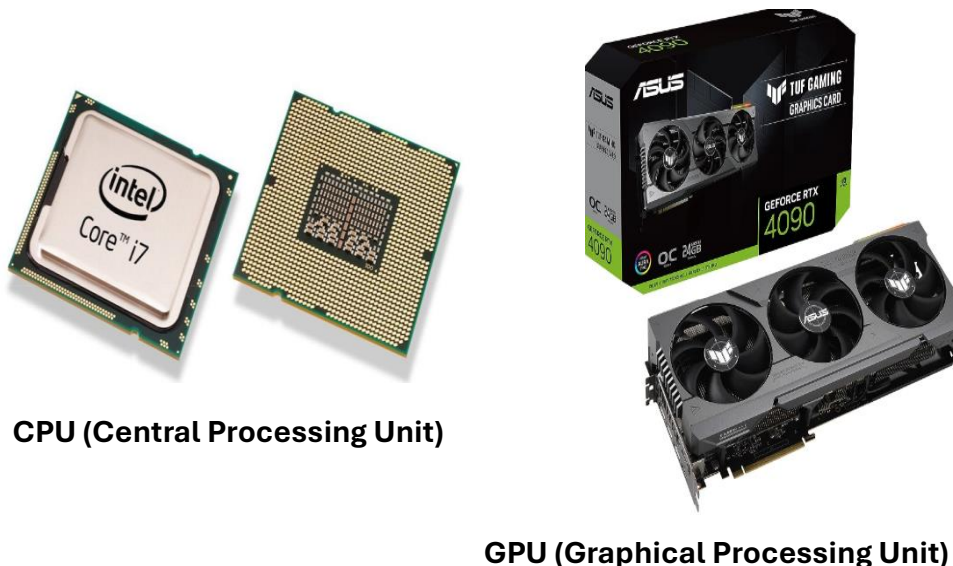


Figure 19. Input devices examples

II. Processing Devices:

The Central Processing Unit (CPU) is the primary processing device in a computer. It interprets and executes instructions from programs to perform tasks. Graphics Processing Units (GPUs) are specialized processors designed to handle graphics processing like game rendering and scientific computations.



CPU (Central Processing Unit)

GPU (Graphical Processing Unit)

Figure 20. Processing Units

III. Output Devices:

Output devices disseminate and display data and information in human-readable forms. Examples include computer monitors, printers and speakers. They convert electronically generated output into visual, audio or physical formats.



Figure 21. Output devices examples

IV. Storage Devices:

Storage devices retain data and information even when the computer is powered off. Primary memory like RAM works closely with the CPU but is volatile. Secondary memory like hard drives provides permanent non-volatile storage. Common secondary storage options include hard disk drives (HDD), optical discs for removable media, and solid-state drives.



Figure 22. Storage devices examples

- **Connectivity:**

Computers connect to external devices through various ports and wireless technologies. Universal Serial Bus (USB), HDMI, DisplayPort, audio jacks, Ethernet ports and wireless adapters facilitate communication with peripherals and other systems. Together, these internal components and connection methods allow computers to function as powerful information processing machines.

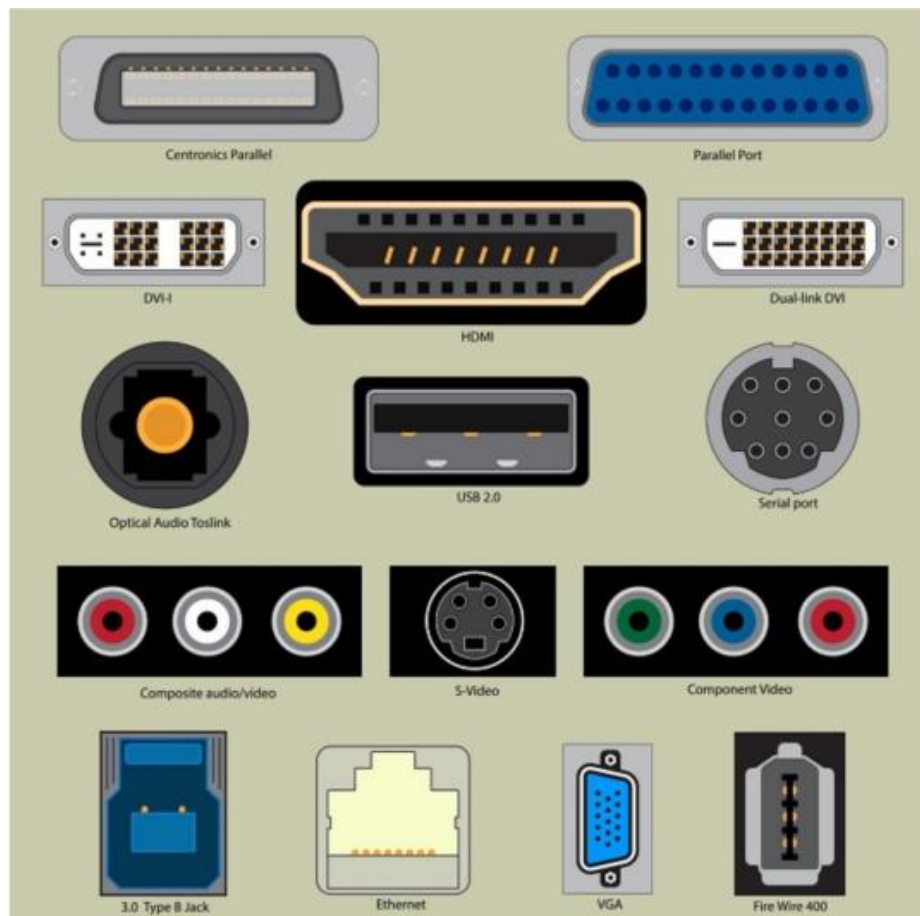


Figure 23. Examples of computer ports

In addition to the central processing units, memory, and storage drives, several other key components come together to form a fully functional computer system. At the foundation is the motherboard, which serves as the main circuit board that all other parts plug into. It facilitates communication between the CPU, memory, storage, ports and expansion slots. Another vital subsystem is the power supply, which receives electrical current from a wall outlet and distributes the proper voltages needed to power all internal components. Display adapters are also essential for connecting one or more

computer monitors to process and relay visual outputs. Rounding out the assembly are other miscellaneous parts like expansion cards, cooling fans and various cable/connector linkages. When expertly assembled and integrated with an operating system, this aggregation of microchips, circuitry, moving parts and peripheral ports empowers the machine to undertake complex computational tasks with high-speed processing capabilities and expandability. The strategic cooperation of the motherboard, power supply and additional modules thus culminates in a fully featured general purpose computer.

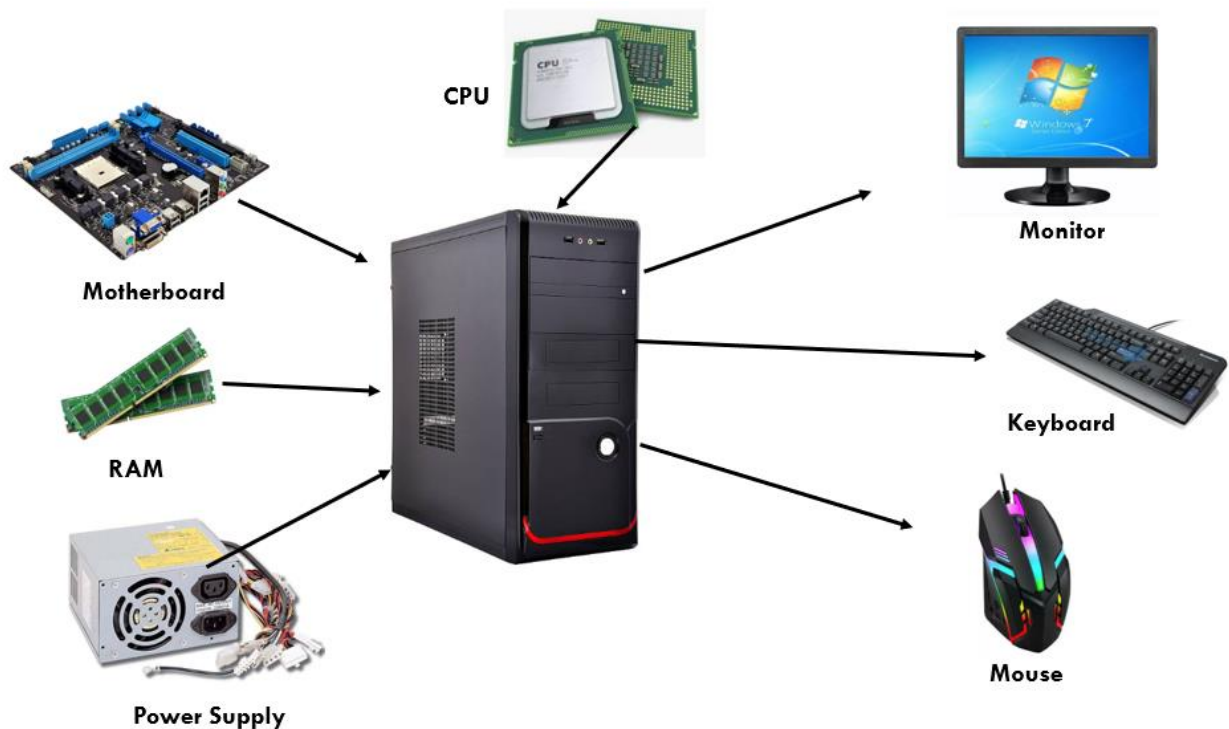


Figure 24. Main hardware components of a computer

5. Software components of a computer

Computer software can generally be categorized into two main types: system software and application software.

System software: includes programs that handle basic underlying computer functions rather than specific tasks. It controls lower-level operations and integrates hardware components. This allows other software and users to interact with the computer without knowing technical details.

The main components of system software are the **operating system** and **drivers**. The operating system acts as an interface between the user and hardware. It manages resources and allows interaction. **Basic Input/Output**

System (BIOS) firmware initializes hardware during startup. **Device drivers** enable communication with specific pieces of hardware.

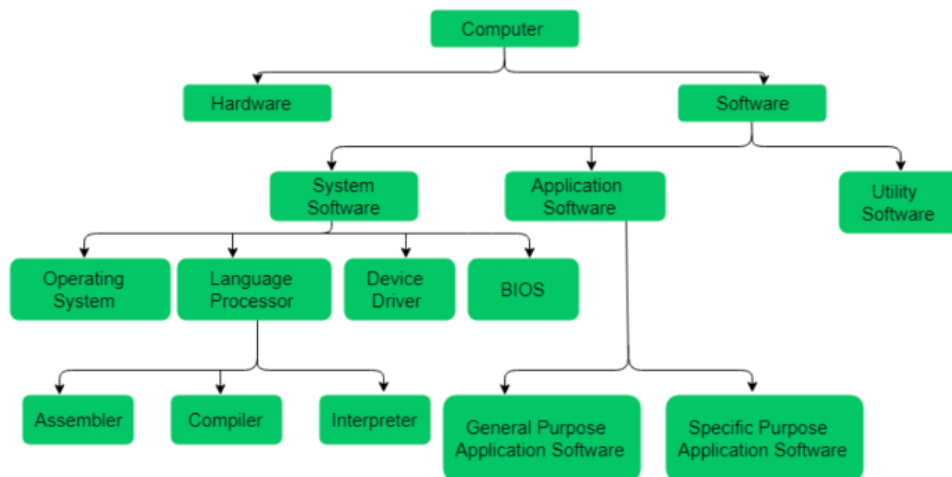


Figure 25. Software Components



Figure 26. Operating systems examples

Application software: is used to perform specialized tasks beyond basic operation. It includes individual programs like image viewers as well as integrated software packages and suites. Examples are word processors, spreadsheets, databases, web browsers and games.

Programming languages: are essential tools that allow humans to communicate instructions to computers. Popular languages include Python, Java, C++ and many others. Each has unique syntax, rules and strengths suiting different tasks and preferences. They provide structured ways to write code and build applications, websites and software solutions.



Figure 27. Application software examples



Figure 28. Programming languages examples

Language processors: translate code between human-readable and machine-executable formats. **Compilers** convert entire programs before running. **Interpreters** translate each line as the program runs. **Assemblers** convert assembly code to machine code. These language processors are core to software development.

In summary, system software handles underlying functions while application software empowers users. Programming languages facilitate human-computer communication. Together this software enables modern computing abilities.

6. Systems for Encoding Information

All information processed by computers, whether numeric, text, images, sounds or videos, is represented internally using binary digits or bits. A bit is the most basic unit of digital information in computing and can have one of two possible values 0 or 1.

Physically, computers encode these binary digits using two different electronic states. The digit 1 is represented by the presence of an electrical pulse within circuits, while 0 corresponds to the absence of a pulse. Having only these two states - pulse or no pulse - allows computers to efficiently encode all types of analog information into digital formats that can then be manipulated computationally.

Through complex arrangements and interpretations of long strings consisting entirely of zeros and ones, computers are able to acquire, store, process and output myriad data types in binary form. Everything a computer works with from numbers to word documents to high resolution photos breaks down fundamentally to sequences of electric pulses and their absence housed within intricate microchip circuitry. This binary system provides the foundation for modern computing and has enabled digital technologies to permeate nearly all aspects of contemporary society.

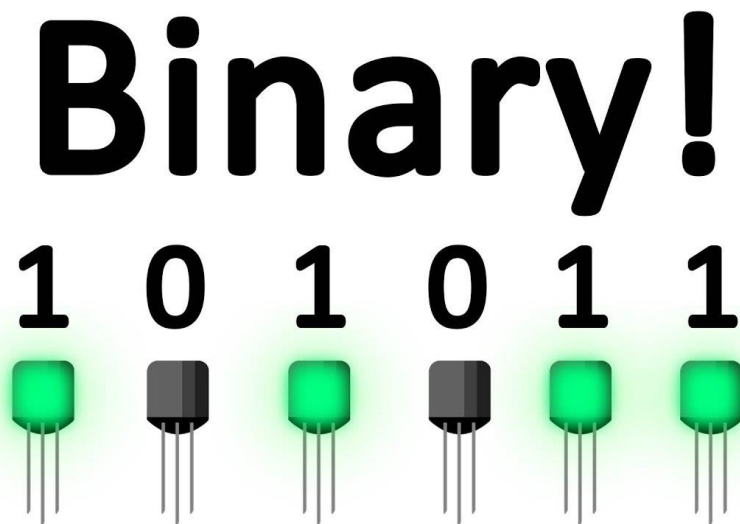


Figure 29. Binary representation

6.1 Data storage:

As an example, the letter "M" can be encoded using the binary combination of 1001101.

Each element in this combination, either 0 or 1, is referred to as a bit. A collection of 8 bits is called a byte.

Bytes are the basic units used to measure digital memory and storage in computing. The units and their relationships are:

- 1 Bit = The smallest possible unit, with a value of either 0 or 1
- 8 Bits = 1 Byte

So, in short, individual zeros and ones are bits, and a group of 8 bits comes together to form a single byte - the standard unit used to represent a single character or other small element of data in computer systems. This allows all types of information to be broken down, stored and reconstructed using only two numeric values.

Data storage amounts in computing double exponentially as the units increase. Here are the common standardized units and their relationships:

- 1 Kilobyte (KB) = 1,024 Bytes
(2^{10} Bytes, where 2 is the exponent representing each doubling)
- 1 Megabyte (MB) = 1,024 Kilobytes
(2^{20} Bytes, where Mega means the units have doubled twice from the byte level)
- 1 Gigabyte (GB) = 1,024 Megabytes
(2^{30} Bytes, with Giga indicating three doublings from bytes)
- 1 Terabyte (TB) = 1,024 Gigabytes
(2^{40} Bytes, with Tera signifying four doubling periods have occurred)

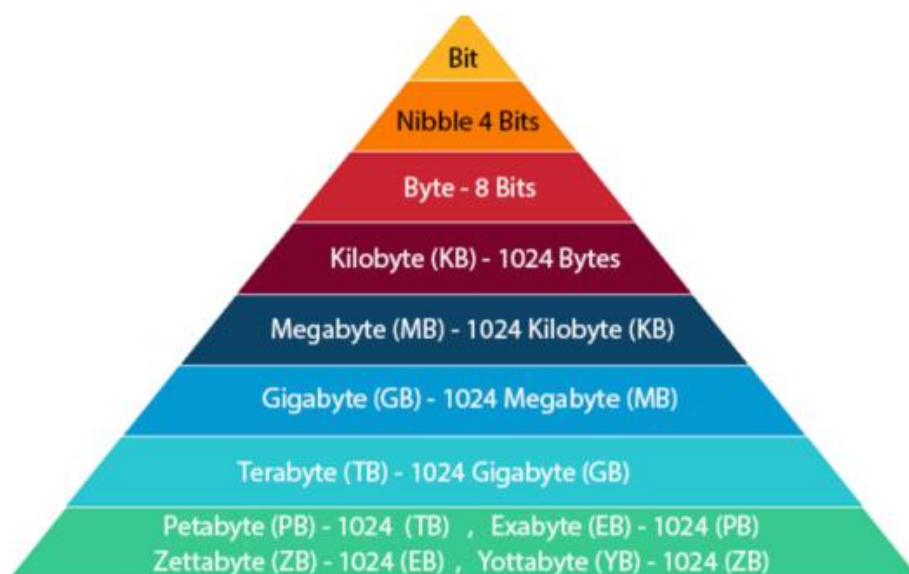


Figure 30. Units of storage

So in summary, as we move up the sequence from bytes to kilobytes to megabytes and beyond, the storage capacity doubles by a power of two at each step, providing an efficient scaling system to quantify digital information storage from small to extraordinarily large amounts.

6.2 Numbering systems

All computer operations are based on the fundamental principles of binary arithmetic (1s and 0s). While humans commonly use the decimal system for calculations, computers cannot because electronic circuits can only distinguish between two states.

Number systems differ based on their numerical base:

- Decimal (base 10) has 10 elements ranging from 0-9, which we learn in school.
- Binary (base 2) has only two elements, 0 and 1, called bits. This maps naturally to the two electric states computers can represent.
- Octal (base 8) uses eight symbols from 0-7.
- Hexadecimal (base 16) uses sixteen values from 0-9 plus A-F.

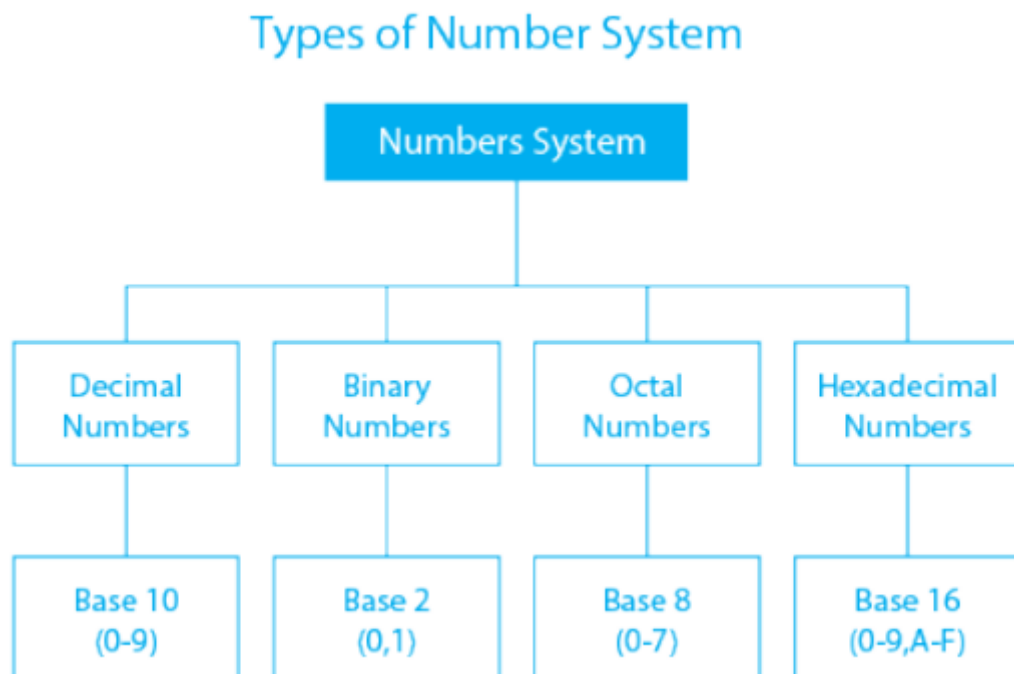


Figure 31. Numbering Systems

Although decimal feels natural, computers internally use binary because of its simplicity - only two states are required versus ten in decimal. The higher bases of octal and hexadecimal are also sometimes employed as shorthand

for binary values, making very long strings more concise for human readability. Ultimately though, all digital number representations map back to the fundamental binary language of electronics.

6.3 Conversion from One Base to Another

a. From Binary to Decimal:

Converting a binary number to its decimal equivalent involves place value.

For example, let's convert the binary number $(101101)_2$:

1. Write out each bit and its corresponding place value in decimal:

Bit 1: 1 (at place value of 32)
Bit 2: 0 (at place value of 16)
Bit 3: 1 (at place value of 8)
Bit 4: 1 (at place value of 4)
Bit 5: 0 (at place value of 2)
Bit 6: 1 (at place value of 1)

Number:	1	0	1	1	0	1
Power:	2^5	2^4	2^3	2^2	2^1	2^0
Weighting:	32	0	8	4	0	1

2. Multiply each bit by its place value and add the results:

$1 \times 32 = 32$
 $0 \times 16 = 0$
 $1 \times 8 = 8$
 $1 \times 4 = 4$
 $0 \times 2 = 0$
 $1 \times 1 = 1$

3. Total the place value weights:

$$32 + 0 + 8 + 4 + 0 + 1 = (45)_{10}$$

Therefore, the decimal equivalent of the binary number (101101) is 45.

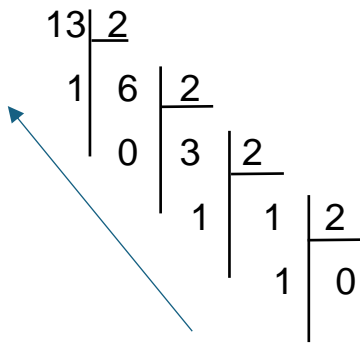
This process of assigning place values according to exponent and summing the weighted bit values converts between binary and decimal number representations. Understanding place value is key to interconverting between these numeric systems.

b. Decimal to Binary Conversion:

Converting a decimal number to binary involves repeatedly dividing by 2.

For example, to convert the decimal number $(13)_{10}$:

1. Initialize with the decimal number: 13
2. Divide 13 by 2: $13 / 2 = 6$ with remainder 1
3. The remainder 1 is the last bit in the binary.
4. Continue dividing the result by 2: $6 / 2 = 3$ with remainder 0
5. The remainder 0 is the next bit in the binary.
6. Divide 3 by 2: $3 / 2 = 1$ with remainder 1
7. The remainder 1 is the previous bit in the binary.
8. Divide 1 by 2: $1 / 2 = 0$ with remainder 1
9. The remainder 1 is the first bit in the binary.
10. The binary equivalent is 1101.

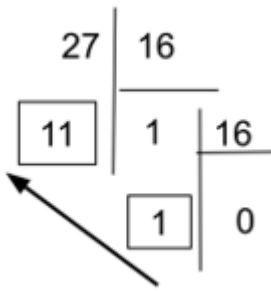


$13 \div 2 = 6$	Remainder 1: 1	↑
$6 \div 2 = 3$	Remainder 2: 0	
$3 \div 2 = 1$	Remainder 3: 1	
$1 \div 2 = 0$	Remainder 4: 1	

By successively dividing the decimal number by 2 and noting the remainders, we obtain the place values that make up the corresponding binary number. This conversion method relies on decomposing the decimal into constituent binary place values.

c. **Decimal to Hexadecimal Conversion**

We perform successive divisions by 16. The remainders will be between 0 and 15.



We start from the remainder of the last division to the first.

$$(27)_{10} = (1B)_{16} \text{ because } (11)_{10} = (B)_{16}.$$

d. Hexadecimal to Decimal Conversion:

$$(1B)_{16} = (?)_{10}$$

1	$(B)_{16} = (11)_{10}$
16^1	16^0
$(1 * 16^1) + (11 * 16^0) = (27)_{10} = (1B)_{16}$	

e. Hexadecimal to Binary and Binary to Hexadecimal Conversion:

To convert between hexadecimal and binary, we can either:

First convert to decimal using place values, then to the target base

Or:

Use a digit-to-bit or bit-to-digit mapping table.

For example, to convert $(9AE1)_{16}$ to binary:

$$(9)_{16} = (1001)_2$$

$$(A)_{16} = (1010)_2$$

$$(E)_{16} = (1110)_2$$

$$(1)_{16} = (0001)_2$$

Therefore, $(9AE1)_{16} = (1001 1010 1110 0001)_2$.

Binary	Decimal	Hex
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Figure 31. Binary-Hexadecimal table

Using decimal as an intermediate step or reference tables allows direct translation between these base systems without needing to fully calculate place values each time. This provides an efficient way to interconvert between hexadecimal, binary and octal number representations.

f. Octal to Binary and Binary to octal:

Converting between octal and binary numbers follows a similar process as hexadecimal, but with a few key differences.

For octal, we group the binary digits into blocks of 3 bits each, since octal has a base of 8 (2^3).

3-bit Binary Number	Octal Number
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Figure 32. Binary-Octal table

Example : Binary to Octal

Now, let's convert 10 110 010 (binary) to octal.

Step 1: Group the binary digits into sets of three, starting from the right
(010 110 010)₂

(Note: We added a leading zero to make a complete group of three on the left)

Step 2: Convert each 3-bit group to its octal equivalent

$$(010)_2 = (2)_8$$

$$(110)_2 = (6)_8$$

$$(010)_2 = (2)_8$$

Step 3: Combine the octal digits: 262

Therefore, 10110010 (binary) = 262 (octal)

g. **Octal to decimal:**

Example:

Let's convert the octal number 237 to decimal.

1. Identify the digits: The digits are 2, 3, and 7.
2. Expand using powers of 8:

- 2×8^2
- 3×8^1
- 7×8^0

3. Calculate each term:

- $2 \times 8^2 = 2 \times 64 = 128$
- $3 \times 8^1 = 3 \times 8 = 24$
- $7 \times 8^0 = 7 \times 1 = 7$

4. Add them together:

- $128 + 24 + 7 = 159$

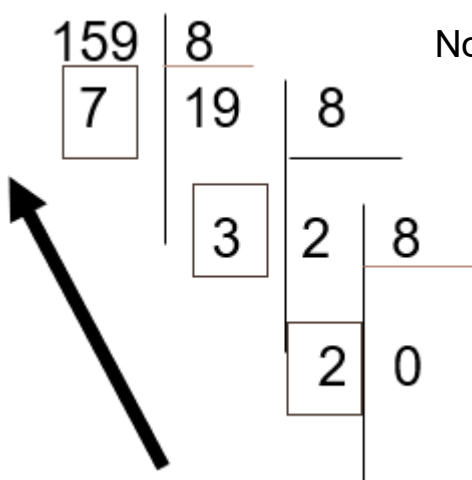
Thus, the octal number 237 is 159 in decimal.

$$(237)_8 = (159)_{10}$$

h. **Decimal to octal:**

Example:

Let's convert the decimal number 159 to octal.

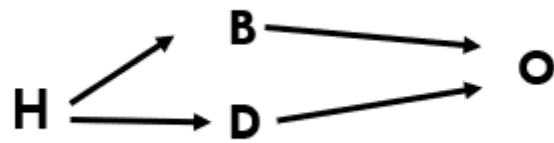


Now, reading the remainders from bottom to top gives:

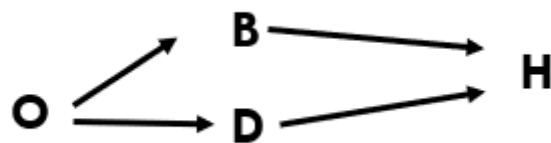
- Remainders: 2, 3, 7
 $(159)_{10} = (273)_8$

i. Hexadecimal to Octal and Octal to Hexadecimal:

To convert from Hexadecimal to Octal we can't directly, so we need first either to convert to decimal then to octal or to binary then octal:



To convert from Octal to Hexadecimal we can't directly, so we need first either to convert to decimal then to hexadecimal or to binary then hexadecimal:



References:

- [1] Zelle, J. M. (2004). Python programming: an introduction to computer science. Franklin, Beedle & Associates, Inc.
- [2] Hennessy, J. L., & Patterson, D. A. (2011). *Computer architecture: a quantitative approach*. Elsevier.
- [3] Kojima, T. (2012). *Japanese Abacus Use & Theory*. Tuttle Publishing.
- [4] Soury-Lavergne, S., & Maschietto, M. (2013). A la découverte de la «pascaline» pour l'apprentissage de la numération décimale. In C. Ouvrier-Buffet (Éd.), *XXXIXe colloque de la COPIRELEM Faire des mathématiques à l'école: De la formation des enseignants à l'activité de l'élève*.
- [5] Swade, D., & Babbage, C. (2001). *Difference engine: Charles Babbage and the quest to build the First Computer*. Viking Penguin.
- [6] Haigh, T., Priestley, P. M., & Rope, C. (2016). *ENIAC in action: Making and remaking the modern computer*. MIT press.
- [7] Weiner, J. R. (1952). Operating experience with UNIVAC SYSTEMS. *Transactions of the IRE Professional Group on Electronic Computers*, (1), 33-46.
- [8] "IBM 701 Electronic analytical control unit". IBM. 23 January 2003. Archived from the original on 21 May 2022. Retrieved 30 July 2022.
- [9] "1960: DEC PDP-1 Precursor to the Minicomputer". CED Magic. Archived from the original on January 3, 2022. Retrieved December 4, 2008.
- [10] Padeogs, A. (September 1981). "System/360 and Beyond" (PDF). *IBM Journal of Research and Development*. 25 (5). IBM: 377–390. doi:10.1147/rd.255.0377.

[11] Rojas, Raúl (2001). Encyclopedia of computers and computer history. Chicago [u.a.]: Fitzroy Dearborn. ISBN 1-57958-235-4.

[12] "June 10, 1977 - Apple II Shipped Today". This Day in History. Mountain View, CA: Computer History Museum. Archived from the original on June 20, 2012. Retrieved August 3, 2012.

[13] Pollack, Andrew (August 13, 1981). "Big I.B.M.'s Little Computer". The New York Times. p. D1. ISSN 0362-4331. Retrieved March 16, 2020.

[14] "Macintosh 128K: Technical Specifications". apple.com. Archived from the original on February 17, 2015. Retrieved March 27, 2015.