POST AND TELECOMMUICATIONS INSTITUTE OF TECHNOLOGY

An Overview of Computer Architecture

Faculty: Computer science – IT1

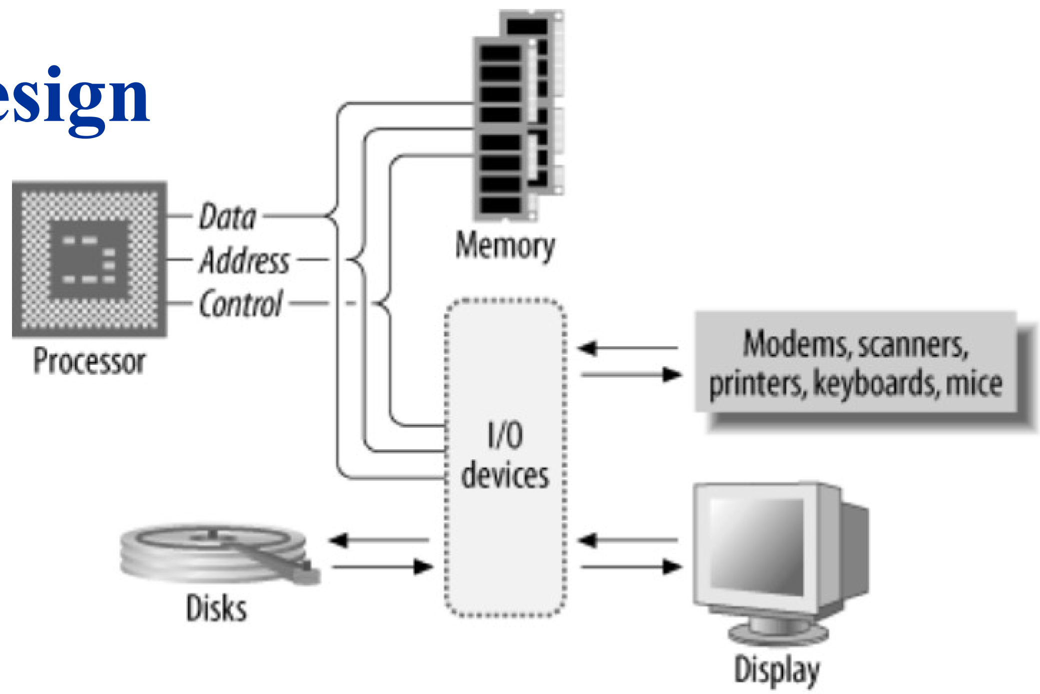
Instructor: Viet Hoang Pham

Email: vietph@ptit.edu.vn

Phone: 0945296388

Contents

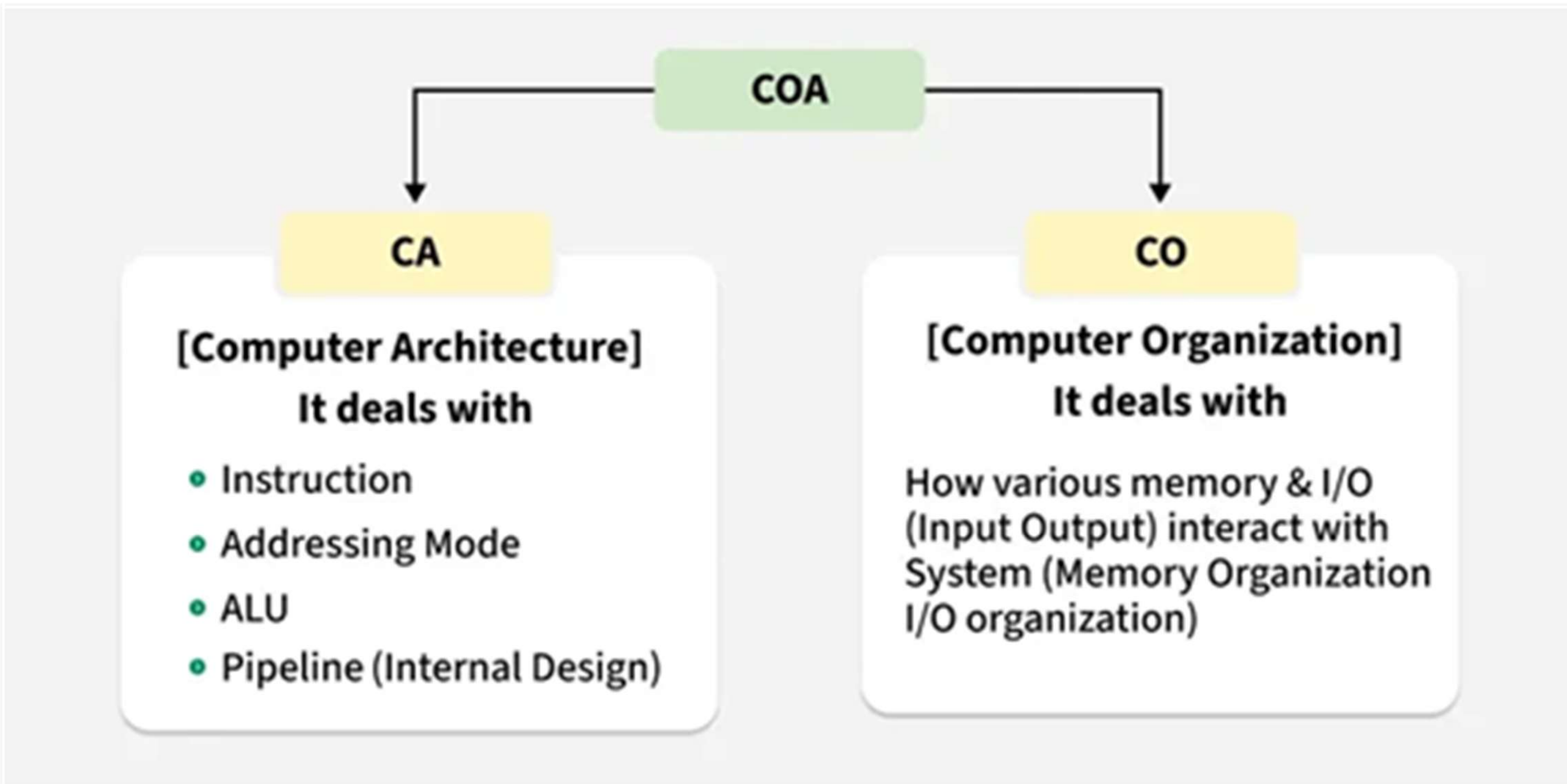
- The concept of computer architecture and computer organization
- The history of computer development
- The structure and functions of a computer
- Computer classification
- Computer architecture design
- Several number systems



Computer architecture and computer organization

Computer Organization	Computer Architecture
➢ Computer Organization deals with a structural relationship.	➢ Computer Architecture deals with the functional behavior of computer systems.
➢ Computer Organization consists of physical units like circuit designs, peripherals, and adders.	➢ Computer Architecture comprises logical functions such as instruction sets, registers, data types, and addressing modes.
➢ Computer Organization handles the segments of the network in a system.	➢ Architecture coordinates the hardware and software of the system.
➢ For designing a computer, an organization is decided after its architecture.	➢ For designing a computer, its architecture is fixed first.
➢ Computer Organization is frequently called microarchitecture.	➢ Computer Architecture is also called Instruction Set Architecture (ISA).

Computer architecture and computer organization



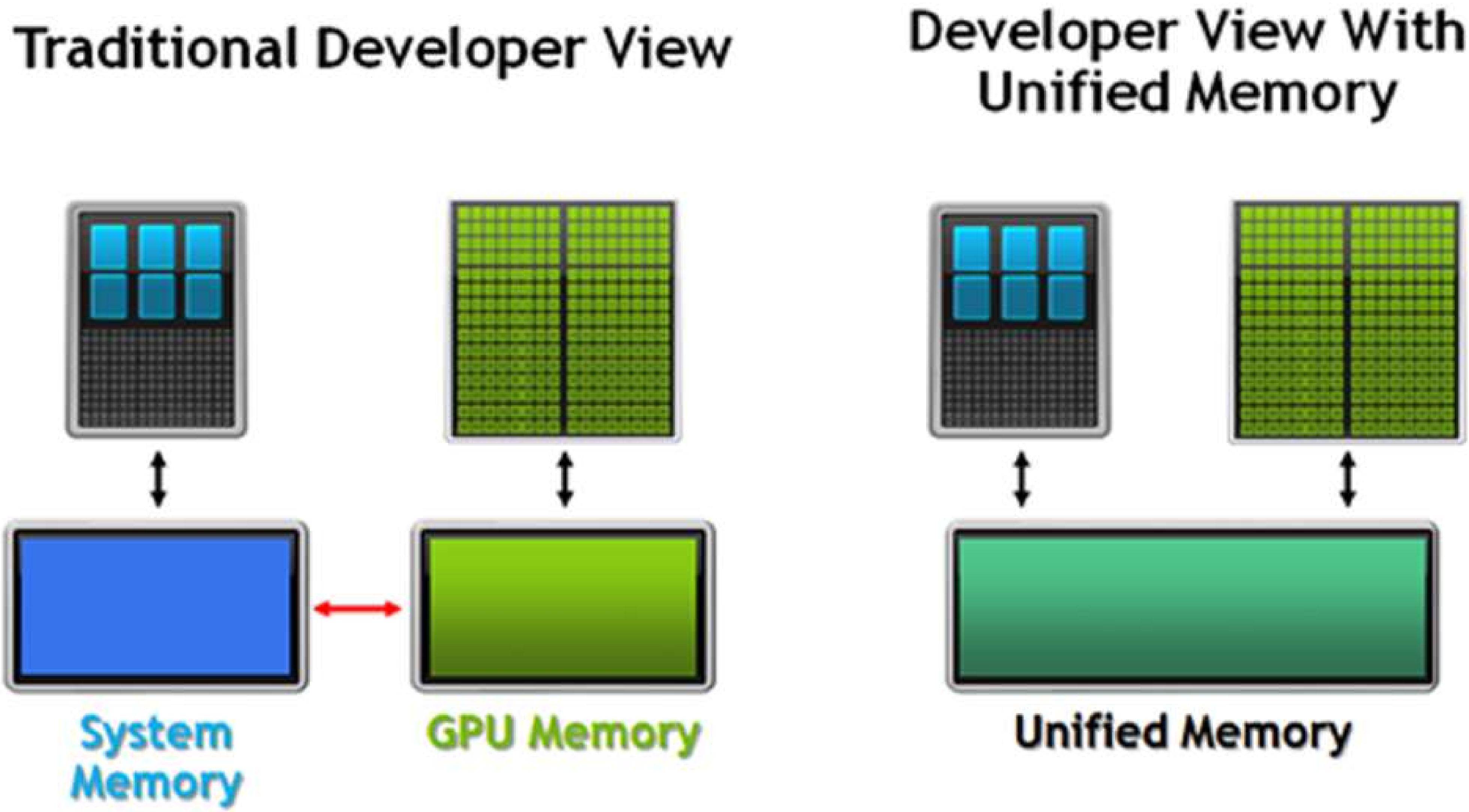
❑Illustrative example: consider the multiplication operation:

- Whether a computer is equipped with a multiplication operation is an architectural issue.
- However, whether that multiplication is implemented using a dedicated multiplier unit or by repeatedly performing additions is a matter of computer organization.

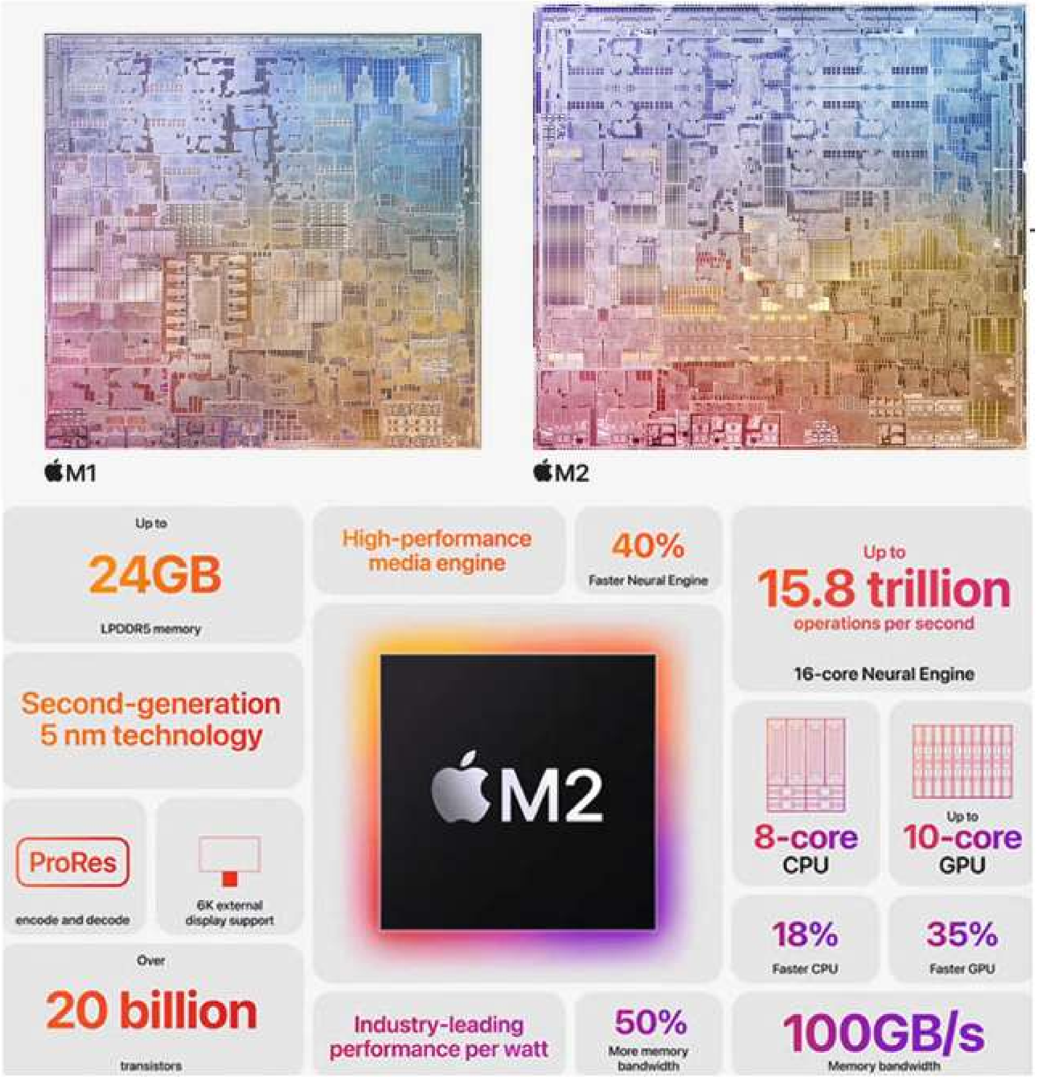
Analogy: Computer architecture can be compared to the blueprint of a house (deciding how many rooms there are and the function of each room), while computer organization corresponds to the actual construction (deciding what type of bricks to use, and how electrical wiring and plumbing are laid inside the walls to ensure that the functions specified in the blueprint are realized).

Computer architecture and computer organization

- **Computer organization**: the discipline that studies the components of a computer and how they operate based on a given architecture.
  - Example: IBM ThinkPad, iPhone, and Android phones represent system design for different product lines.
- **Computer architecture**: the discipline concerned with selecting and interconnecting computer hardware components in order to achieve the following goals:
  - **Performance**: as fast as possible
  - **Functionality**: supporting as many functions as possible
  - **Cost**: as low as possible
  - Example: different generations of machines sharing the same hardware architecture, such as the iPhone 12, iPhone 13, and iPhone 14.


Traditional Developer View
Developer View With Unified Memory
System Memory
GPU Memory
Unified Memory

Apple:

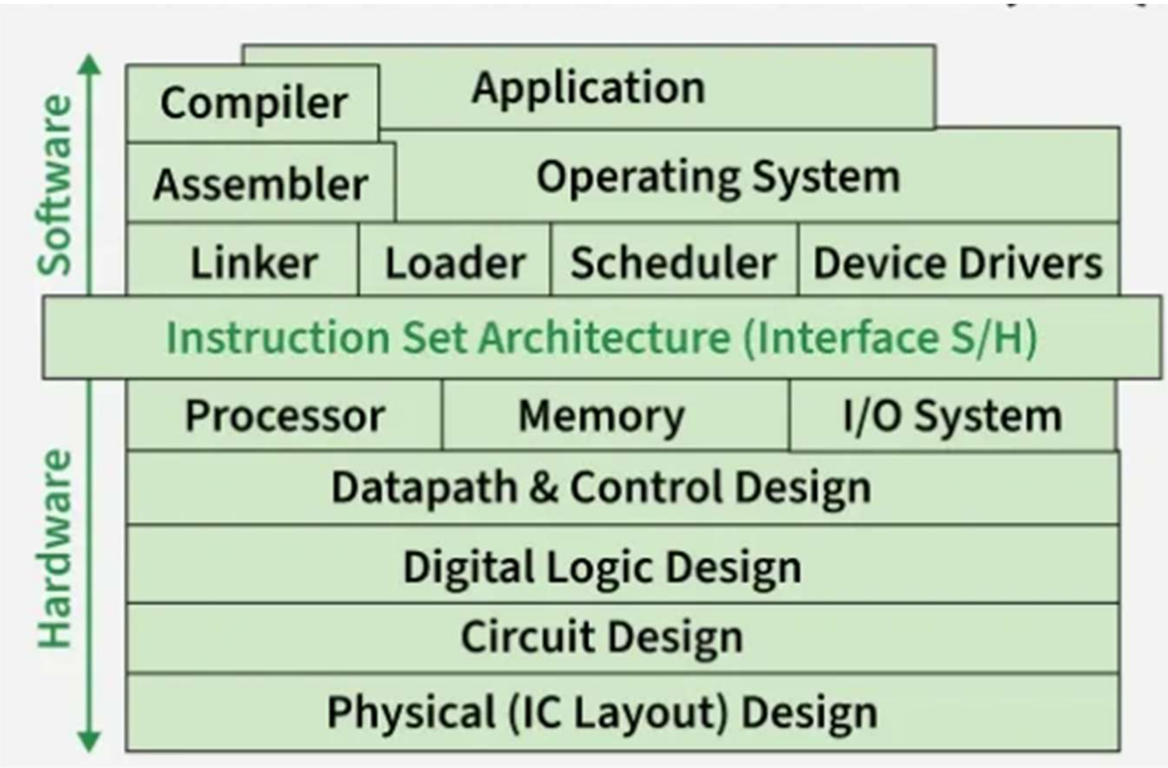

M1
M2
Up to
24GB
LPDDR5 memory
High-performance media engine
40%
Faster Neural Engine
Up to
15.8 trillion
operations per second
16-core Neural Engine
Second-generation 5 nm technology
M2
ProRes
encode and decode
6K external display support
8-core
CPU
Up to
10-core
GPU
18%
Faster CPU
35%
Faster GPU
Over
20 billion
transistors
Industry-leading performance per watt
50%
More memory bandwidth
100GB/s
Memory bandwidth

The basic components of computer architecture consist of three parts:

- ❖**Instruction Set Architecture (ISA)**
- ❖**Micro-architecture**
- ❖**System design**

❖ **Instruction Set Architecture (ISA)**: An abstract model of a computer at the machine-language (assembly-language) level, defining what the processor does and how it does it, including:

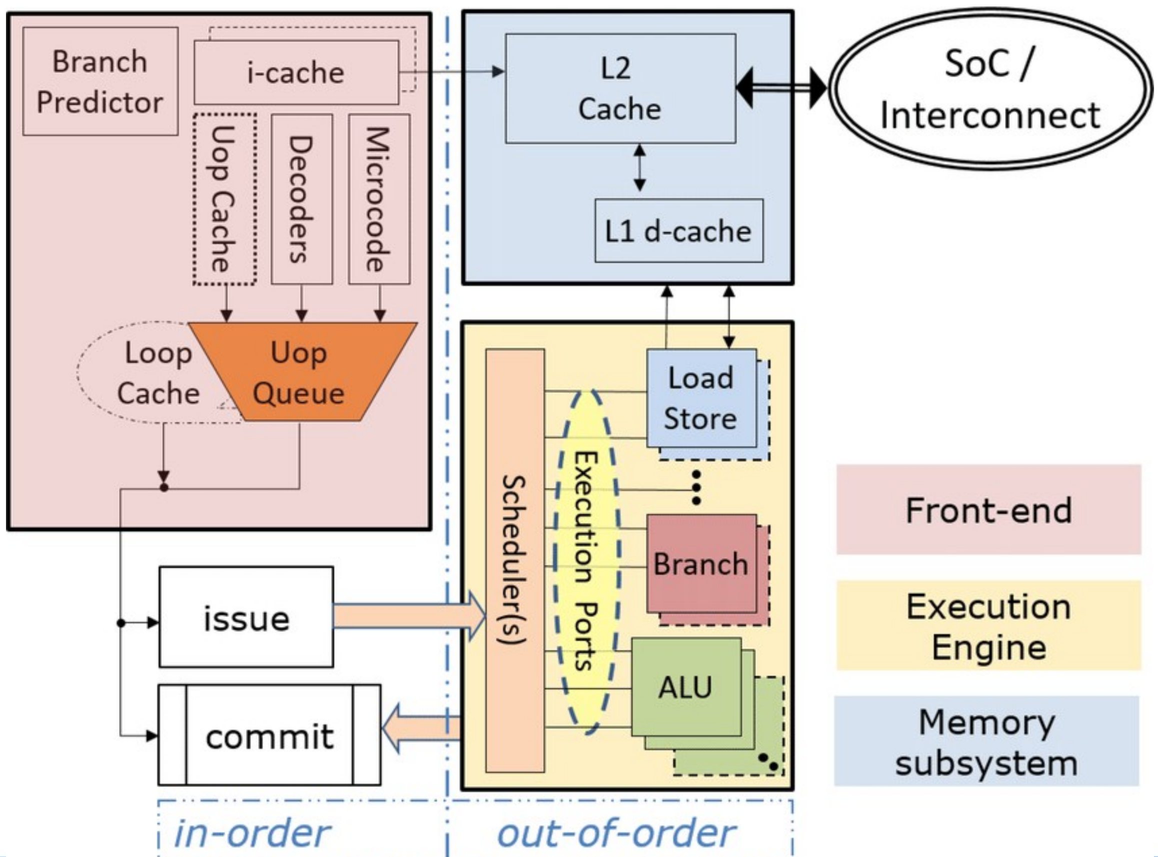
- The instruction set
- Memory addressing modes
- Registers
- Address and data formats



Computer architecture and computer organization

❖ **Micro-architecture**: the computer organization, describing the system at a low level, concerned with:

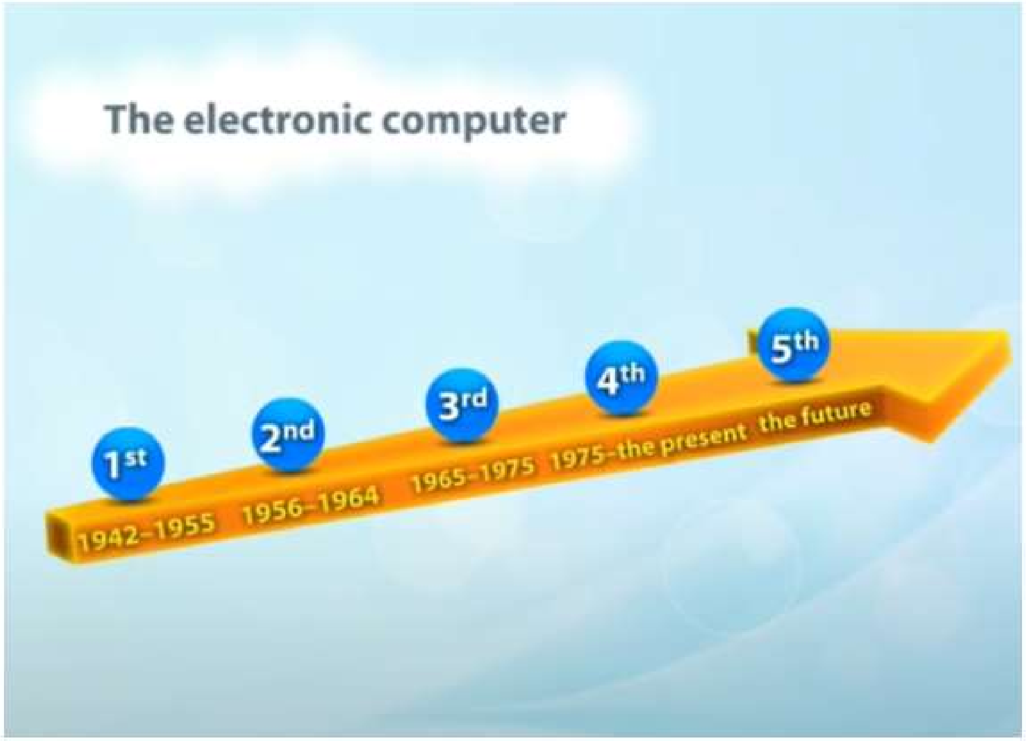
- How hardware components are interconnected
- How hardware components cooperate and interact to execute instructions



❖**System design**: includes all other hardware components in a computer system, such as:

- Interconnection systems like buses and switches
- Memory controllers and the memory hierarchy structure
- Techniques for offloading work from the CPU, such as direct memory access (DMA)
- Issues such as multiprocessing

The history of computer development is a continuous evolution of hardware technology and architecture, and it is divided into five main generations based on changes in electronic components.



Generations of Computer

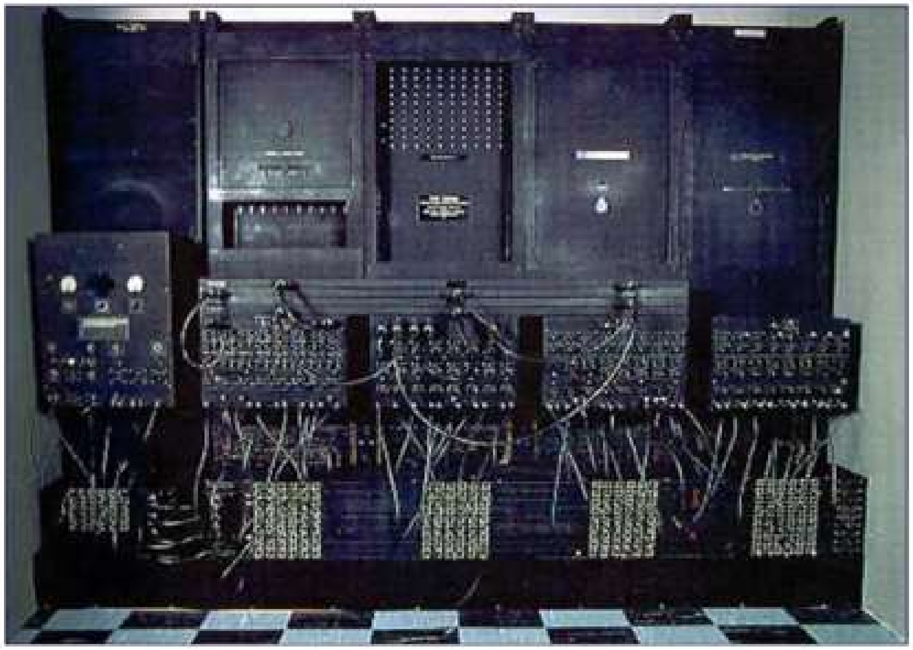
History of machine development

❖**First Generation (1944–1959):**

- Used vacuum tubes
- Used magnetic tape for input/output devices
- Component integration density: about 1,000 components per cubic foot (1 foot = 30.48 cm)

❑Example: **ENIAC** –
Electronic Numerical Integrator and Computer (1946),
costing about USD 500,000

ENIAC



History of machine development

❖ **Second Generation (1960–1964):**

- Used transistors
- Approximately 100,000 components per cubic foot

❑ Examples: UNIVAC 1107, UNIVAC III, IBM 7070, 7080, 7090, 1400 series, 1600 series (1951; the first cost USD 159K, later the UNIVAC I cost over USD 1 million)

UNIVAC

❖Third Generation (1964–1975):

- Used integrated circuits (ICs)
- Approximately 10 million components per cubic foot

❑Examples: UNIVAC 9000 series, IBM System/360, System/3, System/7

UNIVAC 9400

History of machine development

❖ Fourth Generation (1975–1989):

- Used LSI – Large-Scale Integrated Circuits
- Approximately 1 billion components per cubic foot

❑ Examples: IBM System/3090,
IBM RISC 6000, IBM RT, Cray 2 XMP

Cray 2 XMP

History of machine development

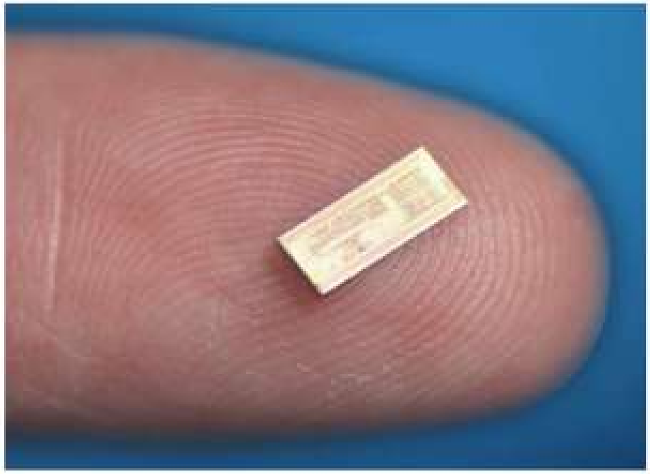
❖ **Fifth Generation (1990–present):**

- Uses VLSI – Very Large-Scale Integrated Circuits

❑ Examples: Pentium II, III, IV, M, D, Core Duo, Core 2 Duo, Core Quad, …

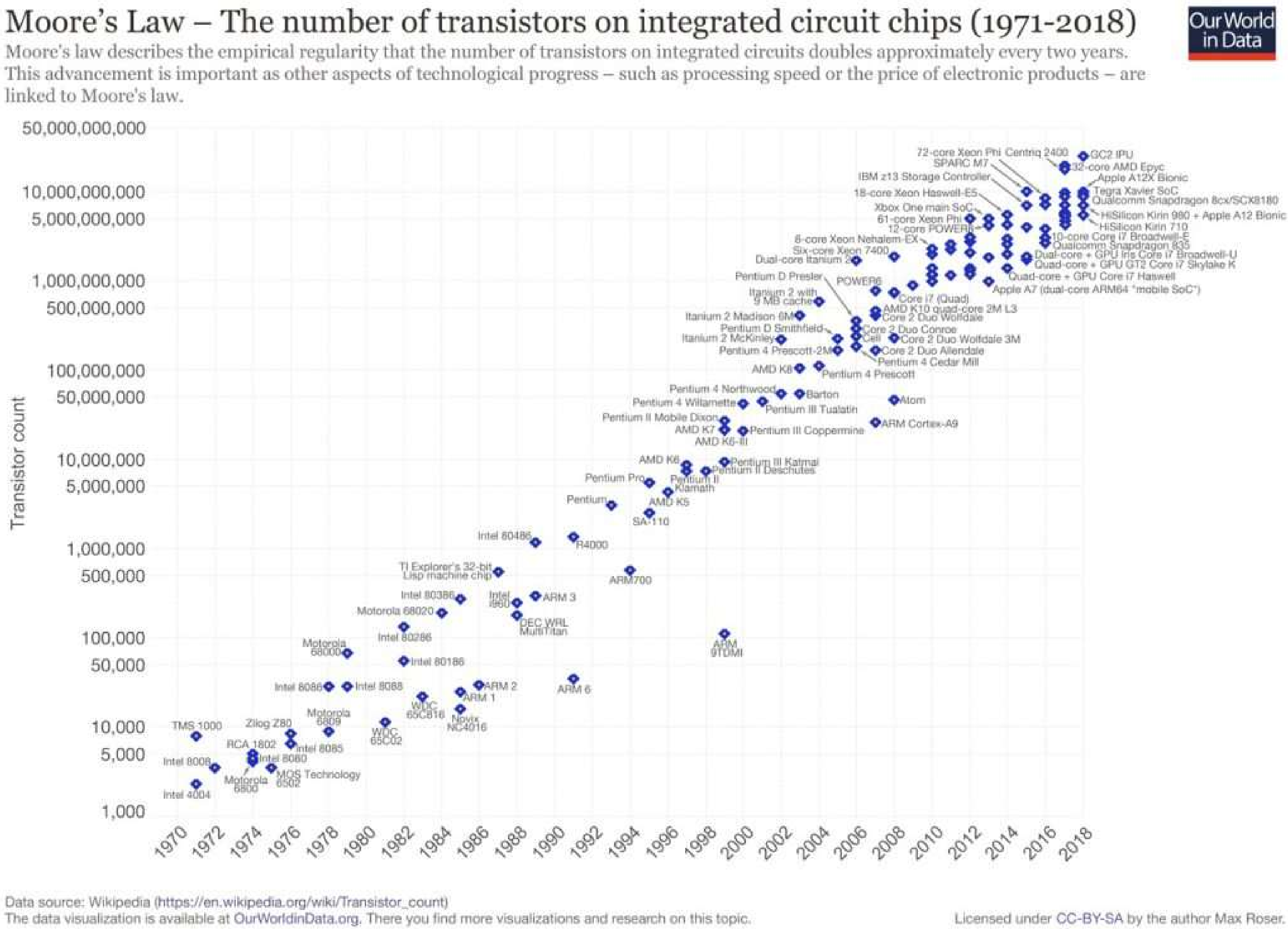
- Supports parallel processing
- Very high performance
- Integrates speech and image processing

Intel Core i9-13980HX



Intel Atom

History of machine development

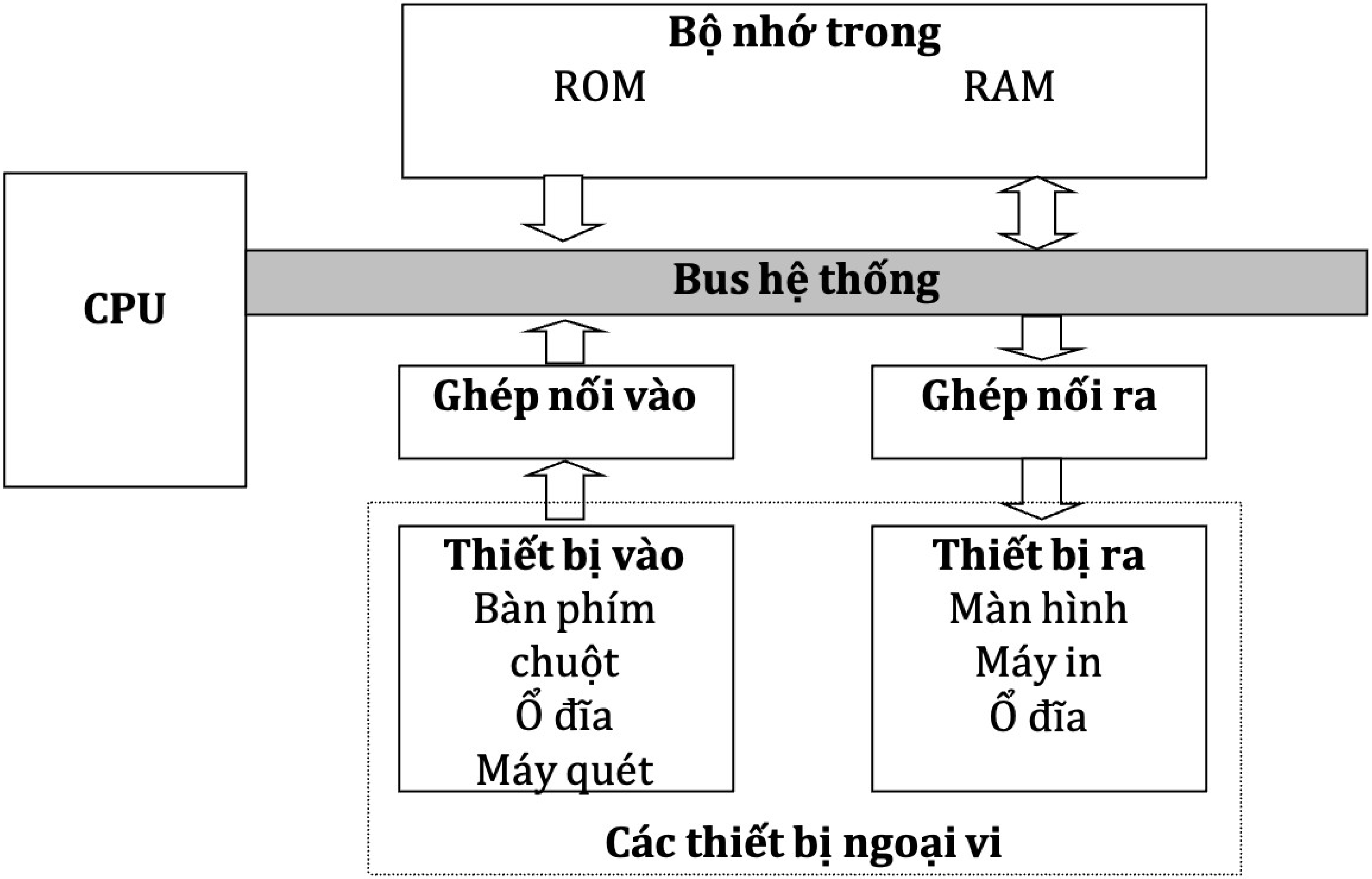


History of machine development

❖ **Moore's Law** (Observed by Gordon Moore, CEO of Intel, in 1965)

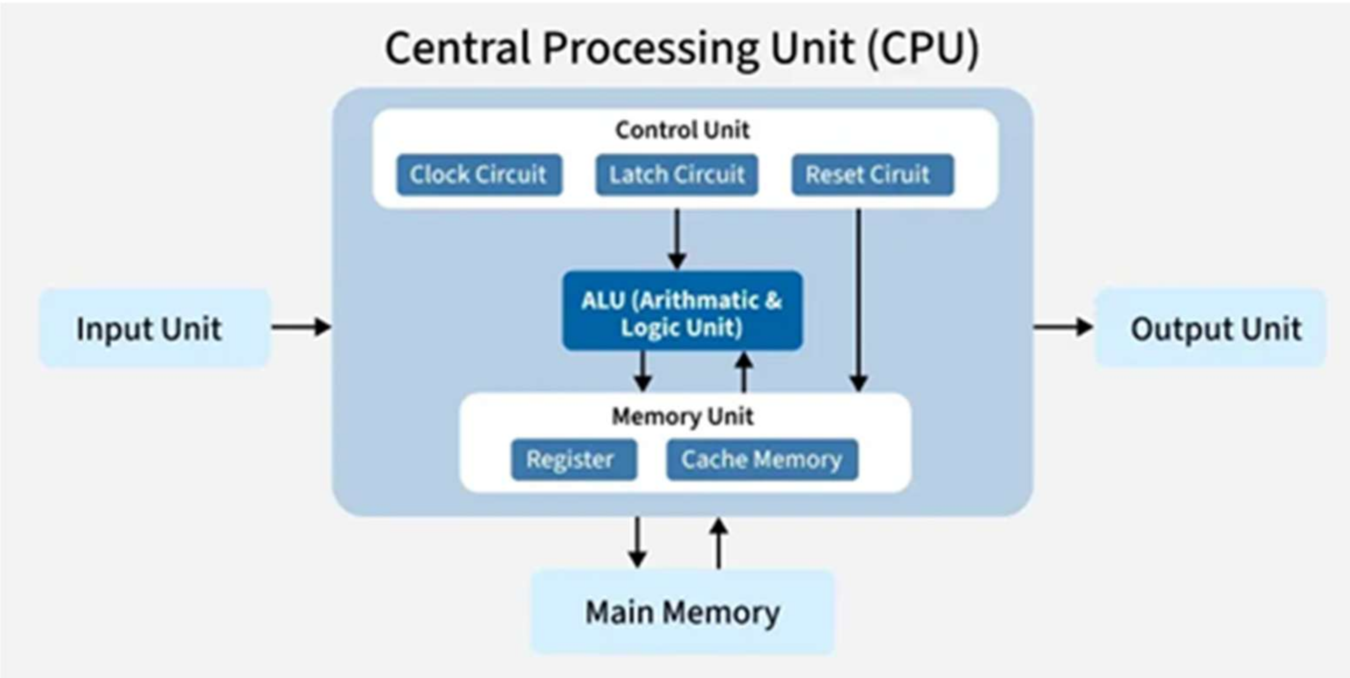
- The number of transistors on a chip doubles approximately every 18 months
- The cost of a chip remains nearly unchanged
- Higher density results in shorter interconnection paths
- Smaller IC sizes lead to increased complexity
- Lower power consumption
- More optimized designs

Structure and function of a computer



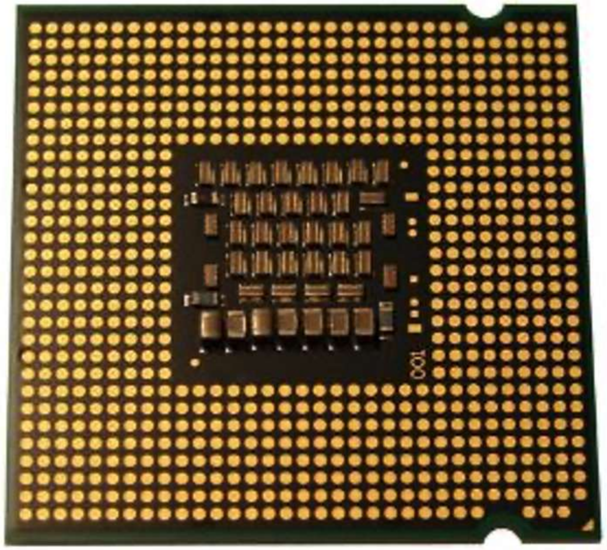
❖Central Processing Unit (CPU):

- **Functions:**
  - Fetch instructions from memory
  - Decode and execute instructions
- **Components:**
  - Control Unit (CU)
  - Arithmetic and Logic Unit (ALU)
  - Registers
  - Internal CPU buses



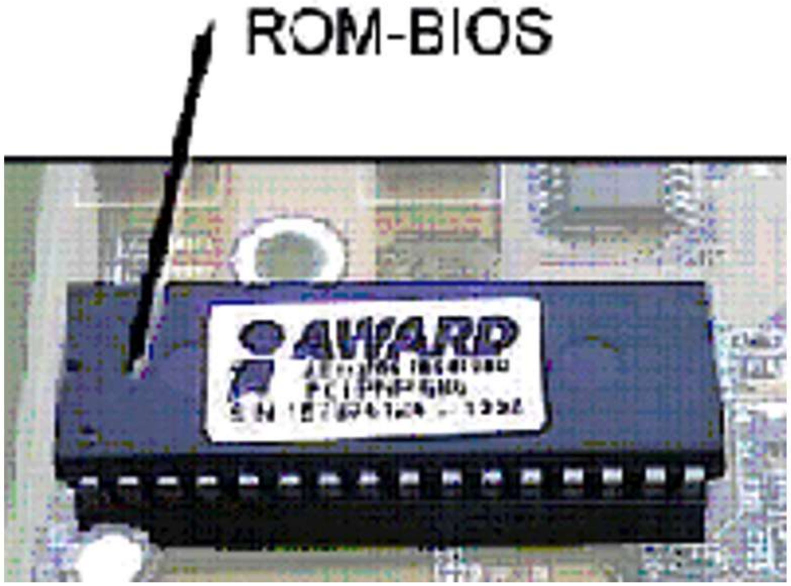
Vi xử lý Intel
8086 (1978)

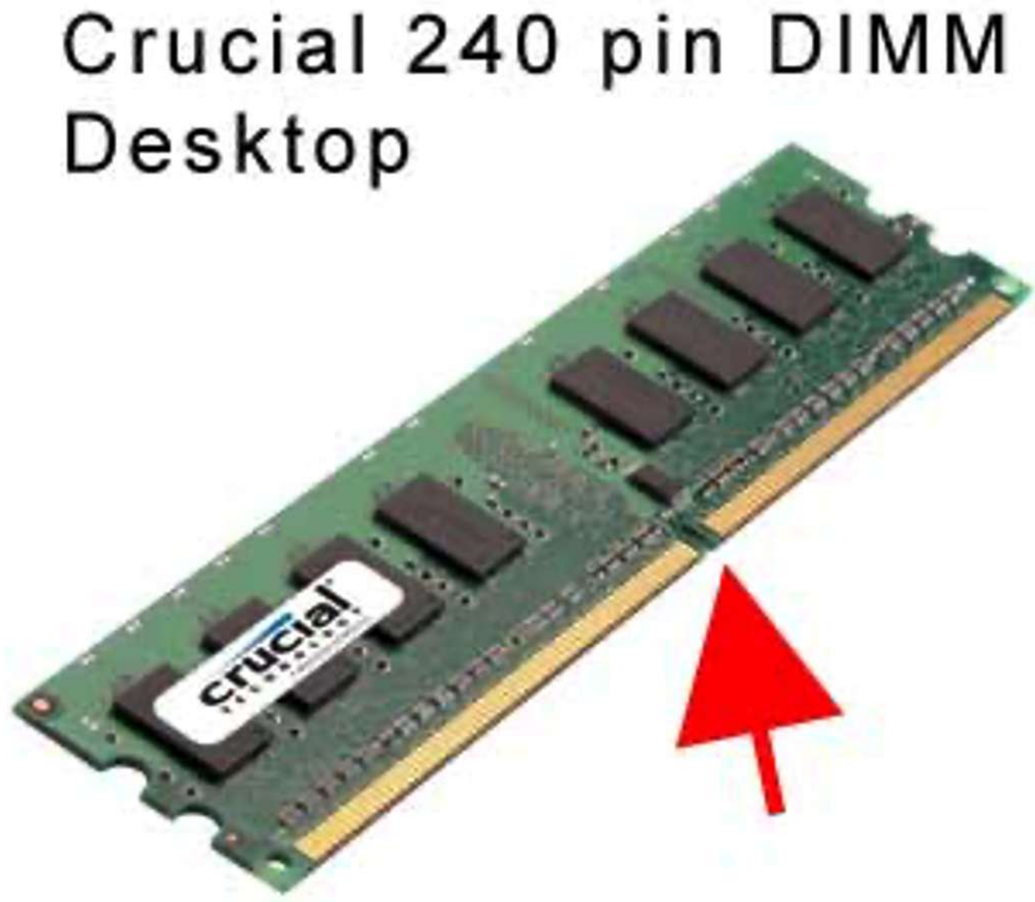
Vi xử lý Intel
Core 2 Duo
(2006)



❖Internal Memory:

- **Function:** Stores instructions and data for the CPU to process
- **Includes:**
  - **ROM – Read-Only Memory:**
    - –Stores system instructions and data
    - –Information in ROM is retained when power is lost
  - **RAM – Random Access Memory:**
    - –Stores system and user instructions and data
    - –Information in RAM is lost when power is removed


ROM-BIOS
AWARD


Crucial 240 pin DIMM
Desktop
crucial

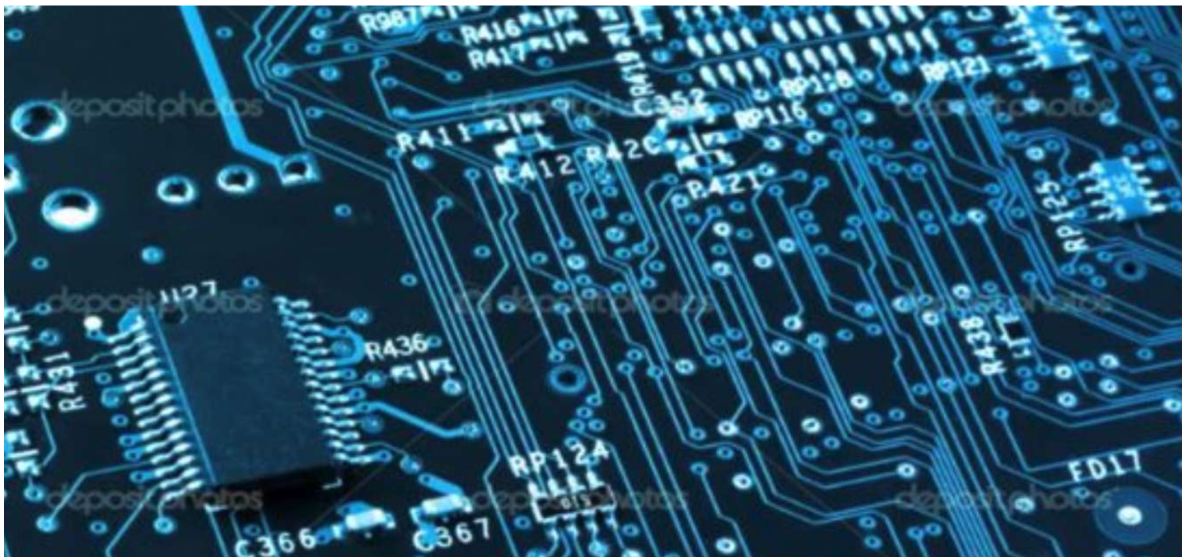
❖Input/Output Devices:

- **Input devices:** used for data entry and control
  - Keyboard
  - Mouse
  - Disk drive
  - Scanner
- **Output devices:** used for data output
  - Monitor
  - Printer
  - Disk drive

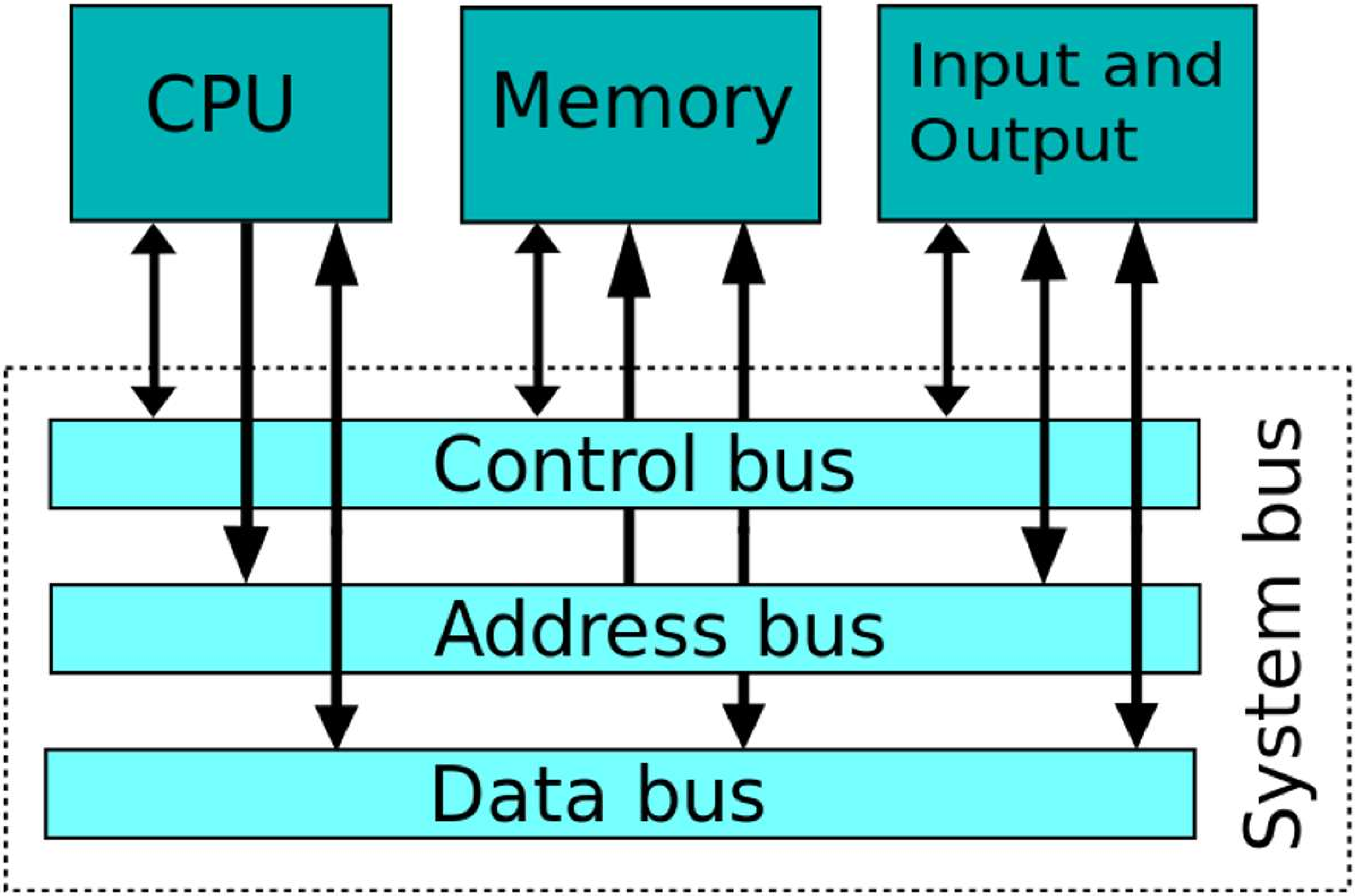
Structure and function of a computer

❖System Bus:

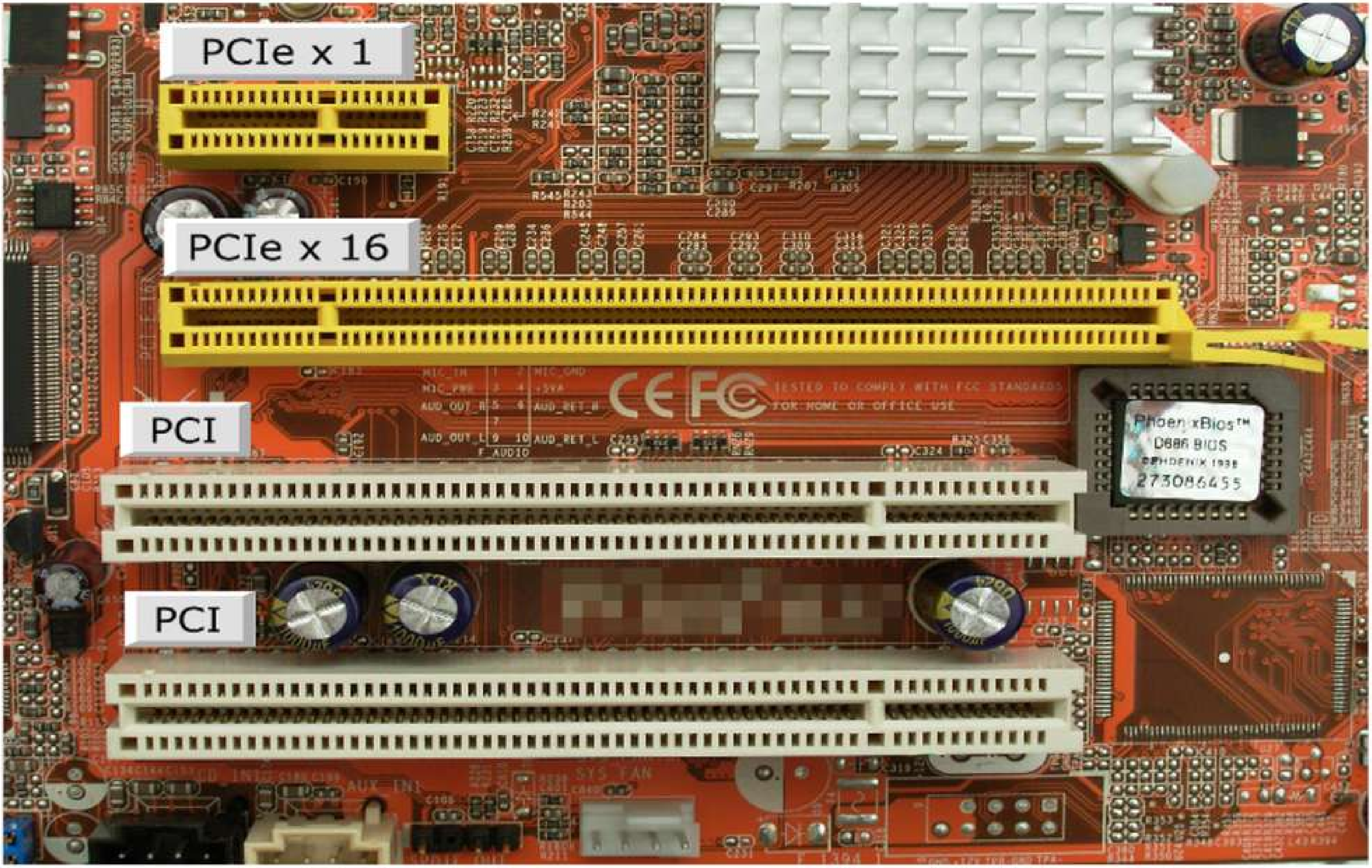
- A set of signal lines that connect the CPU with other components of the computer system
- Includes three types:
  - **Address bus** (bus A)
  - **Data bus** (bus D)
  - **Control bus** (bus C)



Structure and function of a computer



PCI bus:



Structure and function of a computer

❖ **Computer Function:** The basic function of a computer is to execute programs by processing instructions sequentially. This process takes place through a repetitive cycle consisting of the following main stages:

- **Instruction Fetch:** The CPU reads an instruction from main memory and loads it into the instruction register.
- **Instruction Decode:** The Control Unit (CU) analyzes the instruction code to determine the operation to be performed and the associated operands.
- **Instruction Execution:** The CPU performs the operation (such as computations in the ALU or data movement) and writes the result back to memory or to a register if required.

By executing millions of such instruction cycles per second, a computer is able to process complex user requests quickly and accurately.

Computer classification

Computer classification is quite diverse and can be considered from many different perspectives depending on scale, performance, or system architectural characteristics

❖ **Classification by scale and processing performance**

Based on computational power and application scope, computers are typically divided into three main groups:

- **Mainframe computers:** These are very large computers with multiple high-speed processors and extremely large memory capacities. They are commonly used to solve complex problems in fields such as the military, space research, or large-scale banking system management.
- **Minicomputers:** These are smaller versions of mainframe computers, designed to serve scientific and engineering computing needs or medium-scale data processing.
- **Microcomputers:** This is the most common type today, ranging from compact microcontrollers to personal computers with processing capabilities comparable to minicomputers. In addition, at the highest level, there are **supercomputers**, whose costs can reach millions of dollars.

Computer classification

Computer classification is quite diverse and can be considered from many different perspectives depending on scale, performance, or system architectural characteristics

❖ **Classification by system architecture**

Computer architecture can also be classified based on the historical evolution of hardware:

- **By memory organization:** This includes the **pre–von Neumann architecture** (computers without internal program storage, appearing before 1946) and the **von Neumann architecture** (programs and data stored together in main memory, widely used from after 1946 to the present).
- **By instruction set characteristics:** This includes **load–store architecture** (1963–1976), **CISC** (Complex Instruction Set Computer) architecture, and **RISC** (Reduced Instruction Set Computer) architecture.

Computer classification

❖Some common types of computers:

- **Personal Computers (PCs)**
  - Desktop computers, laptop computers
  - General-purpose computers
- **Servers**
  - Used in networks to manage and provide services
  - High performance and high reliability
  - Cost ranges from thousands to millions of USD
- **Supercomputers**
  - Used for high-end computations in science and engineering
  - Cost ranges from millions to hundreds of millions of USD

Computer classification

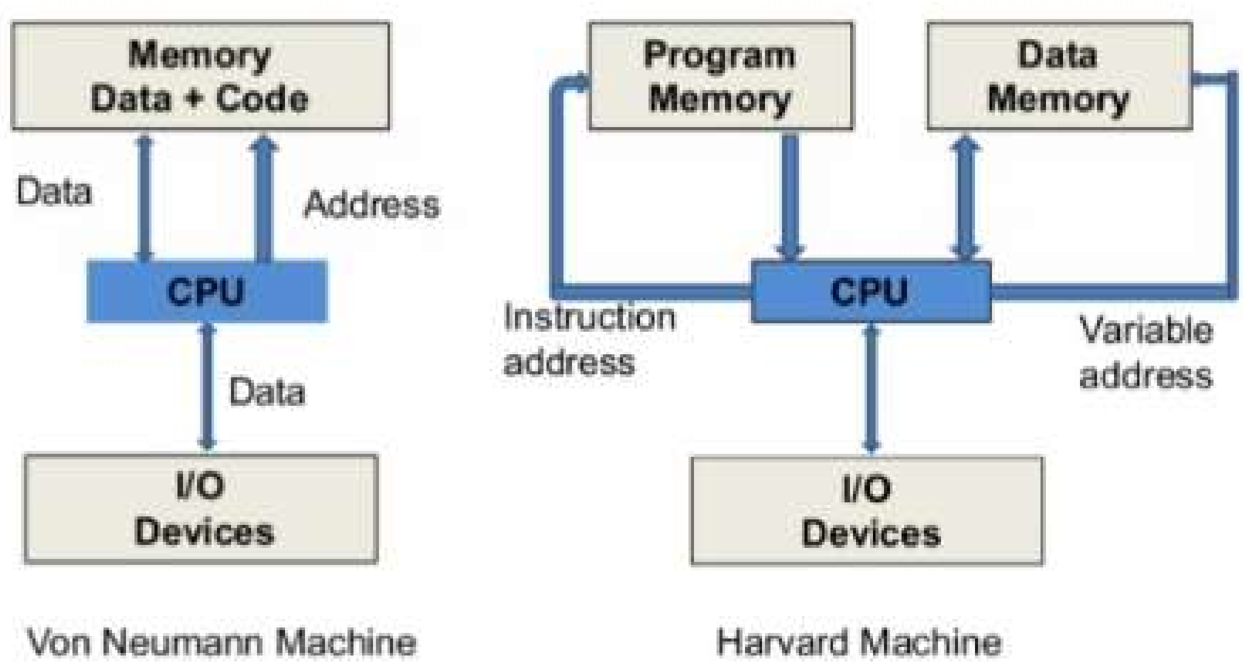
❖ **Some common types of computers:**

- **Embedded Computers**
  - Hidden inside other devices
  - Designed for specialized purposes
- **Personal Mobile Devices (PMDs)**
  - Smartphones, tablets
  - Internet connectivity
- **Cloud Computing**
  - Uses large-scale computing systems (warehouse-scale computers) consisting of many interconnected servers
  - Companies rent a portion of these resources to provide software services
  - **Software as a Service (SaaS):** part of the software runs on PMDs, while another part runs in the cloud
  - Examples: Amazon, Google

Computer architecture design

❖Many different architectures have been developed (more than 20 different computer architectures).
In this course, we focus on the following two main architectures:

- **von Neumann architecture**
- **Harvard architecture**

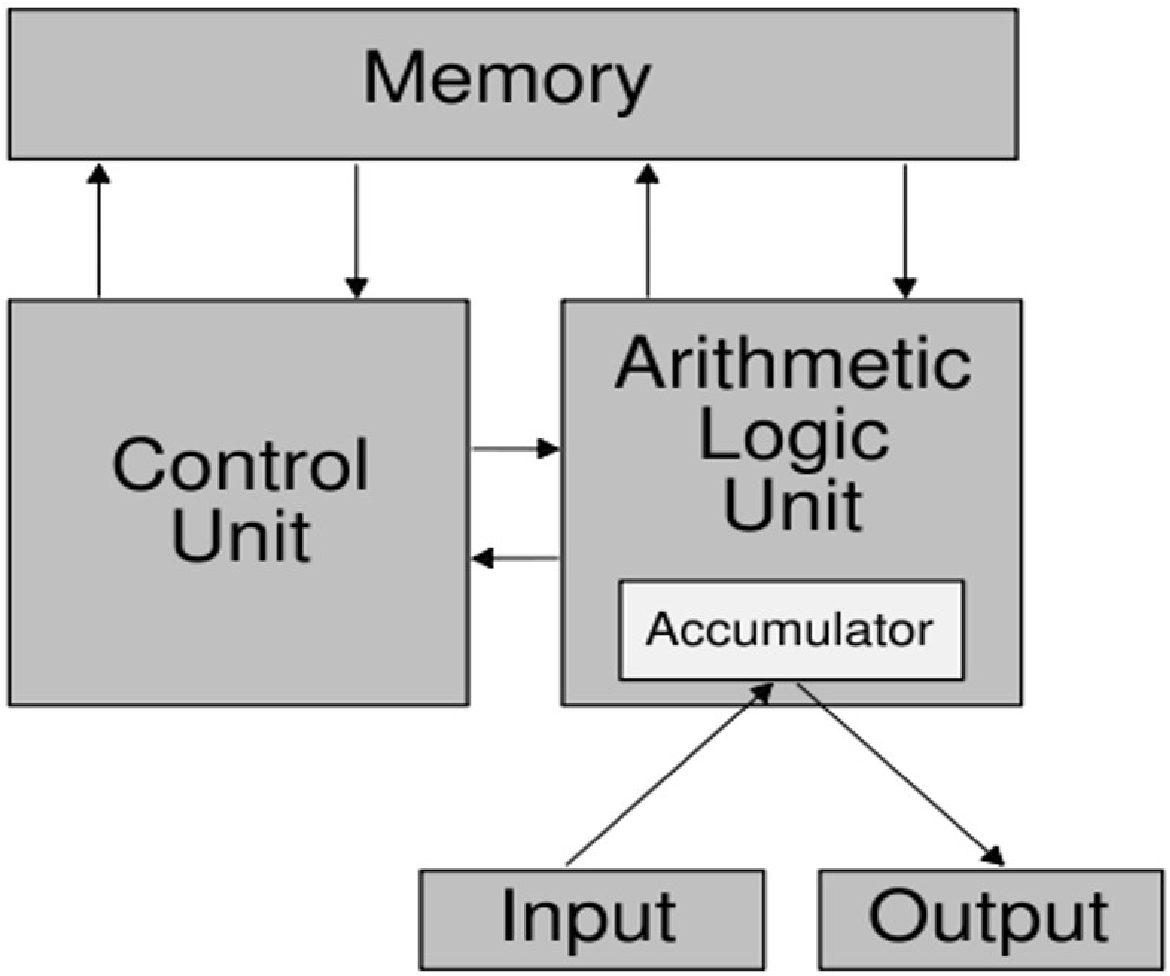


Computer architecture design

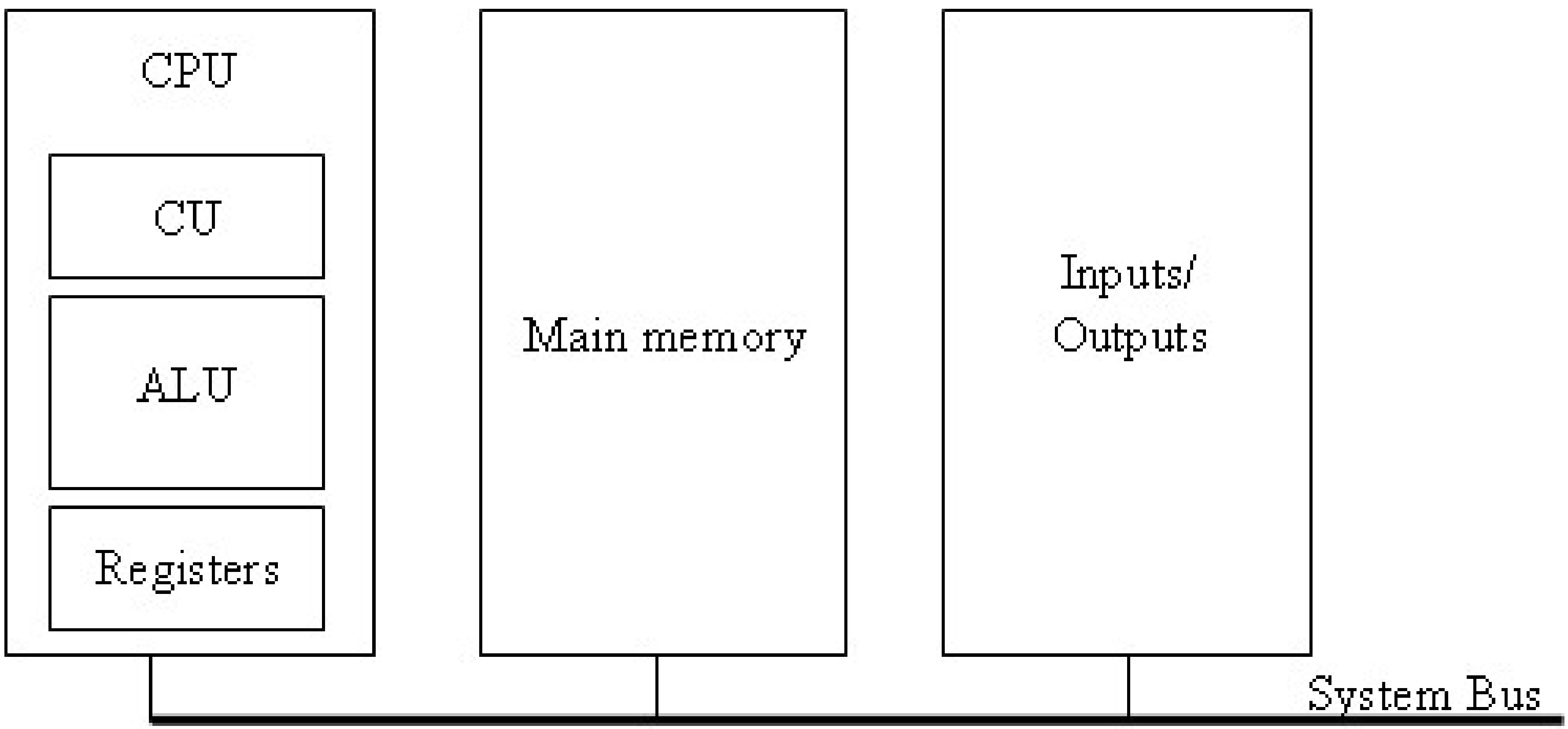
❖ **The von Neumann architecture** was introduced by John von Neumann in 1945. Computers based on the von Neumann architecture rely on three fundamental concepts:

- Data and instructions are stored together in a shared read/write memory
- Memory is addressed by location (address) and is independent of the type of content stored
- Program instructions are executed sequentially, one after another: the **stored-program digital computer** concept

Von-Neumann architecture (old version)



Von-Neumann architecture (modern version)



Computer architecture design

❖ **The instruction execution process** is divided into three main stages:

- The CPU fetches the instruction from memory
- The CPU decodes and executes the instruction; if the instruction requires data, the data are read from memory
- The CPU writes the result back to memory if necessary

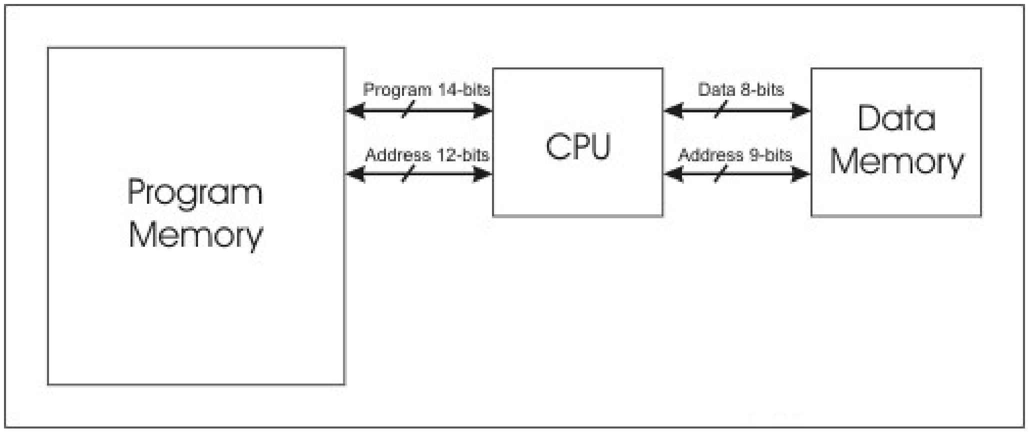
❖ **Limitation:** Instruction memory and data memory (the bottleneck) cannot be accessed simultaneously, so the throughput is much lower than the speed at which the CPU can operate

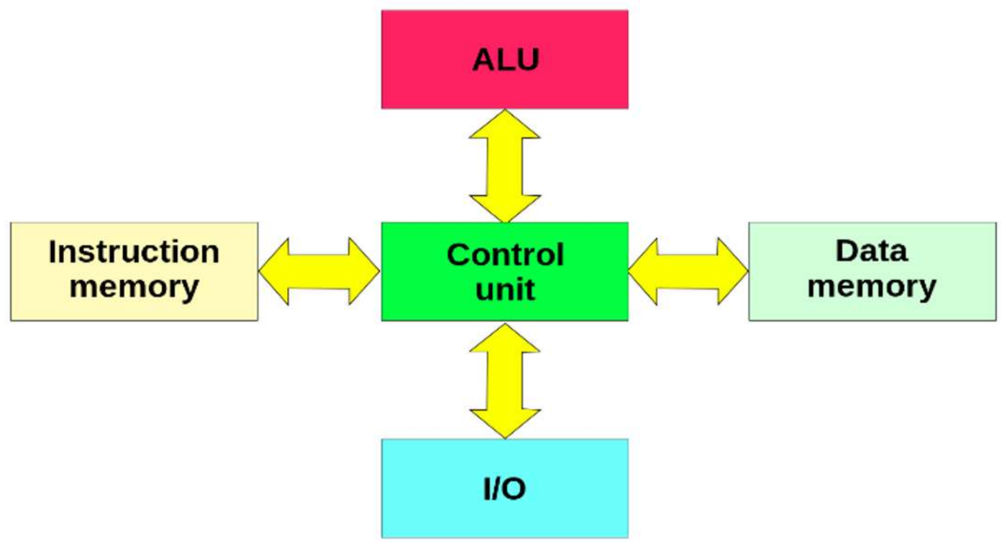
❖ **Mitigation:** Use cache memory between the CPU and main memory

❖The Havard architecture

- **Overcomes the shortcomings of the von Neumann architecture**
- Memory is divided into two parts:
  - Program memory
  - Data memory
- The CPU uses two system buses to interface with memory:
  - The CPU can fetch instructions and access data memory simultaneously
  - One address/data bus pair (A, D) for program memory and one address/data bus pair (A, D) for data memory (with different formats)

a


Program Memory
Program 14-bits
Address 12-bits
CPU
Data 8-bits
Address 9-bits
Data Memory


ALU
Instruction memory
Control unit
Data memory
I/O

❖The Havard architecture

- **Faster performance** due to wider bus bandwidth, since instruction fetching does not contend with data access
- **Supports multiple simultaneous memory read/write accesses**, thereby reducing memory access conflicts
- Today, **modified Harvard architecture** is applied in modern computer architectures such as ARM and Intel x86
- The **Harvard architecture** is also used in embedded systems and specialized signal-processing chips (DSPs)

Computer Organization

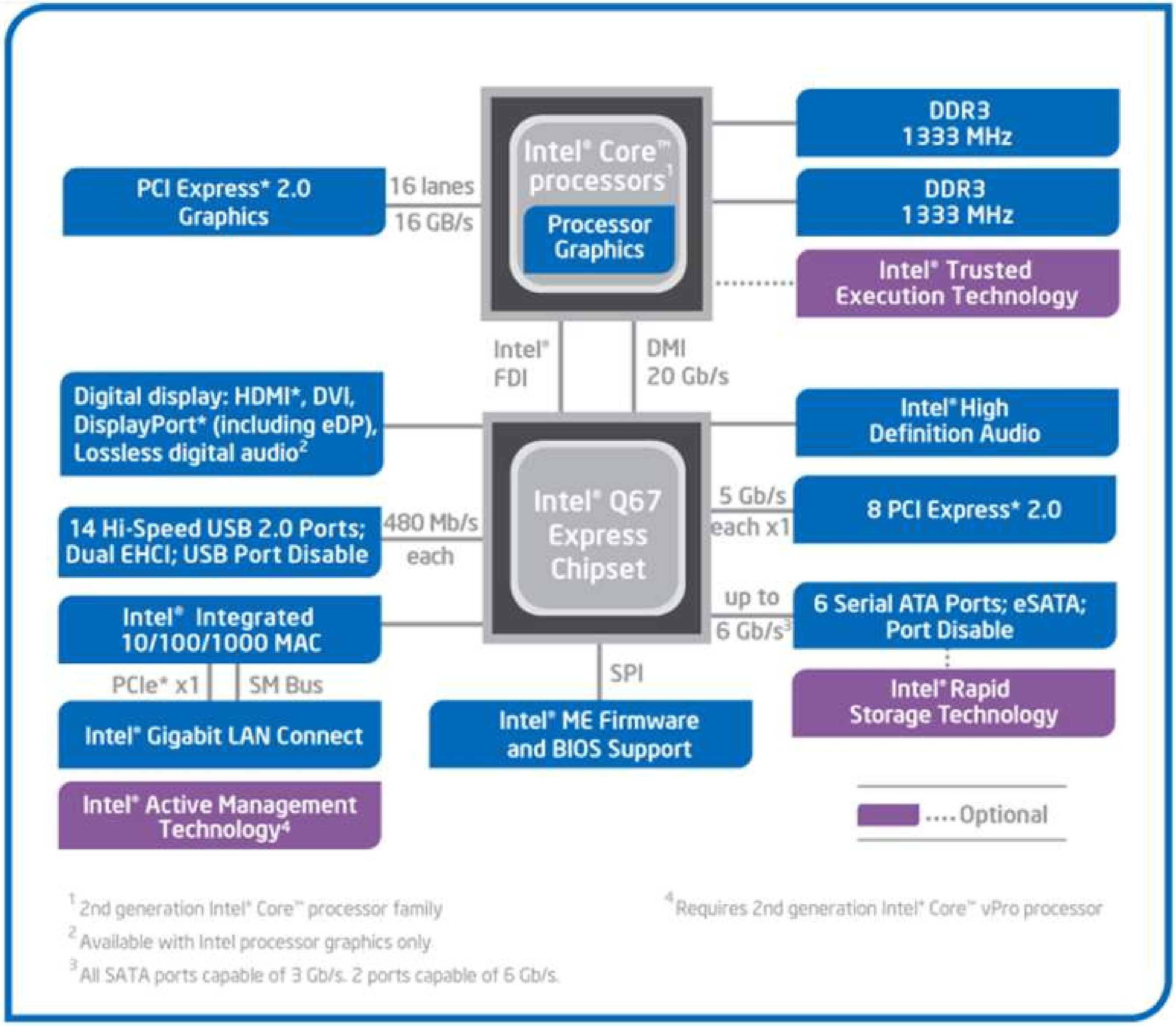
- **Computer Organization** is the field of study that examines the specific components that make up a computer and the ways in which they operate and are interconnected
  - **Object of study:** Focuses on operational units and their interactions, such as control signals, interfaces to peripheral devices, and the specific memory technologies employed.
  - **Relationship to microarchitecture:** Computer organization is conceptually similar to and closely related to **microarchitecture**, which is one of the three main components of computer architecture.

Computer Organization

❖ **Computer Organization** is the field of study that examines the specific components that make up a computer and the ways in which they operate and are interconnected

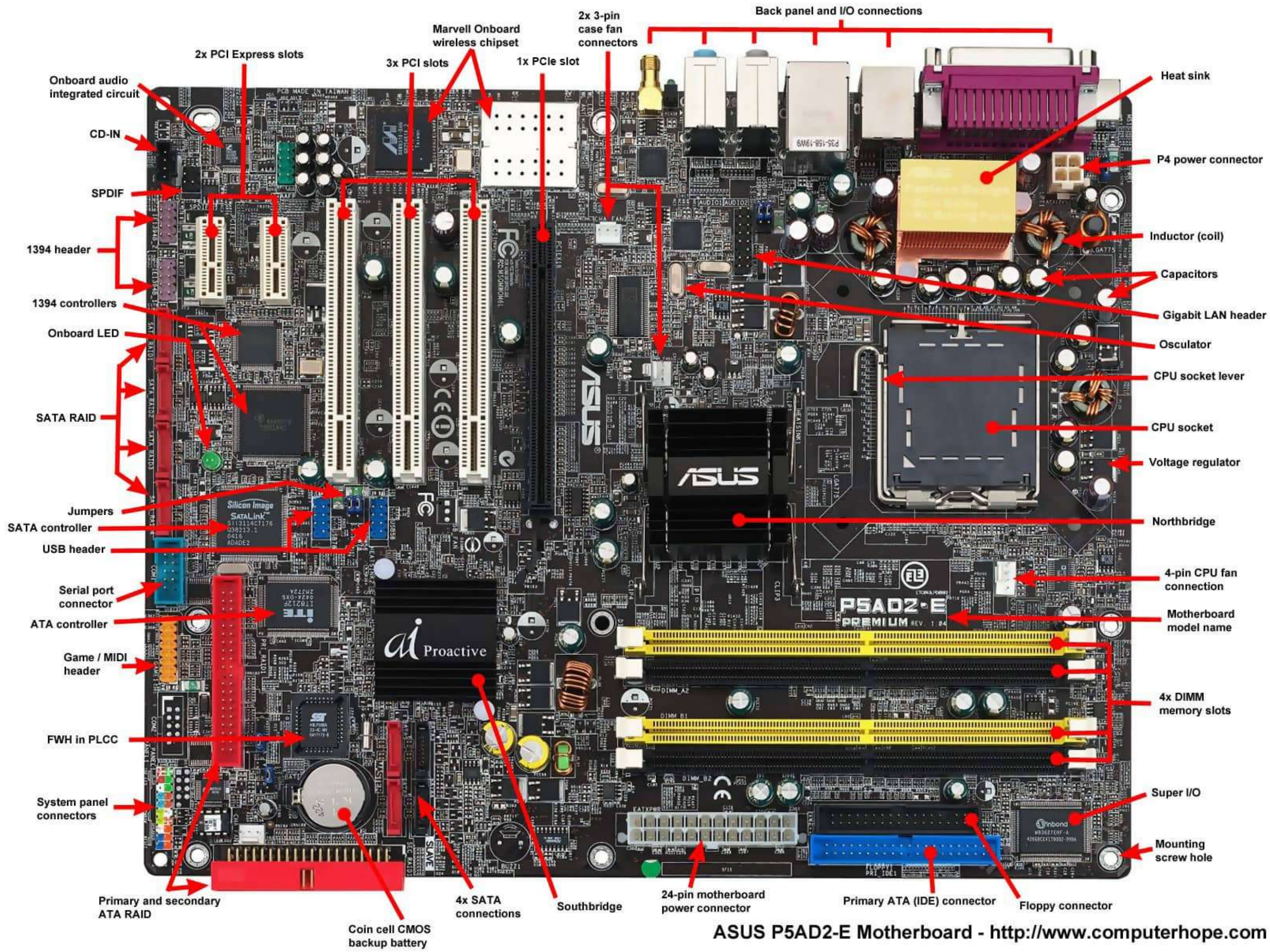
- **Implementation approach:** A key aspect of computer organization is determining how architectural functions are implemented. For example, whether a system supports a multiplication operation is an architectural issue, but whether that operation is carried out using a dedicated hardware multiplier or through repeated addition is a matter of computer organization.
- **Technology dependence:** Unlike architecture, which can remain stable for many years across a family of computers, computer organization tends to change continuously in response to advances in microprocessor technology, as well as considerations of cost and the physical size of components.

Computer Organization

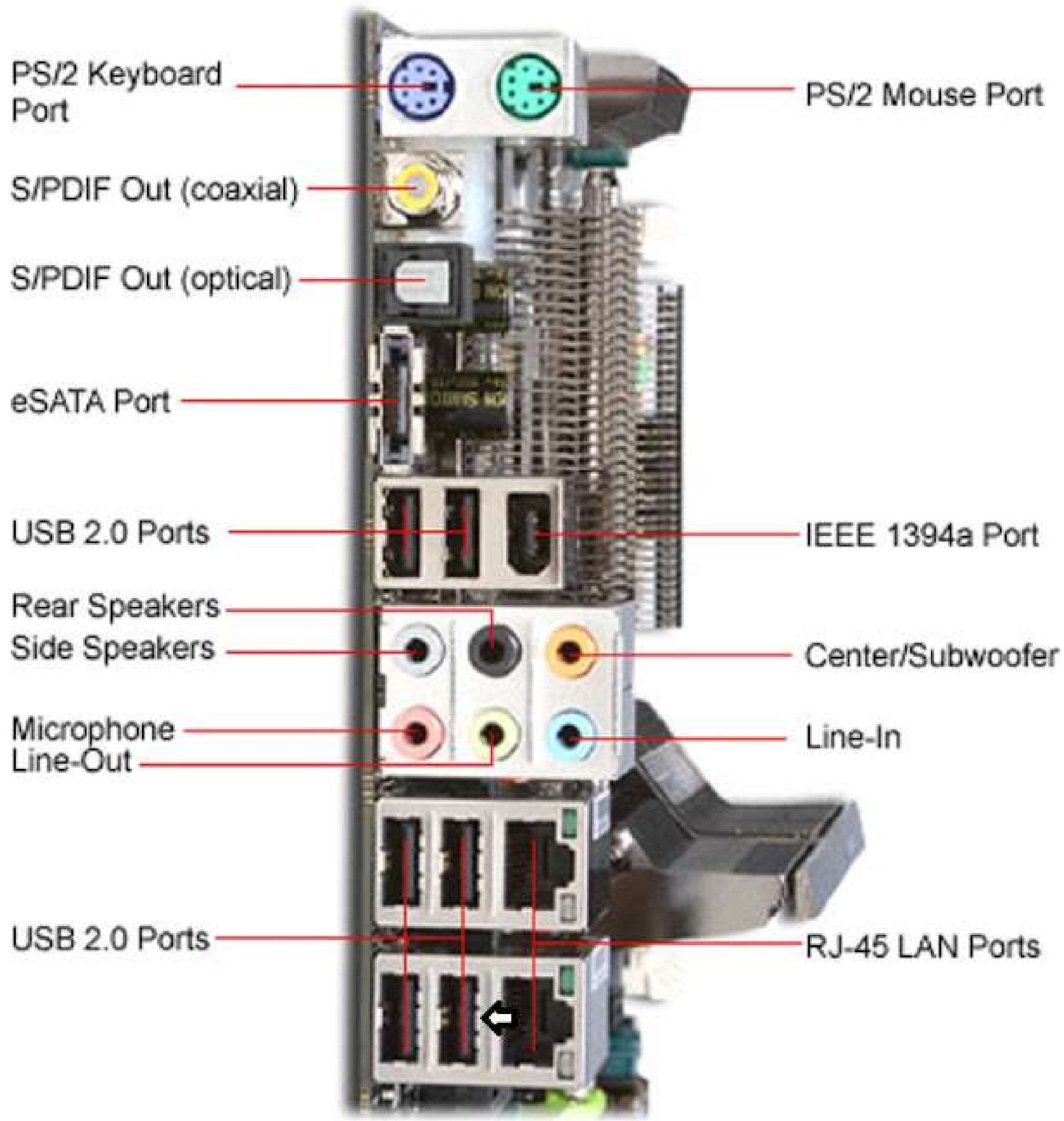


Intel® Q67 Express Chipset Platform Block Diagram

Computer Organization



ASUS P5AD2-E Motherboard - http://www.computerhope.com


PS/2 Keyboard Port
S/PDIF Out (coaxial)
S/PDIF Out (optical)
eSATA Port
USB 2.0 Ports
Rear Speakers
Side Speakers
Microphone
Line-Out
USB 2.0 Ports
PS/2 Mouse Port
IEEE 1394a Port
Center/Subwoofer
Line-In
RJ-45 LAN Ports

Numeral systems

❖ In the field of computer architecture, number systems play an important role in representing and processing data. The basic number systems include:

- **Decimal System**
  - The number system used by humans
  - Uses 10 symbols: 0, 1, 2, …, 9
- **Binary System**
  - The number system used by computers
  - Uses 2 symbols: 0 and 1
- **Hexadecimal System**
  - Used as a compact representation of binary numbers
  - Uses 16 symbols: 0, 1, 2, …, 9, A, B, C, D, E, F

Numeral systems

❖ **Decimal system:** Using n decimal digits, it is possible to represent 10^n different values.

$$00..000 = 0$$

$$99..999 = 10^n - 1$$

- A decimal number can be expressed in polynomial form as follows:

$$a_n a_{n-1}..a_0, a_{-1}..a_{-m} = \Sigma_{i=-m}^{n} a_i * 10^i$$

$$= a_n*10^n + a_{n-1}*10^{n-1} + .. + a_0*10^0 + a_{-1}*10^{-1} + .. + a_{-m}*10^{-m}$$

- **Example:** 123.456

$$123.456 = 1 * 10^2 + 2 * 10^1 + 3 * 10^0 + 4 * 10^{-1} + 5 * 10^{-2} + 6 * 10^{-3}$$

$$= 100 + 20 + 3 + 0.4 + 0.05 + 0.006$$

Numeral systems

❖ **Binary system:** Uses two binary digits, 0 and 1 (*binary digits*), called **bits**.

- Using n bits, it is possible to represent 2^n different values.

$$00..000 = 0$$

$$11..111 = 2^n - 1$$

- A binary number can be expressed in polynomial form as follows:

$$(a_n a_{n-1}..a_0, a_{-1}..a_{-m})_{(2)} = \Sigma_{i=-m}^{n} a_i * 2^i$$

$$= a_n * 2^n + a_{n-1} * 2^{n-1} + .. + a_0 * 2^0 + a_{-1} * 2^{-1} + .. + a_{-m} * 2^{-m}$$

- **Example:** 1101001.1011_2

$$1101001.1011_{(2)} = 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3} + 1 * 2^{-4}$$

$$= 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625$$

$$= 105.6875_{(10)}$$

Numeral systems

❖ **Converting a decimal number to binary:**

▶ **Integer part:** repeatedly divide by 2 and take the remainders

▶ **Fractional part:** multiply by 2 and take the integer part

$105,6875_{(10)}$

105 : 2 = 52	dư	1
52 : 2 = 26	dư	0
26 : 2 = 13	dư	0
13 : 2 = 6	dư	1
6 : 2 = 3	dư	0
3 : 2 = 1	dư	1
1 : 2 = 0	dư	1

0.6875 x 2	= 1.375	phần nguyên =	1
0.375 x 2	= 0.75	phần nguyên =	0
0.75 x 2	= 1.5	phần nguyên =	1
0.5 x 2	= 1.0	phần nguyên =	1

Kết quả:

$105_{(10)} = 1101001,1011_{(2)}$

❖ Converting a decimal number to binary:

▶ Decompose the number into a sum of powers of 2^i ,which is faster for small numbers

$$105{,}6875 = 64+32+8+1+0.5+0.125+0.0625$$
$$= 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
128	64	32	16	8	4	2	1	0.5	0.25	0.125	0.0625
0	1	1	0	1	0	0	1	1	0	1	1

Kết quả: $105{,}6875_{(10)} = 0110\ 1001{,}1011_{(2)}$

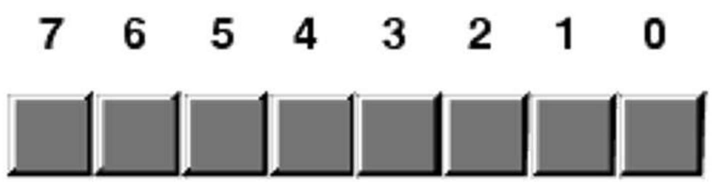
❖Hexadecimal system (Hex):

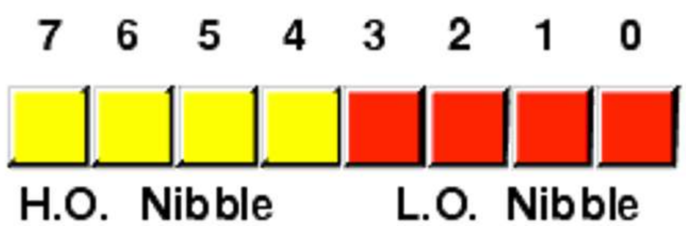
- Uses 16 symbols: 0, 1, 2, 3, …, 8, 9, A, B, C, D, E, F
- Used as a compact representation of binary numbers: each group of 4 bits is replaced by one hexadecimal digit

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

❖ Data Organization

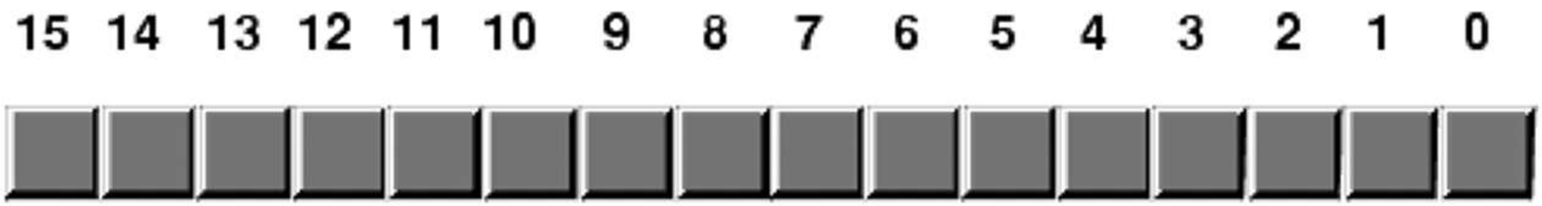
- **Bit:**
  - A bit is the smallest unit of data
  - A bit can store only two values: 0 or 1, true or false
- **Nibble:**
  - A group of 4 bits
  - Can store 16 values, from $(0000)_2$ to $(1111)_2$, or one hexadecimal digit
- **Byte:**
  - A group of 8 bits, or 2 nibbles
  - Can store 256 values, from $(00000000)_2$ to $(11111111)_2$, or two hexadecimal digits, from 00_{16} to FF_{16}

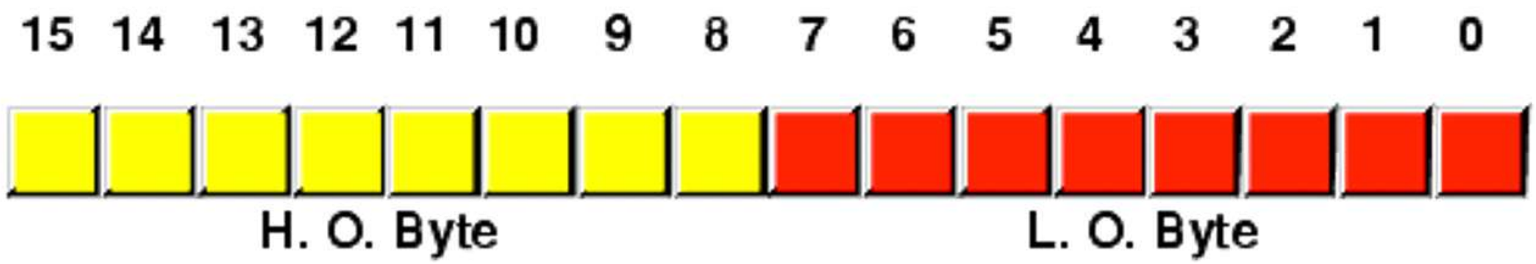




❖ Data Organization

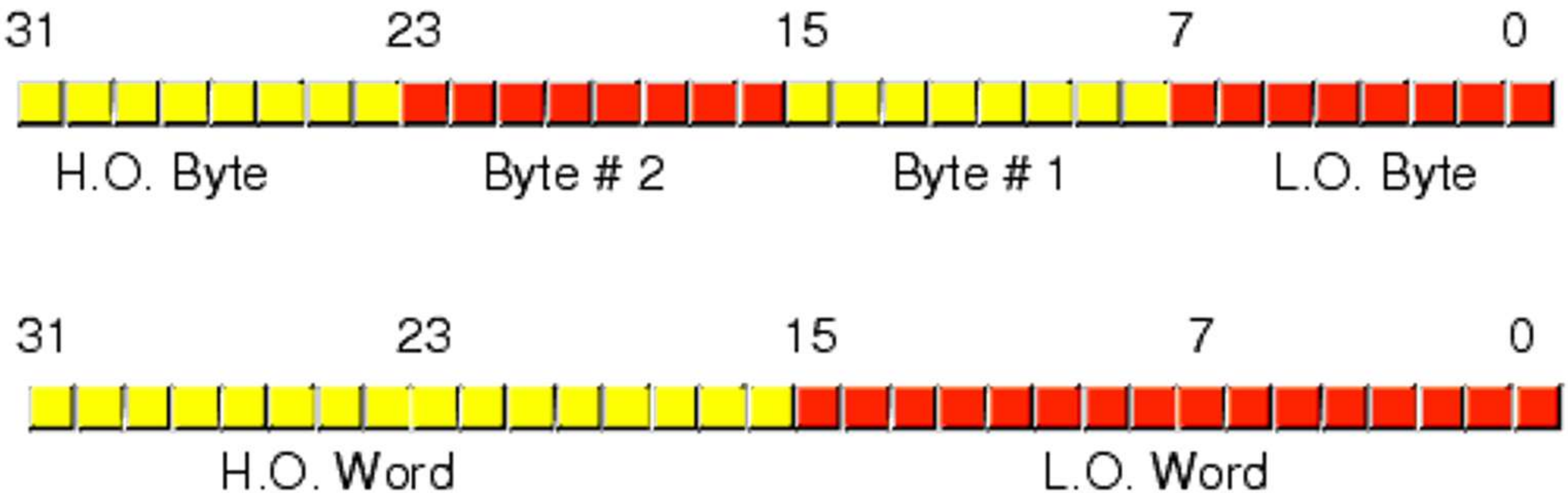
- **Word:** A word is a group of bits that represents a certain type of data.
  - Its size typically ranges from 4 to 64 bits.
  - Common word sizes are 8, 16, 24, 32, or 64 bits, depending on the microprocessor.
  - Example: In a 16-bit microprocessor → **word = 16 bits = 2 bytes**.





❖ **Data Organization**

- **Double word:** A group of bits consisting of two words.
  - Example: In a 16-bit microprocessor → **double word = 2 words = 32 bits = 4 bytes**.



❖Signed and Unsigned Numbers

- **Signed numbers in the binary system:** the leftmost bit is used to represent the sign of the number (**sign–magnitude**):
  - Leftmost bit = 0 → positive number
  - Leftmost bit = 1 → negative number
- **Example:** using 4 bits to represent numbers in sign–magnitude form
  - 0011, 0111, 0101 represent the positive numbers +3, +7, +5
  - 1011, 1111, 1101 represent the negative numbers −3, −7, −5
- **Unsigned numbers:** all bits are used to represent the value.
- One common method for representing signed numbers is **two's complement (2's complement)**.

❖ Signed and Unsigned Numbers

- **Representation range:** *n* bits can represent up to 2^n values:
  - **Unsigned numbers:** from 0 to $2^n - 1$
    8 bits: 0 to 255
    16 bits: 0 to 65,535
    32 bits: 0 to 4,294,967,295
  - **Signed numbers – sign–magnitude representation:** represent values from $-2^{n-1} + 1$ to $+2^{n-1} - 1$
    8 bits: −127 to +127
    16 bits: −32,767 to +32,767
    32 bits: −2,147,483,647 to +2,147,483,647
  - **Signed numbers – two's complement representation:** represent values from -2^{n-1} to $+2^{n-1} - 1$
    8 bits: −128 to +127
    16 bits: −32,768 to +32,767
    32 bits: −2,147,483,648 to +2,147,483,647

❖ASCII Code Table

- **ASCII (American Standard Code for Information Interchange)** is a standard character encoding scheme for the English alphabet used for data exchange in computing systems.
- Uses **8 bits** to represent one character.
- The ASCII code defines **128 characters**, including **33 control characters** and **94 printable characters**.
- The remaining values (129–255) are reserved.

❖ASCII Code Table

Binary	Oct	Dec	Hex	Abbr	PR[t 1]	CS[t 2]	CEC[t 3]	Description
000 0000	000	0	00	NUL	NUL	^@	\0	Null character
000 0001	001	1	01	SOH	SOH	^A		Start of Header
000 0010	002	2	02	STX	STX	^B		Start of Text
000 0011	003	3	03	ETX	ETX	^C		End of Text
000 0100	004	4	04	EOT	EOT	^D		End of Transmission
000 0101	005	5	05	ENQ	ENQ	^E		Enquiry
000 0110	006	6	06	ACK	ACK	^F		Acknowledgment
000 0111	007	7	07	BEL	BEL	^G	\a	Bell
000 1000	010	8	08	BS	BS	^H	\b	Backspace[t 4][t 5]
000 1001	011	9	09	HT	HT	^I	\t	Horizontal Tab
000 1010	012	10	0A	LF	LF	^J	\n	Line feed

❖ASCII Code Table

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	SP
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l