1.COMPUTER ORGANIZATION

Content

- 1.1 Turing model
- 1.2 Von Neumann model
- 1.3 Computer generations
- 1.4 Subsystems and the role of subsystems
- 1.5 Central Processing Unit
- 1.6 Memory: main memory and cache memory
- 1.7 Input/Output subsystems
- 1.8 Different architectures

Objectives

**After studying this chapter, the student should be able to:**

- Define the Turing model of a computer.

- Define the von Neumann model of a computer.

- Describe the three components of a computer: hardware, data, and software.

- List topics related to computer hardware.

- List topics related to data.

- List topics related to software.

- Give a short history of computers.

1-TURING MODEL

1. Introduction

- The idea of a universal computational device was first described by Alan Turing in 1937.

- All computation could be performed by a special kind of a machine, now called a **Turing machine.**

- The model on the actions that people perform when involved in computation.

- Actions into a model for a computational machine that has really changed the world.
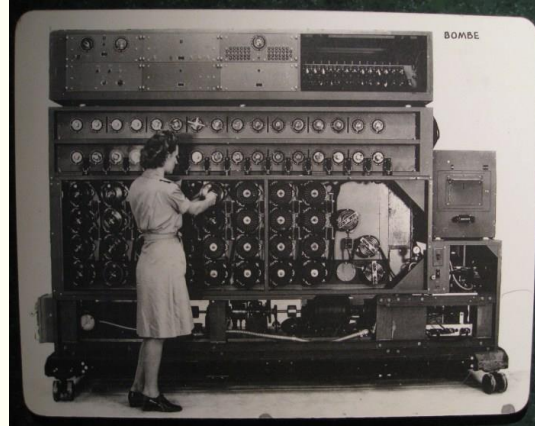
2. Data processors

- A computer acts as a black box that accepts input data, processes the data, and creates output data (Figure 1.1).

- This model could represent a specific-purpose computer (or processor) that is designed to do a single job, such as controlling the temperature of a building or controlling the fuel usage in a car.

- However, computers, as the term is used today, are general-purpose machines. They can do many different types of tasks
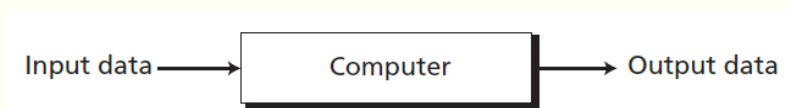
Input data → Computer → Output data

Figure -1.1 A single-purpose computing machine

3. Programmable data processors

An extra element is added to the specific computing machine: the *program*.

- A **program** is a set of instructions that tells the computer what to do with data.

The **output data** depends on the combination of two factors: the **input data** and the **program**.

- With the same input data, we can generate different output if we change the program.

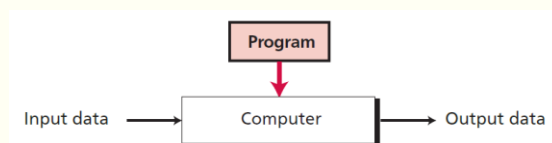- Similarly, with the same program, we can generate different outputs if we change the input data.

Figure -1.2 : A computer based on the Turing model: programmable data processor

4. The universal Turing machine

- A universal Turing machine, a machine that can do any computation if the **appropriate program is provided**, was the first description of a modern computer.

- It can be proved that a very powerful computer and a universal Turing machine can compute the same thing

- A universal Turing machine is capable of computing anything that is computable.

Figure - 1.3 Programmed Data Processor (PDP-1)

2-VON NEUMANN MODEL

1. Introduction

- The modern microcomputer has roots going back to USA in the 1940's.

- Of the many researchers, the Hungarian-born mathematician, John von Neumann (1903-57), is worthy of special mention.

- He developed a very basic model for computers which we are still using today.

Figure - 1.4. John von Neumann (1903-57). Progenitor of the modern, electronic PC.

2. Von Neumann Model Subsystems

Von Neumann model is divided into 4 subsystems :

- **Memory** is the storage area. This is where programs and data are stored during processing

- A**rithmetic logic unit (ALU)** is where calculation and **logical operations** take place

- **Control unit** controls the operations of the memory, ALU, and the input/output

- The **input subsystem** accepts input data and the program from outside the computer, while the **output subsystem** sends the result of processing to the outside world.
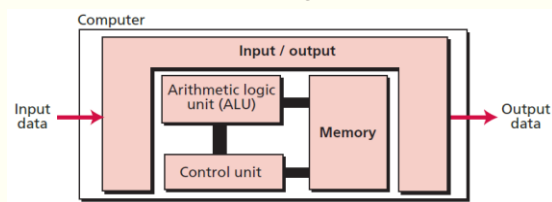

Figure 1.5 The Von Neumann model Subsystems

3. The stored program concept

- The von Neumann model states that the program must be stored in memory.

- This is totally different from the architecture of early computers in which only the data was stored in memory: the programs for their task were implemented by manipulating a set of switches or by changing the wiring system.

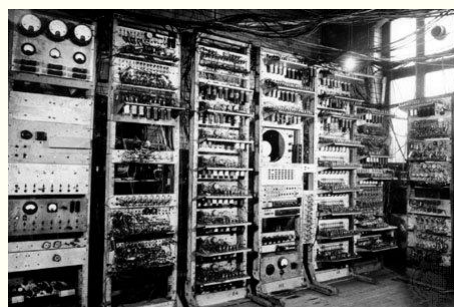- The memory of modern computers hosts both a program and its corresponding data

Figure 1.6 The Manchester Mark I, the first stored-program digital computer, 1949.

4. Sequential execution of instructions

- A program is made of a finite number of instructions. In this model, the control unit fetches one instruction from memory, decodes it, then executes it.

- One instruction may request the control unit to jump to some previous or following instruction, but this does not mean that the instructions are not executed sequentially.
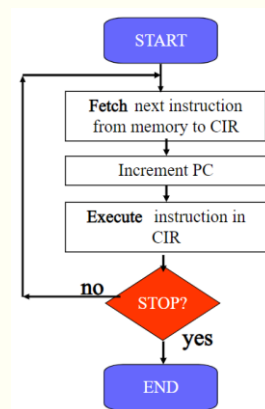

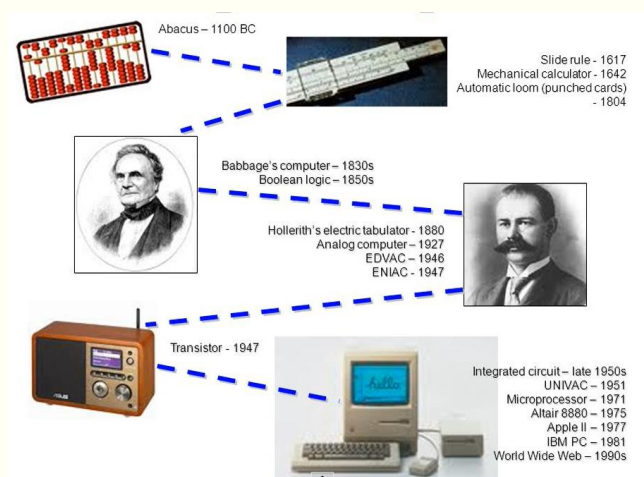**Figure 1.7** Fetches Execute cycle

History of Computers

Figure 1.7 History of Computers
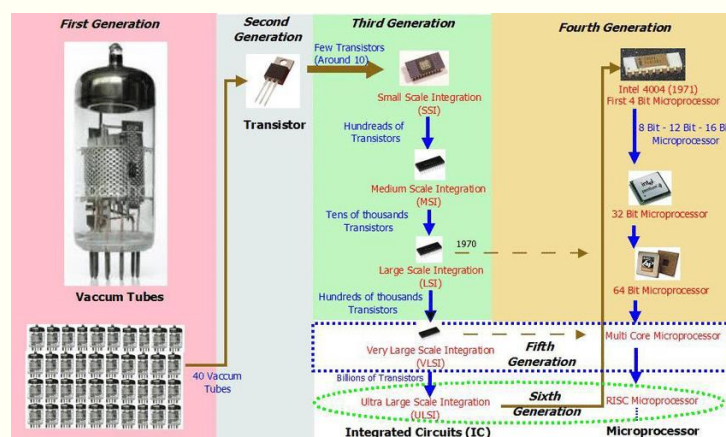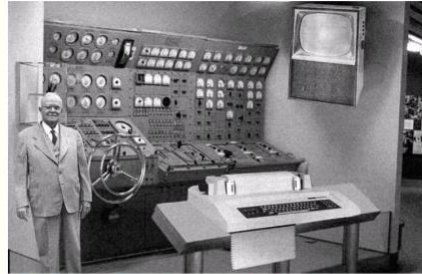
3 - COMPUTER GENERATIONS

1. Overview

Figure 1.8 Generation of Computers

2. First Generation (1945-1956)

- First Generation Computers were working during the 1940-1956 with proper maintenance of Vacuum Tubes on those computers. Vacuum Tubes most useful to process the data in memory.

- First generation computers use more power from electricity and that produce high heat.

- Those devices vulnerable to the attacks and get malfunctions.

First Generation Computers were working during the 1940-1956 with proper maintenance of Vacuum Tubes on those computers.

Figure 1.9 First Generation Computers

3. Second Generation Computers (1959-1965)

- Second-generation computers (roughly 1959–1965) used transistors instead of vacuum tubes. This reduced the size of computers, as well as their cost, and made them affordable to small and medium-size corporations.

- Two high-level programming languages, **FORTRAN** and **COBOL** (see Chapter 9), were invented and made programming easier.

Second Generation computers were working in between the 1956-1963 with Transistors.

Figure 1.10 Second Generation Computers

4. Third Generation Computers (1965-1975)

- The invention of the integrated circuit (transistors, wiring, and other components on a single chip) reduced the cost and size of computers even further.

- Minicomputers appeared on the market. Canned programs, popularly known as software packages, became available.

- A small corporation could buy a package, for example for accounting, instead of writing its own program.
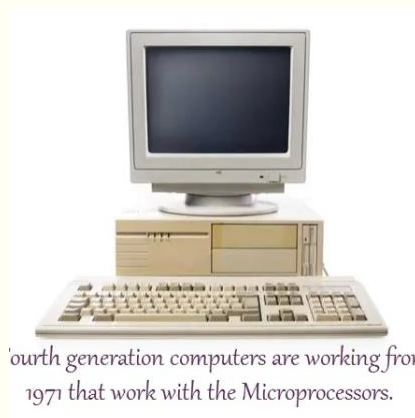

Third generation computers were working during the 1964-1971 with Integrated Circuits.

Figure 1.11 Third Generation Computers

5. The fourth generation (1975–1985)

- The fourth generation (approximately 1975–1985) saw the appearance of microcomputers.

- The first desktop calculator, the Altair 8800, became available in 1975.

- Advances in the electronics industry allowed whole computer subsystems to fit on a single circuit board.

Fourth generation computers are working from 1971 that work with the Microprocessors.

Figure 1.12 The fourth Generation Computers

6. Fifth generation (Present and Beyond)

- Fifth Generation computing devices, based on artificial intelligence, are still in development, though there are some applications such as voice recognition, that are being used today.

- The ability to translate a foreign language is also moderately possible with fifth generation computers.

- The goal of fifth generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

ARTIFICIAL INTELLIGENCE

ASIMO ROBOT
ARTIFICIAL INTELLIGENCE
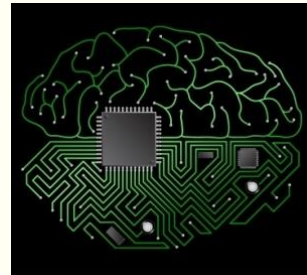
FUTURE COMPUTER BRAIN

Figure 1.13 The Fith Generation Computers

4 - SUBSYSTEMS AND THE ROLE OF SUBSYSTEMS

INTRODUCTION

- We can divide the parts that make up a computer into three broad categories or subsystem: **the central processing unit (CPU)** , **the main memory**, and the **input/output subsystem**.
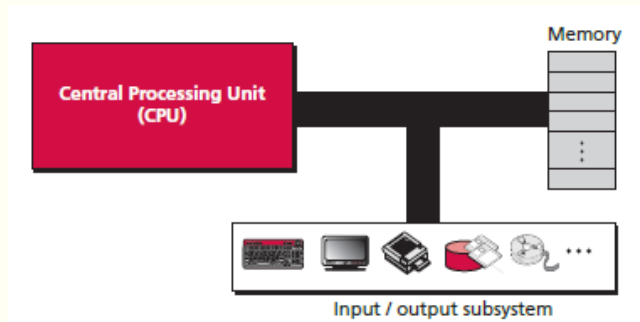

Figure 1.18 *Computer hardware (subsystems)*

5 - CENTRAL PROCESSING UNIT

Introduction

- The **central processing unit (CPU)** performs operations on data.

- In most architectures It has three parts: an arithmetic logic unit (ALU), a control unit, and a set of registers
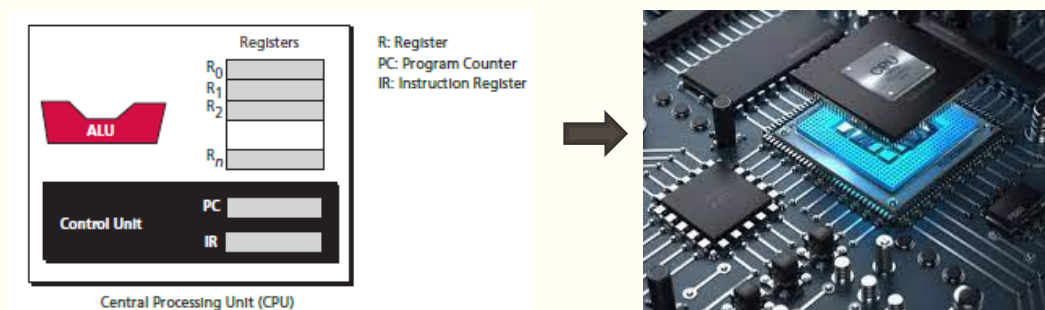

Figure 1.19 *Central processing unit (CPU)*

The arithmetic logic unit (ALU)

- **The arithmetic logic unit (ALU)** performs **logic**, **shift**, and **arithmetic operations** on data.

- **Logic operations:** NOT, AND, OR, and XOR.

- **Shift operations:** logic shift operations and arithmetic shift operations

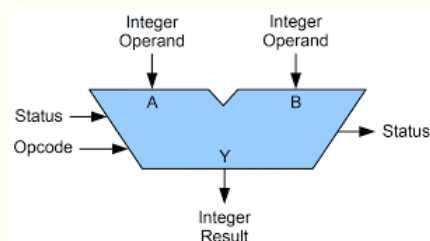- Arithmetic operations: arithmetic operations on integers and reals.

Figure 1.20 Construction of an ALU

Registers

- **Registers are fast stand-alone storage locations that hold data temporarily.** Multiple registers are needed to facilitate the operation of the CPU.

  ❑ **Data registers**
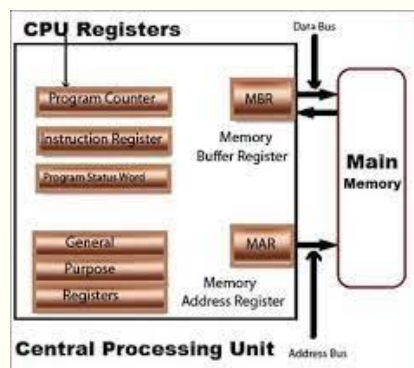 ❑ **Instruction register**
  ❑ **Program counter**

Figure 1.21 Construction of an Register

The control unit

- **The control unit controls the operation of each subsystem**. Controlling is achieved through signals sent from the control unit to other subsystems.
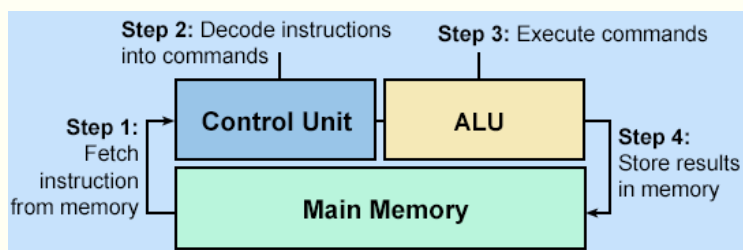

Figure 1.22 Using an control unit

6-MAIN MEMORY

Introduction

- **Main memory** consists of a collection of storage locations, each with a unique identifier, called an **address**.

- Data is transferred to and from memory in groups of bits called **words**.

Figure 1.23 Main memory

Address space

- To access a word in memory requires an identifier. Although programmers use a name to identify a word (or a collection of words), at the hardware level each word is identified by an address.

- The total number of uniquely identifiable locations in memory is called the address space. For example, a memory with **64 kilobytes** and a word size of **1 byte** has an address space that ranges from **0** to **65,535**.

Unit	Exact Number of Bytes	Approximation
kilobyte	2^{10} (1024) bytes	10^3 bytes
megabyte	2^{20} (1,048,576) bytes	10^6 bytes
gigabyte	2^{30} (1,073,741,824) bytes	10^9 bytes
terabyte	2^{40} bytes	10^{12} bytes

Table 6.1 Memory units

Examples

- **Example 5.1**

- A computer has 32 MB (megabytes) of memory. **How many bits** are needed to address **any single byte** in memory?

- **Solution** The memory address space is **32 MB**, or 225 (25 × 220). This means that we need log2 225, or **25 bits**, to address each byte.

- **Example 5.2**

- A computer has 128 MB of memory. **Each word in this computer is eight bytes**. **How many bits** are needed to address **any single word** in memory?

- **Solution** The memory address space is **128 MB**, which means 227. However, each word is eight (23) bytes, which means that we have 224 words. This means that we need log2 224, or **24 bits**, to address each word.

Memory types

Two main types of memory exist: RAM and ROM.

- **Random access memory (RAM)**

  ❑ Static RAM (**SRAM**)
  ❑ Dynamic RAM (**DRAM**) – e.g., **DDR SDRAM**
  - ❑ **NAND-based Flash RAM**

- **Read-only memory (ROM)**

  ❑ Programmable read-only memory (**PROM**).
  ❑ Erasable programmable read-only memory (**EPROM**).
  ❑ Electrically erasable programmable read-only memory (**EEPROM**).

Memory hierarchy

- Computer users need a lot of memory, especially memory that is very fast and inexpensive. This demand is not always possible to satisfy**— very fast memory is usually not cheap**. A compromise needs to be made.

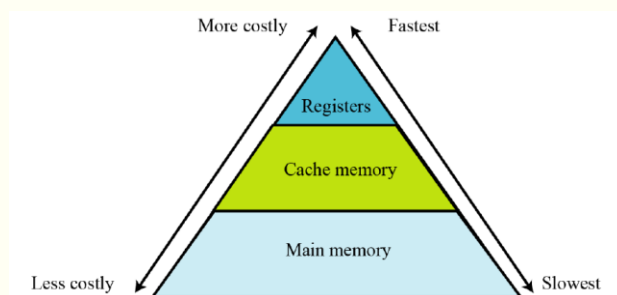- The solution is **hierarchical** levels of memory.

Figure 1.24 Hierarchical levels of memory.

Cache memory

- Cache memory is **faster than main memory**, but **slower than the CPU and its registers**.

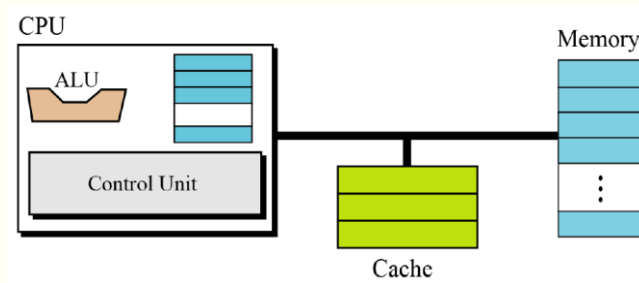- Cache memory, which is normally small in size, is placed between the CPU and main memory.

Figure 1.25 Hierarchical levels of memory.

7 - INPUT/OUTPUT SUBSYSTEM

Introduction

- The input/output (I/O) subsystem in a computer **is the collection of devices**. T

- his subsystem allows a computer to communicate with the outside world and to store programs and data even when the power is off.

- Input/output devices can be divided into two broad categories: **non-storage** and **storage** devices.

Non-storage & Storage devices

- **Non-storage devices** allow the CPU/memory to communicate with the outside world, but they cannot store information.

  ❑ **Keyboard and monitor**
  ❑ **Printer**

- **Storage devices**, although classified as I/O devices, can store large amounts of information to be retrieved at a later time. They are cheaper than main memory, and their contents are **nonvolatile**—that is, **not erased when the power is turned off**. They are sometimes referred to as auxiliary storage devices. We can categorize them as either **magnetic** or **optical**.

Storage devices (Magnetic or Optical)
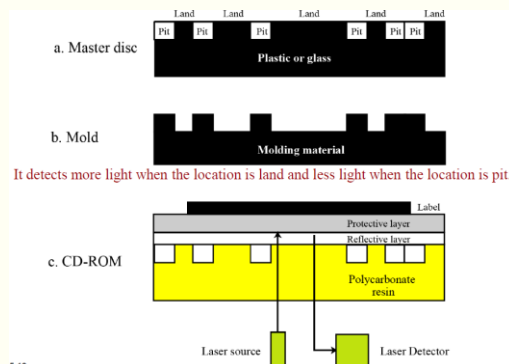
HDD disk
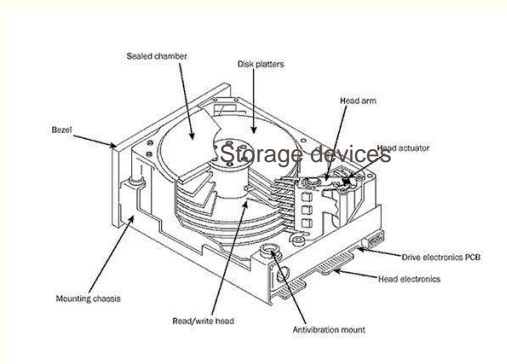
CD-ROM

Figure 1.26 Storage devices

SSD – Solid State Disk

- A solid-state drive (SSD uses NAND-based flash memory, which retains data without power.

- It is also known as a **solid-state disk** or electronic disk, though it contains no actual "disk" of any kind, nor motors to "drive" the disks.

- SSD technology uses electronic interfaces compatible with traditional block input/output (I/O) hard disk drives, thus permitting simple replacement in common applications, like SATA Express.

- SSDs are about 7 to 8 times more expensive per unit of storage than HDDs.

Figure 1.27 Solid State Disk

8 - SUBSYSTEM INTERCONNECTION

Introduction

- In this section, we explore how these three subsystems (CPU, main memory, and I/O) are interconnected.

- The interconnection plays an important role because information needs to be exchanged between the three subsystems.

Connecting CPU and memory

- The CPU and memory are normally connected by three groups of connections, each called a **bus**: *data bus*, *address bus* and *control bus*.

Figure 1.28 CPU and memory

Connecting I/O devices

- I/O devices are electromechanical, magnetic, or optical devices and **also operate at a much slower speed than the CPU/memory**.

- There is a need for some sort of intermediary to handle this difference. Input/output devices are therefore attached to the buses through input/output controllers or interfaces.

- There is one specific controller for each input/output device
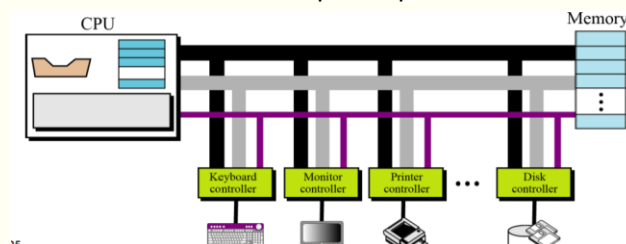
Figure 1.29 Connecting I/O devices

SCSI controller to connect I/O devices

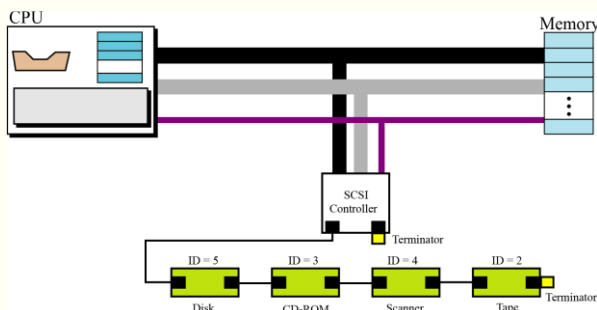- **Small Computer System Interface (SCSI is created in 1984): 32 components**

Figure 1.30 SCSI controller

FireWire controller to connect I/O devices

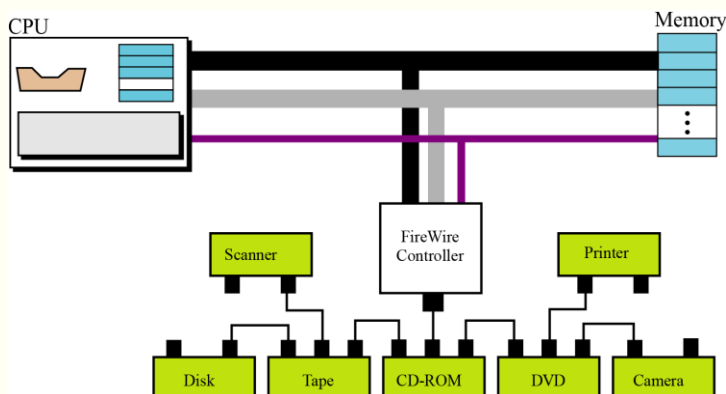- **IEEE Standard 1394: 400Mbps and 63 devices**

Figure 1.31 FireWire controller

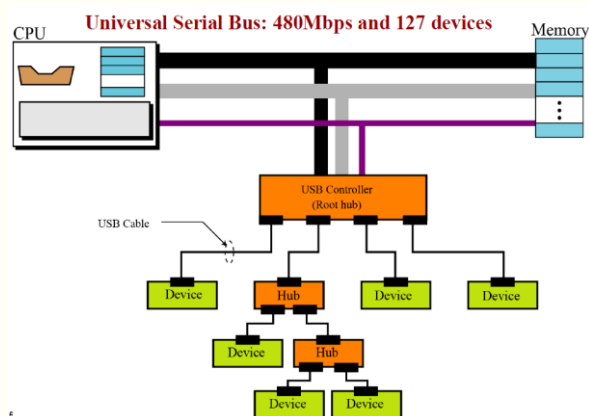## USB controller to connect I/O devices

Figure 1.32 USB controller

Addressing input/output devices

- The CPU usually uses the same bus to read data from or write data to main memory and I/O device. The only difference is the instruction.

- If the instruction refers to a word in main memory, data transfer is between main memory and the CPU.

- If the instruction identifies an I/O device, data transfer is between the I/O device and the CPU.

- here are two methods for handling the addressing of I/O devices: **isolated I/O** and **memory-mapped I/O**.
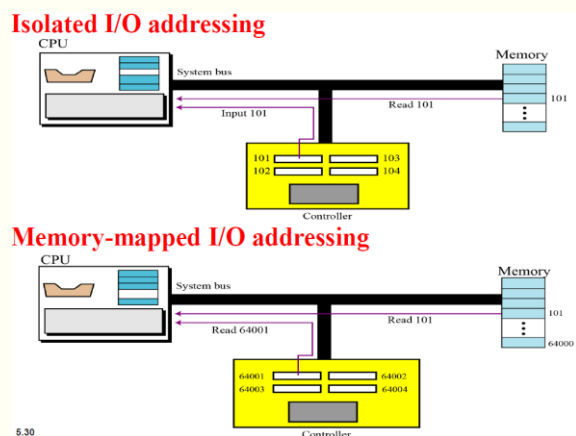
Addressing input/output devices (cont)

Figure 1.33 Addressing input/output

9 - DIFFERENT ARCHITECTURES

Introduction

- The architecture and organization of computers has gone through many changes in recent decades.

- In this section we discuss some common architectures and organization that differ from the simple computer architecture we discussed earlier.

CISC

- **CISC** stands for **complex instruction set computer.** The strategy behind CISC architectures is to **have a large set of instructions, including complex ones**.

- Programming CISC-based computers is easier than in other designs because there is a single instruction for both simple and complex tasks. Programmers, therefore, **do not have to write a set of instructions to do a complex task**.
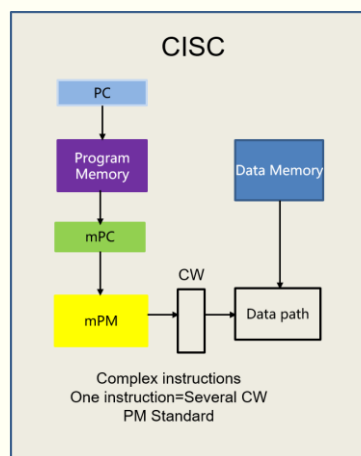

Figure 1.34 Complex instruction set computer.

RISC

- **RISC** stands for **reduced instruction set computer**. The strategy behind RISC architecture is to **have a small set of instructions that do a minimum number of simple operations**.

- **Complex instructions are simulated using a subset of simple instructions**. Programming in RISC is more difficult and time-consuming than in the other design, because most of the complex instructions are simulated using simple instructions.
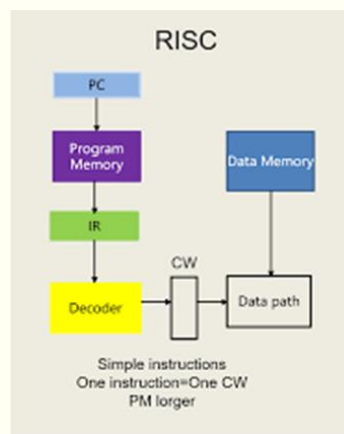

Figure 1.35 Reduced instruction set computer.

Pipelining

- Modern computers use a technique called **pipelining** to improve the **throughput** (the total number of instructions performed in each period of time).

- The idea is that if the control unit can do two or three of these phases simultaneously, the next instruction can start before the previous one is finished.
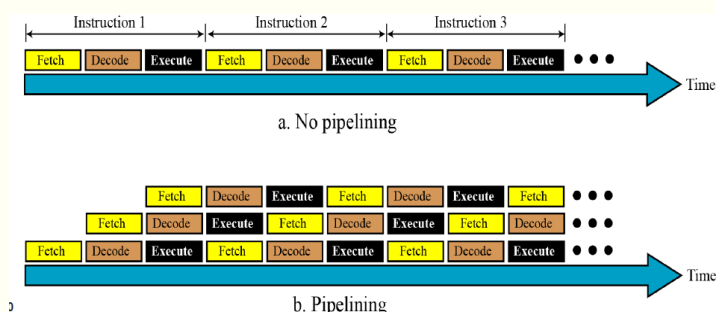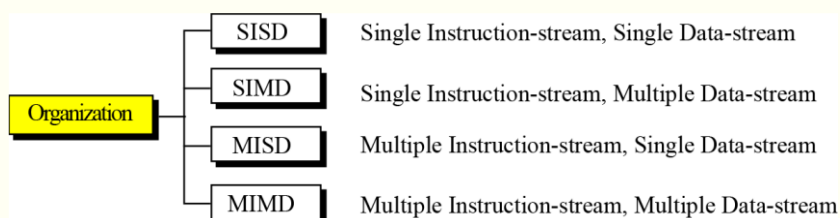

Figure 1.36 Pipelining technique.

Parallel processing

- Traditionally a computer had a single control unit, a single arithmetic logic unit and a single memory unit. With the evolution in technology and the drop in the cost of computer hardware, today we can have **a single computer with multiple control units, multiple arithmetic logic units and multiple memory units.**

- This idea is referred to as **parallel processing**. Like pipelining, parallel processing can improve throughput.
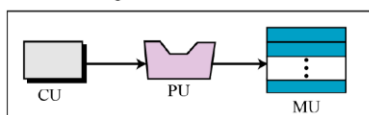
Organization		
	SISD	Single Instruction-stream, Single Data-stream
	SIMD	Single Instruction-stream, Multiple Data-stream
	MISD	Multiple Instruction-stream, Single Data-stream
	MIMD	Multiple Instruction-stream, Multiple Data-stream

Parallel processing (cont)

SISD organization

CU: Control unit
MU: Memory unit
PU: Processing unit

Figure 1.36 SISD organization

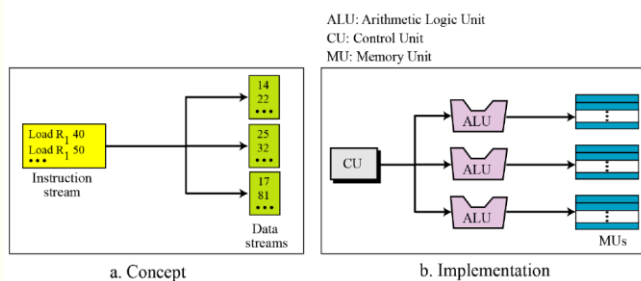# Parallel processing (cont)

SIMD organization

Figure 1.37 SIMD organization
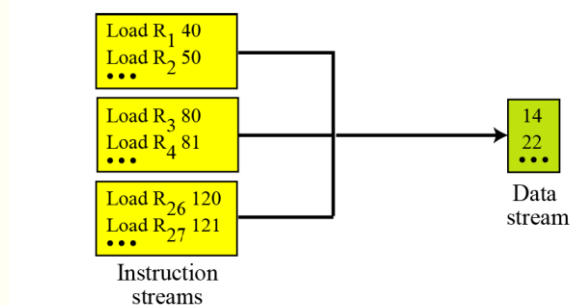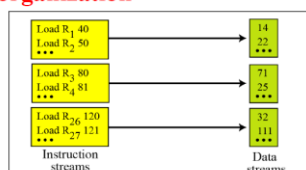
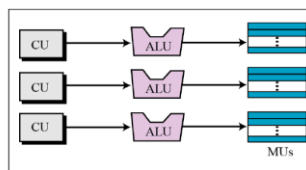# Parallel processing (cont)

MISD organization

Figure 1.38 Parallel processing.

Parallel processing (cont)

Figure 1.38 MIMD organization