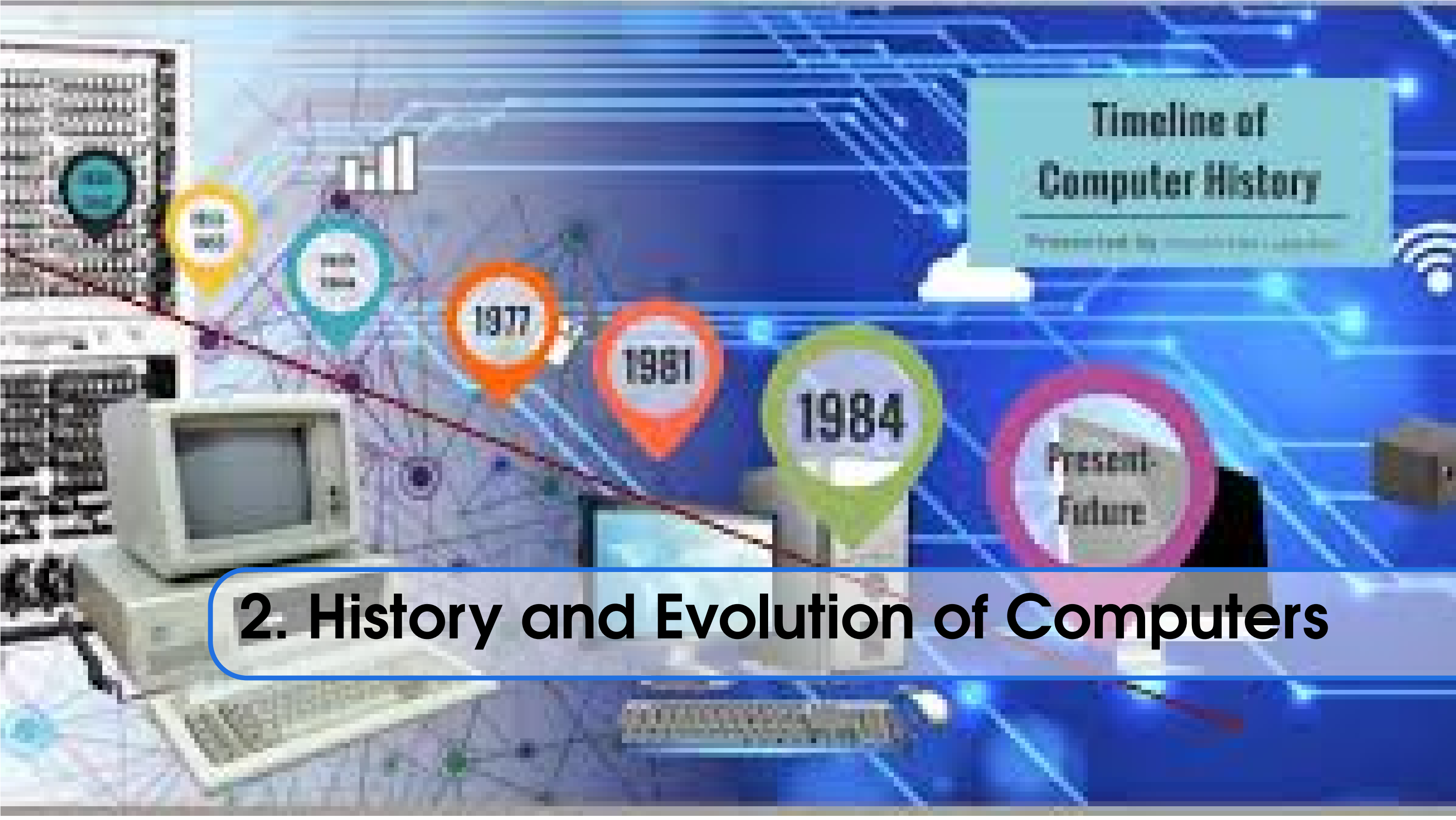

2. History and Evolution of Computers

2.1 History of Computers

The history of computers is a fascinating journey that spans centuries. Here's a brief overview:

- ✓ **Ancient Calculating Tools:** The earliest devices for computation include the abacus, which dates back thousands of years and was used for basic arithmetic;
- ✓ **Mechanical Calculators (17th Century):** Inventors like Blaise Pascal and Gottfried Wilhelm Leibniz developed mechanical calculators in the 17th century. These machines could perform addition and subtraction;
- ✓ **Charles Babbage and Analytical Engine (19th Century):** Often considered the "father of the computer," Charles Babbage conceptualized the Analytical Engine in the 1830s. Although it was never built during his lifetime, his designs laid the groundwork for modern computing concepts;
- ✓ **Ada Lovelace (19th Century):** Ada Lovelace, an associate of Babbage, is credited with writing the first algorithm for the Analytical Engine. She is often regarded as the world's first computer programmer;
- ✓ **Mechanical and Electromechanical Computers (Early to Mid-20th Century):** Devices like the Mark I, developed by Howard Aiken and Grace Hopper, were electromechanical computers that utilized punched cards for input;
- ✓ **ENIAC (1940s):** The Electronic Numerical Integrator and Computer (ENIAC) is considered the first general-purpose electronic digital computer. It was massive and used vacuum tubes for processing;

- ✓ **Transistors and Integrated Circuits (1950s-1960s):** The invention of transistors and later integrated circuits revolutionized computing, leading to smaller, more powerful, and more reliable computers;
- ✓ **Microprocessors and Personal Computers (1970s):** The development of microprocessors, such as the Intel 4004, paved the way for the creation of personal computers. Companies like Apple and Microsoft emerged during this era.
- ✓ **Home Computers and the Internet (1980s):** The 1980s saw the rise of home computers like the Commodore 64 and the IBM PC. Additionally, the development of the internet began during this period;
- ✓ **Advancements in Processing Power (1990s-Present):** Moore's Law, which predicts the doubling of transistors on a microchip approximately every two years, has held true, leading to consistent advancements in computing power;
- ✓ **Mobile Computing and Smart Devices (21st Century):** The 21st century brought about the era of mobile computing, with smartphones and tablets becoming ubiquitous. Cloud computing also became a significant trend;
- ✓ **Artificial Intelligence and Quantum Computing (Present-Future):** Recent years have seen remarkable progress in artificial intelligence, machine learning, and quantum computing, pushing the boundaries of what computers can achieve.

2.2 Categories of computers

Computers are typically classified into several categories based on their size, functionality, and purpose. The main categories include:

- **Supercomputers:** They are designed for high-performance computing tasks. They are used for weather forecasting, studying climate, and generating physical or financial simulations. They currently utilize tens of thousands of microprocessors (IBM Blue Gene, 250,000 µP, Columbia, 10,240 µP).
- **Mainframe Computers:** They are large, powerful, and centrally managed computing systems designed for handling massive volumes of data and complex computing tasks. They are Used for data processing, handling large databases, and running enterprise-level applications.
- **Minicomputerss:** This category was popular in the 1960s and 1970s, and its role is now absorbed by other categories of computers.
- **Microcomputers:** These are small-sized computers whose central unit consists of one or more microprocessors. A microcomputer is an individual computer that operates autonomously. It can be classified into four groups: personal computer (PC), workstation, single-board computer, and single-chip microcomputer or microcontroller.

- ▶ **Servers:** They are computers designed to provide services or resources to other computers (clients) in a network which include web servers, file servers, and database servers.

- ▶ **Quantum computers:** Quantum computers are a type of computing device that leverages principles from quantum mechanics to perform computations. Unlike classical computers, which use bits to represent information as either 0 or 1, quantum computers use quantum bits or qubits that can exist in multiple states simultaneously.

- ▶ **Wearable computers:** Wearable computers are electronic devices that can be worn as accessories, embedded in clothing, or even implanted in the body. These devices are designed to provide functionality and convenience to the user such as smartwatches and fitness trackers.

2.3 Organization of a Computer

Computer (or microprocessor-based system) can read instructions from memory, acquire binary data as input for processing, and provide the results as output.

A microprocessor-based system has four essential components: Central Processing Unit, memory, input units, and output units, in addition to buses that transfer data between them, as shown in Figure 2.1."

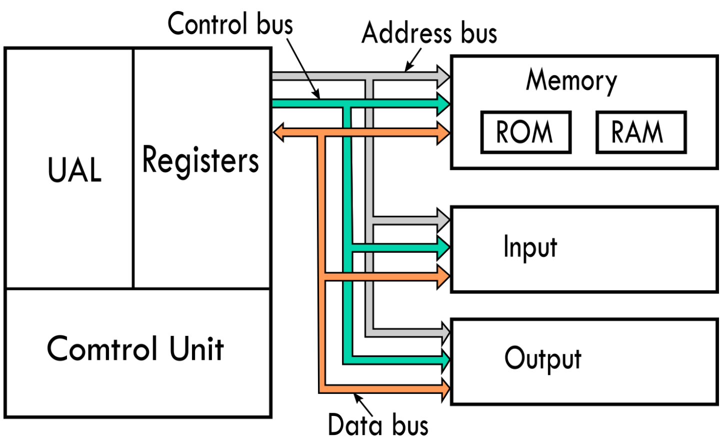

Figure 2.1: Essential Components of a Microprocessor System

2.3.1 Microprocessor

It is the primary component of a computer responsible for executing instructions and performing calculations. It acts as the brain of the computer, processing data and controlling other hardware components. Its main components include:

- **Arithmetic and Logic Unit (ALU):** The ALU performs arithmetic (+,-,x,/) and logical (AND, OR, NOT, XOR,..) operations.

- **Control Unit (CU):** The Control Unit manages the execution of instructions. It fetches instructions from memory, decodes them, and controls the flow of data between the CPU and other components.

- **Registers:** Registers are small, fast storage locations within the CPU used for temporarily holding data that is being processed..

2.3.2 Memory

Memory stores such binary information as instructions and data, and provides that information to the microprocessor whenever necessary. To execute programs, the microprocessor reads instructions and data from memory and performs the computing operations in its ALU section. Results are either transferred to the output section for display or stored in memory for later use.

ROM memory: ROM provides a reliable and non-volatile storage solution for the fundamental instructions and data that are integral to the system. such as bootstrap and firmware.

RAM memory: It is a volatile storage used to temporarily store data and instructions actively being processed by the system. It provides fast access times, allowing the processor to quickly retrieve and modify data during program execution.

2.3.3 Input/Output

Input/Output (I/O) allows the transfer of information between the microcomputer system and an external device. This transfer can be accomplished through three methods: programmed I/O, interrupt I/O, and direct memory access (DMA).

2.3.4 System bus

The microcomputer's system bus consists of three buses: address, data, and control buses. These buses connect the microprocessor (CPU) to each of the ROM, RAM, and I/O chips, allowing information transfer between the microprocessor and any of the other elements.

1. **Data Bus:** It is a bidirectional bus that allows data transfer between various components of a microprocessor system. The width of the data bus determines the volume of data that can be transmitted. In some microprocessors, the data pins are time-shared or multiplexed with addresses or other information pins.
2. **Address Bus:** It is a unidirectional bus, and its size determines the total number of memory addresses available for program execution by the microprocessor. The address bus is defined by the total number of address pins on the microprocessor chip, which, in turn, determines the direct addressing capability or the size of the main memory of the microprocessor.
3. **Control Bus:** The control bus consists of a number of signals that are used to synchronize the operation of the individual microcomputer elements. The microprocessor sends some of these control signals to the other elements to indicate the type of operation being performed such as ($\overline{CS}, \overline{RD}, \overline{WR}, EN, OE, CLK$..).

2.3.5 Clock Signals

The system clock signals are contained in the control bus. These signals generate the appropriate clock periods during which instruction executions are carried out by the microprocessor.

2.4 Input/Output devices

The I/O devices serve as an effective communication link between the microprocessor and the external environment, often referred to as "peripherals." These peripherals encompass devices like Keyboard, Mouse, Scanner, Microphone, Webcam, Contacts ,Screen, Speaker and printer. I/O devices typically differ from the microprocessor in terms of speed, word length, and data format. To ensure compatibility between the characteristics of I/O devices and the microprocessor, interface hardware circuitry is essential. Interfaces facilitate input and output transfers between the microprocessor and peripherals through an I/O bus.

Figure 2.2 illustrates the connection of certain peripherals to the microprocessor through input/output interfaces.

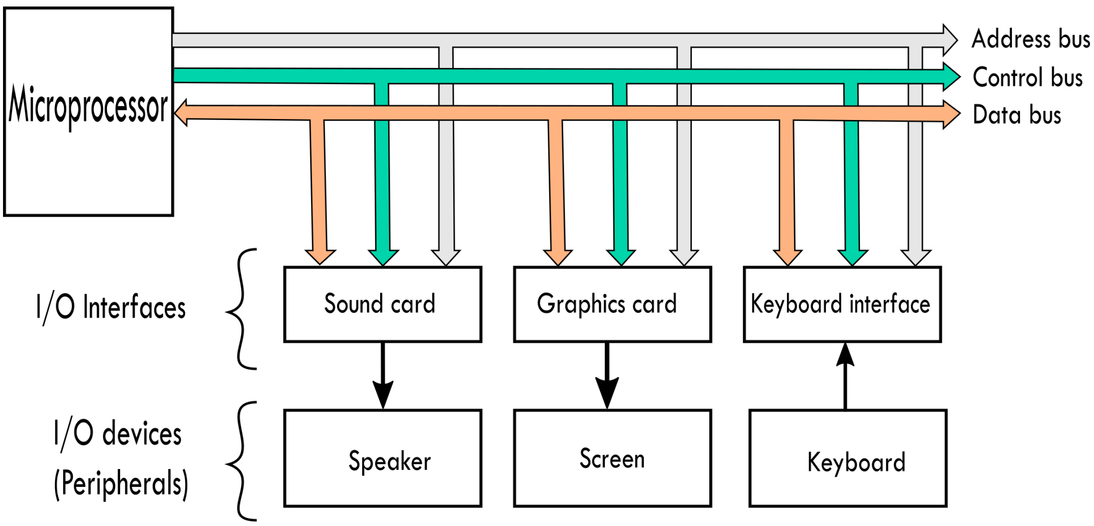

Figure 2.2: Connecting peripherals to the microprocessor through input/output interfaces

2.5 Microprocessor

In 1971, Intel successfully integrated all the transistors that constitute a processor onto a single integrated circuit, giving rise to the microprocessor. A microprocessor executes instructions and performs arithmetic and logic operations, playing a crucial role in processing data and controlling other components within a computer system.

The main characteristics of a microprocessor are:

(a) **Instruction Set :** The instruction set of a microprocessor is the list of commands it is designed to execute. Typical instructions include ADD, SUBTRACT, and STORE. Each instruction is coded as a unique bit pattern, recognized and executed by the microprocessor. If a microprocessor allocates 3 bits for instruction representation, it can recognize a maximum of 2^3 instructions in its instruction set. The instructions set varies depending on the microprocessor type and manufacturer. Microprocessor can execute from several tens to thousands of instructions.

(b) **Architecture:** Microprocessors can have different architectures, such as Complex Instruction Set Computing (CISC) or Reduced Instruction Set Computing (RISC), influencing how instructions are processed.

(c) **Data Bus Width:** It indicates the number of bits that can be transferred between the microprocessor and memory in a single operation. It also reflects the number of bits the processor can process simultaneously. The first microprocessors began with 4 bits, and currently, they have widths of 64 bits.

(d) **Address Bus Width:** The number of bits used to address memory locations, determining the maximum addressable memory.

(e) **Clock Speed:** The speed at which the microprocessor can execute instructions; the higher the clock speed, the more instructions the microprocessor executes, it is measured in hertz (Hz) or gigahertz (GHz).

(f) **Cache Memory:** Microprocessors often have small, high-speed memory caches to store frequently accessed instructions and data, reducing the need to fetch them from slower main memory. This feature was introduced in more advanced microprocessor.

(g) **Multicore Capability:** Many modern microprocessors are multicore, meaning they have multiple processing units on a single chip, allowing for parallel processing and improved performance.

Note: The metric used to express the processing speed of a microprocessor is how many Million Instructions Per Second (MIPS) it can execute.

2.6 History of Microprocessors

Table 1 summarizes the history of microprocessors from the appearance of the first microprocessor until recent years.

2.7 Assembly and Machine Languages

The processor can only execute binary code (1s and 0s). The group of binary instructions written in binary code is called machine language. It is the low-level programming language that permits communication with the processor. However, as direct programming in machine language is quite tedious, each binary code is associated with a mnemonic to facilitate human-readable program writing. The instruction set written using these mnemonics is called assembly language. The translator from assembly language to machine language is called an assembler. The example below makes clear the difference between these two languages with three instructions.

Exemple:

Mnemonic	*Operation Code*	*Operand*	*Function*
JMP	00_H	Address	*Jump to the specified address*
INR	$3C_H$	Implicit	*Increment the content of the accumulator*
ADD B	80_H	Data	*Add the content of register B to that of the accumulator*

Table 2.1: History of Microprocessor Development

Processor	**Year**	**No. of transistor**	**Clock speed**	**Address Bus**	**Data Bus**	**MIPS**
Intel 4004	1971	2300	108KHz	4-bit	4-bit	0.06
Intel 4040	1974	3000	108KHz	10-bit	4-bit	0.06
Intel 8008	1972	3500	108KHz	14-bit	8-bit	0.064
Intel 8080	1974	4500	2MHz	16-bit	8-bit	0.5
Intel 8085	1976	6500	3MHz	16-bit	8-bit	0.77
Intel 8086	1978	29000	10MHz	20-bit	16-bit	2.5
.	.	.	.	.	.	.
Intel 80386	1986	275000	33MHz	32-bit	32-bit	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
Pentium	1993	12M	66MHz	32-bit	32-bit	110
.	.	.	.	.	.	.
Pentium 4	2000	42M	1.4GHz	32-bit	32-bit	100
.	.	.	.	.	.	.
Core 2 Duo	2006	291M	2.4GHz	64-bit	64-bit	2200
.	.	.	.	.	.	.
Core i7	2008	781M	3.33GHz	64-bit	64-bit	.
.	.	.	.	.	.	.

2.8 High-level Languages

To bring the language closer to human comprehension and be machine-independent , high-level languages are designed, rendering programs more readable and understandable. Examples of such languages include Python, Java, C++, and others. Programs written in high-level languages undergo translation into machine language through a process known as compilation or interpretation. This process is carried out by a compiler or an interpreter, respectively. Figure 2.3 clarifies the three levels of programming languages with examples.

2.9 Von Neumann and Harvard Architectures

There are two fundamental computer architectures that differ in the way they handle data and instructions.

2.9.1 Von Neumann Architecture

In this architecture, as shown in Figure 2.4, the data and the program codes are both stored in the computer's memory in the same address space.

2.9.2 Harvard Architecture

This architecture, as illustrated in Figure 2.5, stores code and data in two distinct memories that operate independently, each possesses its own communication path (i.e., bus) that enables the simultaneous transfer of data and instructions, improving the execution time.

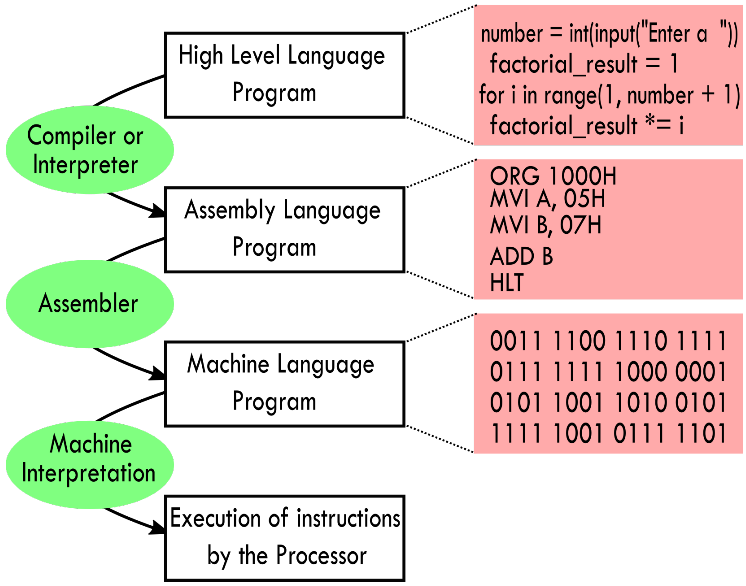

Figure 2.3: Levels of Programming Languages

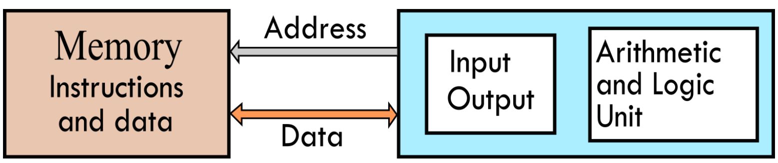

Figure 2.4: Principle of Von Neumann Architecture

This architecture is often used in Digital Signal Processing (DSP) and microcontrollers, notably Microchip's PIC and Atmel's AVR.

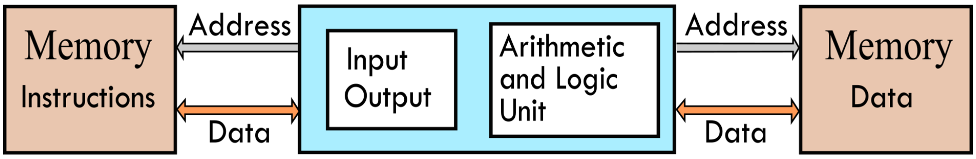

Figure 2.5: Principle of Harvard Architecture

Example:

LDA 2000_H *Load the content of the address into the accumulator*

The LDA instruction and the address 2000_{H} are in the same memory for Von Neumann. For Harvard, LDA is in the instruction memory, and 2000_{H} is in the data memory.