(تقنيـات وتركيـــب الحاســـــــــوب)

Computer Organization College of Education for pure Science/ Ibn Al-Haitham
Dept. of Computer Science 1st Class 2021-2022
Dr. Ali Yahya M.SC. Samera Shams

CHAPTER ONE

Introduction

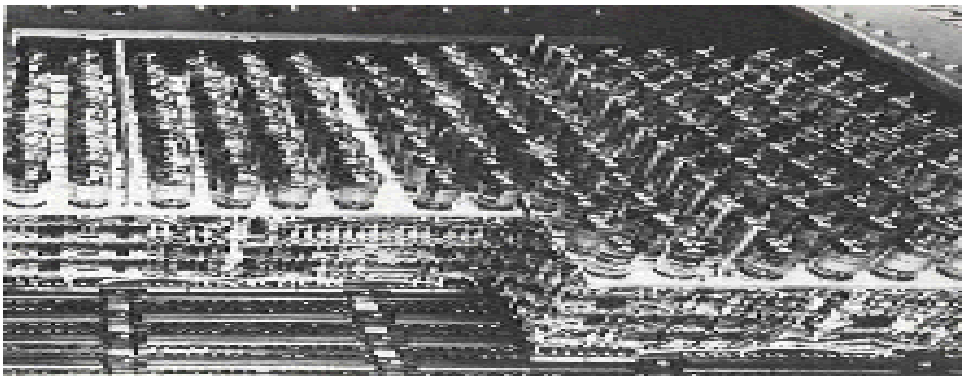
Introduction

The technological advances witnessed in the computer industry are the result of a long chain of immense and successful efforts made by two major forces. These are the academia, represented by university research centers, and the industry, represented by computer companies. It is, however, fair to say that the current technological advances in the computer industry owe their inception to university research centers. In order to appreciate the current technological advances in the computer industry, one has to trace back through the history of computers and their development.

The history of computer development is often referred to in terms of five distinct eras, or "generations" of computing devices. Each generation of computer is characterized by a major technological development that fundamentally changed the way computers operate.

The First Generation:

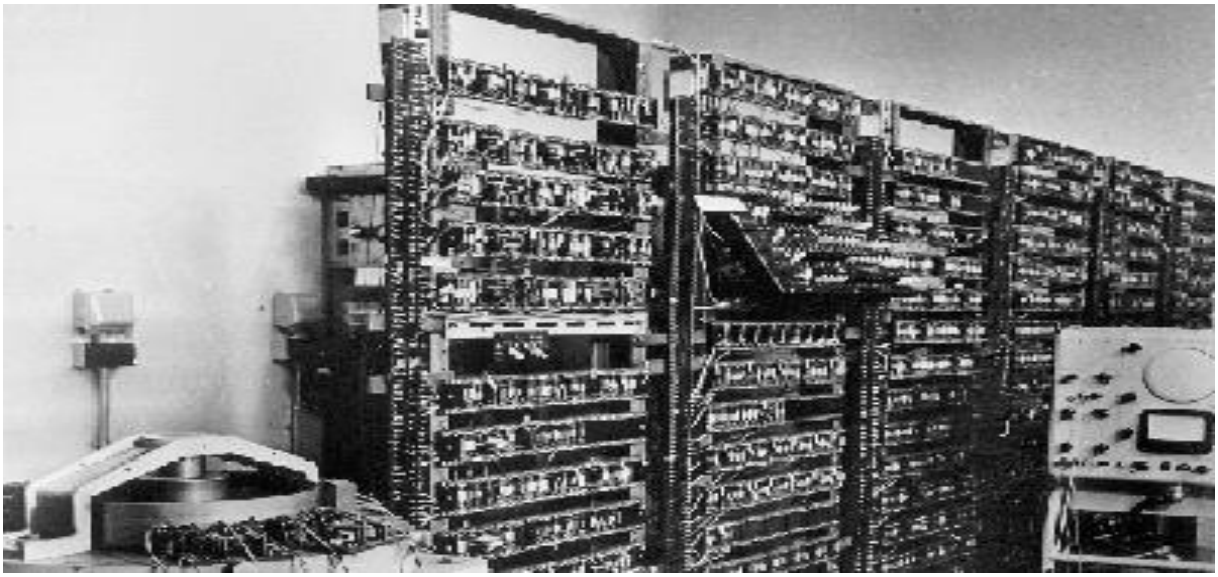
The first computers used **vacuum tubes** for circuitry, magnetic drums and magnetic cores for memory, and were often enormous, taking up entire rooms. They were very expensive to operate and in addition to using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions.



Vacuum tube processing unit in a first-generation computer

The Second Generation:

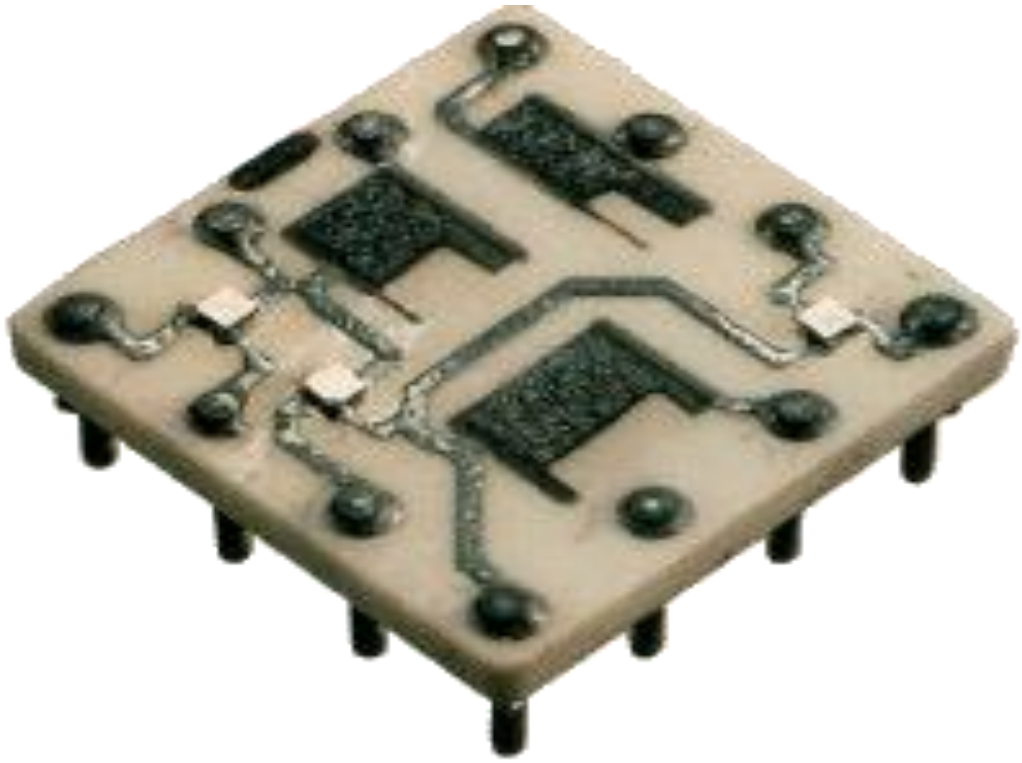
Transistors replaced vacuum tubes and ushered in the second generation of computers. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, more energy-efficient and more reliable than their first-generation predecessors.



Transistor-based processing unit in a second-generation computer

The Third Generation:

The third generation is characterized by the development of the integrated circuit - a complete electrical circuit whose components (transistors, capacitors,) are put onto a small "chip" made of silicon. Users interacted with third generation computers through keyboards and monitors and interfaced with an operating system, which allowed the device to run many different applications at one time.



An early chip, containing 3 transistors

The Fourth Generation:

The fourth generation computers started with the invention of Microprocessor. The Microprocessor contains thousands of ICs.Which makes it, more powerful and reliable, small in size, fast processing than previous generations.

The original IBM PC

The Fifth Generation:

The Fifth Generation of artificial intelligence and robotics. Increase in speed and storage capacity, and Development in the field of networking.

Definition of computer organization

Computer organization is a study of a Computer Architecture. E.g. Memory, Registers, RAM, ROM, CPU, ALU architecture, what different parts makes a computer.

Data Representation:

Data Representation refers to the methods used internally to represent information stored in a computer. Computers store lots of different types of information:

- Numbers.
- Text.
- Graphics of many varieties (stills, video, animation).
- Sound.

Memory Structure in Computer:

- Memory consists of bits (0 or 1)
  a single bit can represent two pieces of information
- bytes (=8 bits)

  a single byte can represent 256 = $2\times2\times2\times2\times2\times2\times2\times2 = 2^8$

Unit	Abbreviation	Pronounced	Approximate Value (bytes)	Actual Value (bytes)

Kilobyte	**KB**	**Kill-uh-bite**	**1,000**	$2^{10} = 1,024$
Megabyte	**MB**	**Mehg-uh-bite**	**1,000,000** **1 million**	**1,048,576** **1024k =** $2^{10}.2^{10} = 2^{20}$
Gigabyte	**GB**	**Gig-uh-bite**	**1,000,000,000** **1 billion**	**1,073,741,824**
Terabyte	**TB**	**Terr-uh-bite**	**1,000,000,000,000** **1 trillion**	**1,009,511,627,776**

Binary Number:

Normally we write numbers using digits 0 to 9. This is called decimal numbers (base10). However, any positive integer (whole number) can be easily represented by a sequence of 0's and 1's. Numbers in this form are said to be binary numbers in (base 2).

Decimal numbers use a positional system based on powers of 10 to indicate their value. The number 1 2 3 is really 1 hundred +2 tens + 3 ones. The value of each position is determined by ever-higher powers of 10, read from left to right.

Base 2 works the same way, just with different powers. The number 101 in base 2 is really 1 four + 0 twos +1 one (which equals 5 in base 10).

Binary Representation of Numbers:

Binary Numbers are numbers represented with 0's and 1's.

They work much the same way as our normal decimal numbers except instead of 10 digits (0 to 9) there are only 2 digits (o and 1).

Decimal number (subscript indicates base 10):

8401.32

$= 8\times1000 + 4\times100 + 0\times10 + 1\times1 + 3\times1/10 + 2\times1/100$

$= 8\times10^3 + 4\times10^2 + 0\times10^1 + 1\times10^0 + 3\times10^{-1} + 2\times10^{-2}$

Binary number (subscript indicates base):

1101.01

$= 1\times8 + 1\times4 + 0\times2 + 1\times1 + 0\times1/2 + 1\times1/4$

$= 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$

Which in base 10 is 1101.01= 8 + 4 + 1 + 1/4 = 13.25

Text:

Text can represented easily by assigning a unique numeric value for each symbol used in the text. For example, the widely used ASCII code (**American Standard Code for Information Interchange**) defines 128 different symbols and assigns to each a numeric code between 0 and 127.

In ASCII, an "A" is 65, "B" is 66, "a" is97, "b" is 98, and so forth. When you save a file as plain text", it is stored using ASCII.

ASCII format uses 1 byte per character 1byte gives only 256 (128 standard and non-standard) possible characters. The code value for any character can be

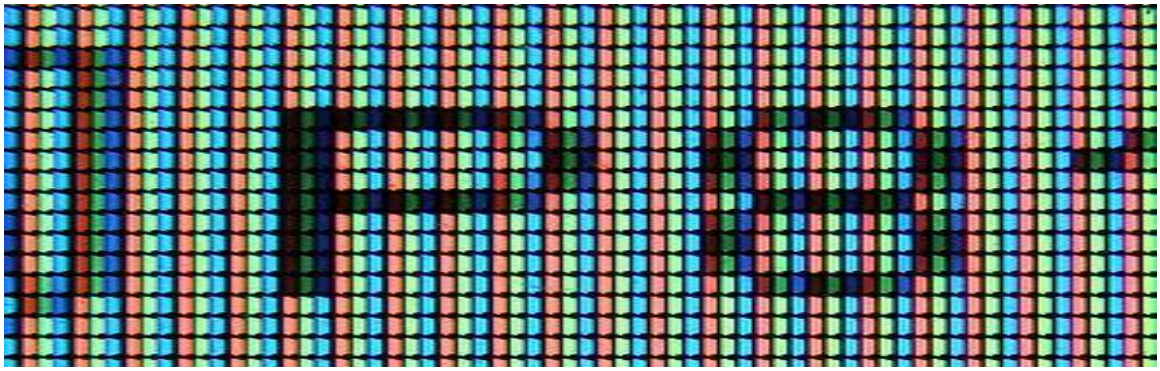
Converted to base 2 so any written message made up of ASCII characters can be converted to a string of 0′s and 1′s.

Graphics:

Graphics that are displayed on a computer screen consist of pixels: the tiny"dots" of color that collectively"paint " a graphic image on a computer screen.

The pixels are organized into many rows on the screen. In one common configuration, each row is 640 pixels long, and there are 480 pixel colomn.

Another configuration (and the one used on the screens in the lab) is 800 pixels per row with 600 colomn, which is referred to as a "resolution of 800×600". Each pixel has two properties: its location on the screen and its color. A pixel's color is represented by a binary code, and consists of a certain number of bits. Ina monochrome (black and white) image, only 1 bit is needed per pixel: 0 for black, 1 for white, for example. A 16 color image requires 4 bits per pixel. Modern display hardware allows for 24 bits per pixel, which provides an astounding array of 16.7 million possible colors for each pixel.



CHAPTER TWOO

COMPUTER ARCHITECTURE

Computer Architecture

Definitions of Computer:

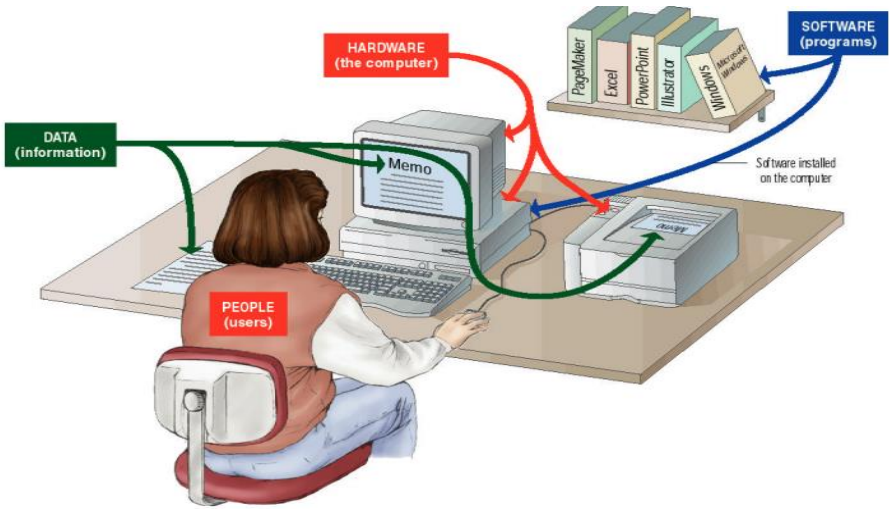
A computer is an electronic device, operating under the control of instructions stored in its own memory that can accept data, process the data according to specified rules, produce results, and store the results for future use.

Data and Information:

Computers process data into information. Data is a collection of unprocessed items, which can include text, numbers, images, audio, and video. Information conveys meaning and is useful to people.

The Complete Computer System consists of four parts:

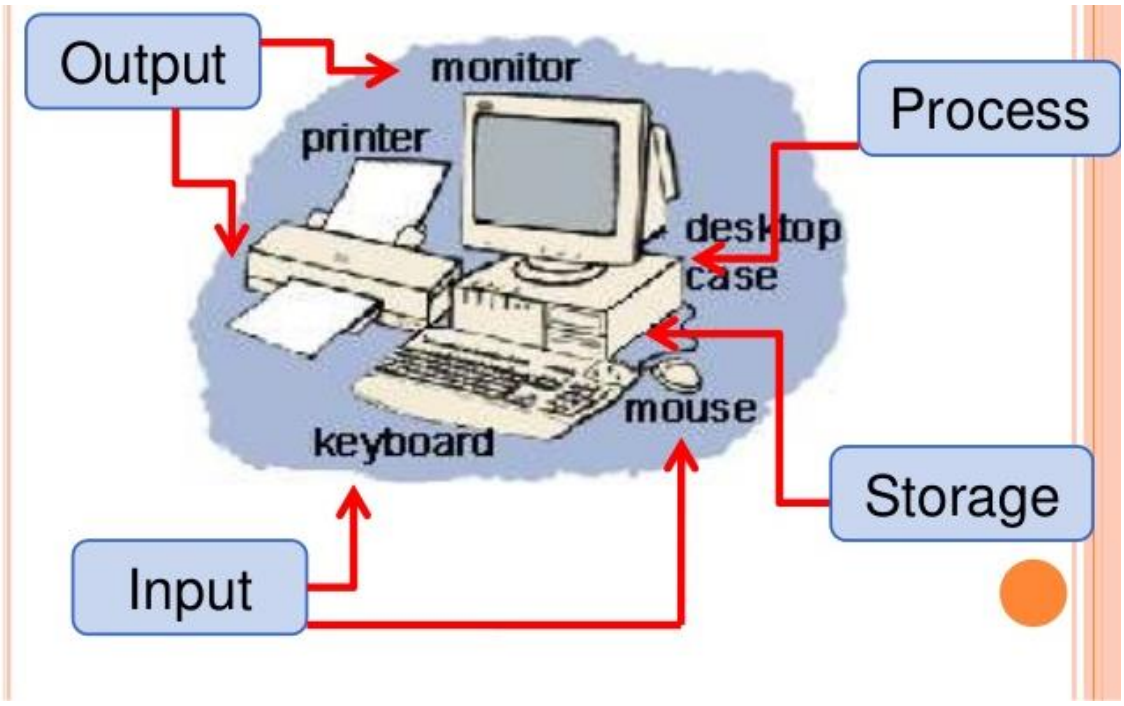
1- Hardware 2- Software 3- Users, and Data



- Hardware: is any part of the computer you can touch, such as (Disk, CD, CPU, Memory, I /O Device).
- Software: is a set of instructions that tells the computer what to do and how to do it, such as (Word Processing, Computer Games and programs.
- Users: Are people who use the software on the computer to do some tasks.
- Data: consists of individual facts or bits of information, the computer reads and stores data of all kinds, words, number, images or sound.

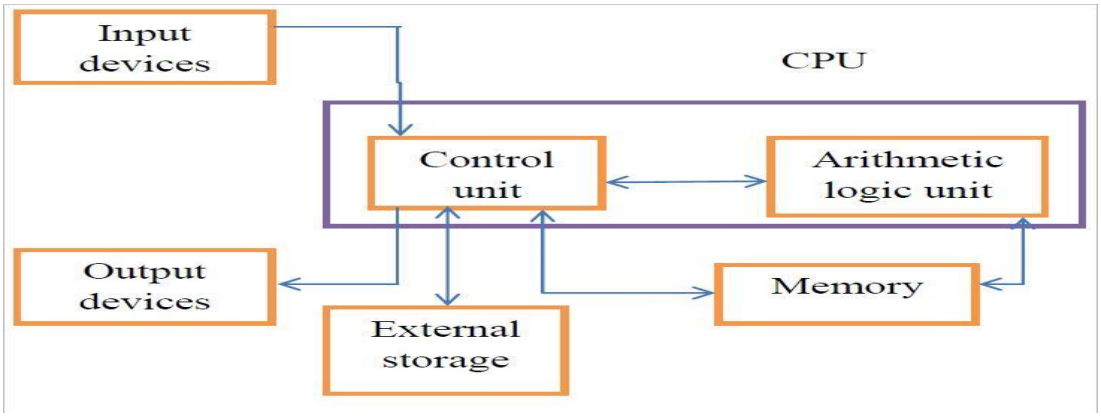
Parts of the computer itself (hardware):

1- Central Processing Unit (CPU)
2- Memory
3- Input and output devices
4- Storage device



1- Central Processing Unit(CPU):

Every PC′s system unit contains at least one chip called microprocessor or CPU, attached on the motherboard, to perform computer processing. The CPU follows the instructions of the software (or a program) to process data producing information.



Parts of CPU:

A- Arithmetic and Logic Unit (ALU):

Performs arithmetic operations such as (Addition, subtraction, multiplication and division) and logical operations such as (OR, AND, XOR) and controls the speed of those operations.

B- Control Unit:

The control unit has the following functions:

1- Read and interpret program instructions.

2- Directing operations inside the CPU.

3- Control the flow of data and instructions to and from main memory and input and output controllers units.

C- Register:

Fast memory used for storing the numbers in processor that he wants to perform his calculations, the arithmetic and logic unit cannot perform any mathematical operation only after bringing the numbers that wants conducting operations on them to register.

The register size is very important because determined data size that can the computer conducting the calculating, the size can be 32 bit or 64 bit.

2-Memory:

Is like an electronic scratch pad inside the computer. When you launch a program it is loaded into and run form memory. Data used by the program is also loaded into memory for fast access.

As new data is entered into the computer it is also stored in memory _ but only temporarily. The most common type of memory is called random access memory, or RAM.

3-Input and output devices (I\O devices):

A computer would be useless if you could not interact with it because the machine could not receive instruction or deliver the results of its work. **Input devices** accept data and instructions from the users or from another computer system (such as a computer on the Internet). **Output devices** return processed data to the user or to another computer system.

4- Storage device:

The computer needs a place to keep program files and related data when they are not in use. The purpose of storage is to hold data permanently. Computer users often confuse storage with memory, although the functions of storage and memory are similar, they work in different ways. There are three major distinctions between storage and memory:

1- There is more room in storage than in memory.
2- Contents are retained in storage when the computer is turned off, whereas the program or the data you put into memory disappear when you shut down the computer.
3- Storage is very slow compared to memory, but it is much cheaper than memory.

The Bus

A bus is a path between the components of computers. There are two main buses in computer: the internal (or system) bus and the external (or expansion) bus.

The system bus resides the motherboard and connects the CPU to other devices that reside on the motherboard. An expansion bus connects external devices, such as the keyboard, mouse, modem, printer, and so on, to the CPU. Cables from disk drives and other internal devices are plugged into the bus. The system bus has two parts: the data bus and the address bus.

- **The Data Bus**

The **data bus** is an electronic path that connects the CPU, memory, and the other hardware devices on the motherboard. The bus is a group of parallel wires. The number of wires in the bus affects the speed, at which data can travel between hardware components, because each wire can transfer 1 bit of data at a time, an 8-wire bus can move 8 bits at a time. A 16-bit bus can transfer 2 bytes, and a 32-bit bus can transfer 4 bytes at a time. Newer model computers have a 64-bit data bus, which transfers 8 bytes at a time. However, buses now are also being measured according to their **data transfer rates** the amount of data they can transfer in a seconds. The performance of computer buses usually is measured in **megabits per second** (Mbps) or **megabytes per second** (Mbps).

- **The Address Bus**

The **Address Bus** is a set of wires similar to the data bus. The Address Bus connects only the CPU and RAM and carries only memory addresses. Each byte in RAM is associated with a number, which is it memory address.

We have another type of bus is a **Control bus** to transmits control signals between different parts of the computer.

Machine Cycles:

Each time the CPU executes an instruction, it takes a series of steps. The completed series of steps is called a **machine cycle** itself can be divided into parts: the **instruction cycle** and the **execution cycle**. At the beginning of the machine cycle (that is, during the instruction cycle), the CPU takes two steps:

1- Fetching: before the CPU can execute an instruction, the control unit must retrieve (or fetch) a command or data from the computer's memory.
2- Decoding: before a command can be executed, the control unit must break down (or decode) the command into instructions that correspond to those in the CPU's instruction set.

At this point, the CPU is ready to begin the execution cycle:

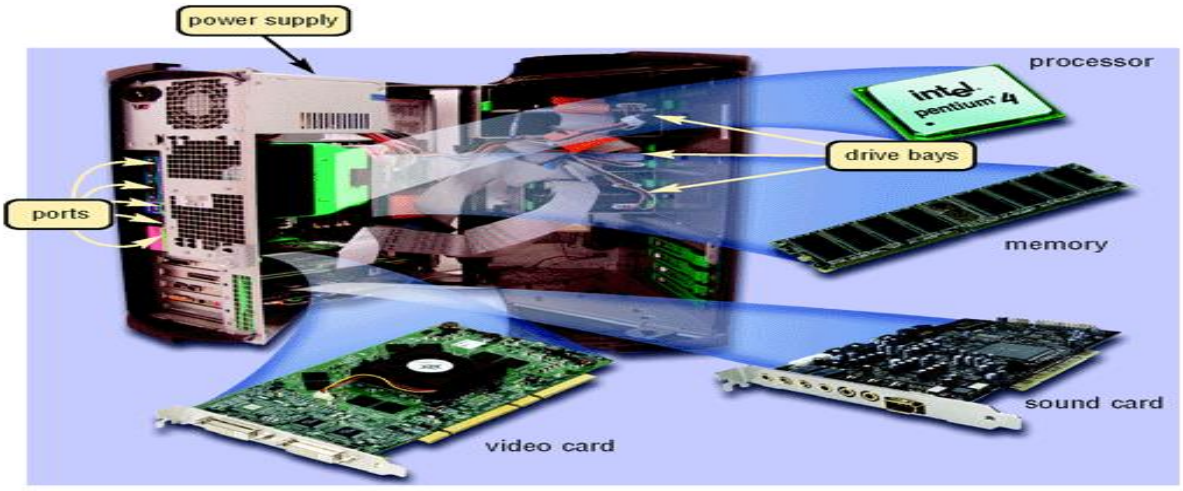
1- Executing: the CPU executes a command.
2- Storing: the CPU may be required to store the results of an instruction in memory (but this condition is not always required).

Depending on the type of processor in use, a machine cycle may include other steps. Although the process is complex, the computer can accomplish it an incredible speed, translating millions of instructions every second.

Mother Board:

A motherboard is the physical arrangement in a computer that contains the computer's basic circuitry and components. On the typical motherboard, the circuitry is affixed to the surface of a firm planar surface. The computer components included in the motherboard are processor, Memory, basic input/output system (BIOS), VGA and audio.

The motherboard provides many jobs where the distribution of electrical energy from power supply to the parts that are installed it also provide connectors (Ports) for each of the keyboard and mouse, printer, etc. The basic function of the motherboard is acting as communication environment and basic connections for all components where pass through the data and information to move from one part to another of the components of the device.



Ports:

is a connection point or interface between a computer and an external or internal device. Internal ports may connect such devices as hard drives and CD ROM or DVD drives; external ports may connect modems, printers, and other devices.

Types of computer ports include:

- Serial Ports: send and receive one bit at a time via a single wire pair such as modem, mouse.
- Parallel Ports: send multiple bits at the same time over several sets of wires such as printer. The main difference between a serial and parallel port is the way information is communicated: a parallel port is only able to transfer information from the hard drive, while a serial port can transfer information both to and from a hard drive.
- Universal Serial Bus (USB) Ports: the most common type of computer port used in today's computers. It can be used to connect keyboards, , game controllers, printers, scanners, digital cameras, and removable media drives can be used to connect many devices including all previously mentioned plus keyboards, scanners.
- Accelerated Graphics Port: The Accelerated Graphics Port (AGP) is a high-speed point-to-point channel for attaching a video card to a computer's motherboard, primarily to assist in the acceleration of 3D computer graphics. It is designed especially for the throughput demands of 3-D graphics.
- Peripheral Component Interconnect (PCI): is a computer bus for attaching hardware devices in a computer. These devices can take either the form of an integrated circuit fitted onto the motherboard itself.
- Industry Standard Architecture (ISA): is a computer bus standard for IBM compatible.

Interface:

Boundary across which two independent systems meet and act on, or communicate with each other. In computer technology, there are several types of interface:

- User interface – the keyboard, mouse, menus of computer system. The user interface allows the user to communicate with the operating system.
- Software interface – the language and codes that the applications use to communicate with each other and with the hardware.
- Hardware – the wires, plugs and sockets that hardware devices use to communicate with each other.

CHAPTER THREE

TYPES OF MEMORY

Types of Memory

1- Random Access Memory (RAM):

The main memory in a computer. It keeps system software, programs and data, which are need when the computer is working. It is a volatile memory.

RAM is divided into same sized locations that each of them has a unique address.

There are two type of RAM:

- Dynamic RAM (DRAM):
  DRAM chips must be recharged many times each second, or they will lose their contents.
- Static RAM (SRAM):
  SRAM does not need to be recharged and can hold its contents longer, faster than DRAM, but it is more expensive.

2- Read Only Memory (ROM):

Nonvolatile chips and cannot write on it. It content a set of startup instructions and also contents of system BIOS (Basic Input Output System).

There are three types of nonvolatile:

- **Programmable read - only memory (PROM)**:

Programmed only once for a specific program if got any change on this program will ignore and replaces, and also called one-time programmable ROM (OTP).

- **Erasable programmable read-only memory (EPROM):**

Is a special type of PROM, Programmed several times, can change and delete it by using special devices and ultraviolet rays to change information.

- **Electrically erasable programmable read-only memory (EEPROM):**

Is based on a similar structure to EPROM, but allows its entire contents to be erased, and then rewritten, so that they not need to be removed from the computer, and also called flash memory.

Cache Memory:

Moving data between RAM and the CPU's registers is very important operation depend on the time which the CPU perform, because RAM, is much slower than the CPU. A partial solution to this problem is to found cash memory, it is similar to RAM in their work, except that it is extremely fast compared to RAM.

Flash Memory:

Flash memory is a type of (EEPROM) is a non-volatile computer storage medium that can be erased and reprogrammed. Flash memory offers fast read access times. It features that it does not need to be defined or programs to run.

Virtual Memory:

If your computer lacks the random access memory (RAM) needed to run a program or operation, Windows uses virtual memory to compensate.

Virtual memory combines your computer's RAM with temporary space on your hard disk.

When RAM run slow, virtual memory moves data from RAM to a space called a **paging file**. Moving data to and from the paging file frees up RAM to complete its work.

The more RAM your computer has, the faster your programs will generally run. If a lack of RAM is slowing your computer, you might to increase virtual memory to compensate. However, your computer can read data from RAM much more quickly than from a hard disk, so adding RAM is a better solution.

CHAPTER FOUR

- Basic Input Output System (BIOS)
- Complementary Metal- Oxide-Semi-Conductor (CMOS)
- Data Storage

BIOS & CMOS

Basic Input Output System (BIOS):

Is a set of instructions that a Personal Computer (PC) uses to successfully start up. It is located on a chip on the motherboard inside of a computer.

One of the main functions of the BIOS is to give instructions for the Power-On Self Test (POST). This self test ensures that the computer has all of the necessary parts and functionality needed to successfully start itself, such as use of memory and a keyboard and other components.

If errors are detected during the test, the computer gives a code that reveals the problem. Error codes are typically presented as a series of beeps heard shortly after startup.

The BIOS also works to give the computer basic information about how to interact with some critical components, such as hard drives and memory, needed to load the Operating System (OS). The basic instructions have been loaded and the self-test has been passed, the computer can proceed with loading the OS from one of the attached drives.

Complementary Metal- Oxide- Semi-Conductor (CMOS)

A complementary metal oxide semiconductor (CMOS) is a type of integrated circuit technology. The term is often used to refer to a battery-powered chip found in many personal computers that holds some basic information, including the date and time and system configuration settings, password, needed by the basic input/output system (BIOS) to start the computer.

CMOS Chip:

Store the information needed by the BIOS, such as the size of disks and so on and need a battery to retain its contents.

BIOS chip:

Store BIOS system which return at the beginning of the computer the next time and do not need a battery to retain its contents.

Data Storage

Classified according to the access method:

- Sequential Access Media is a storage system where the data is stored and read in a fixed order. One example of this is video cassette.
- Random-access media is a form of computer data storage; allow stored data to be accessed directly in any random order such as hard disks, floppy disk, CD, and DVD.

1-Hard Disk:

Is a data storage device used for storing and retrieving digital information using rapidly rotating disks (platters) coated with magnetic material. HD retains its data even when powered off, it is found in the basic unit of compute (case) that is built –into the system unit.

HD consists of one or more rigid ("hard") rapidly rotating disks (platters) with magnetic heads arranged on a moving actuator arm to read and write data to the surfaces.

The hard disk platters spin at a high rate of speed, typically 5400 to 7200 revolutions per minute (RPM).

Storage capacities of hard disks for personal computers range from 10 GB to 120 GB (one billion bytes are called a gigabyte).

Hard Disc contains mechanical and electronic parts:

a- Mechanical parts:

- One Platter or several platters coated with magnetic material.
- Reading and writing heads.
- Arm holds the Reading and writing heads.

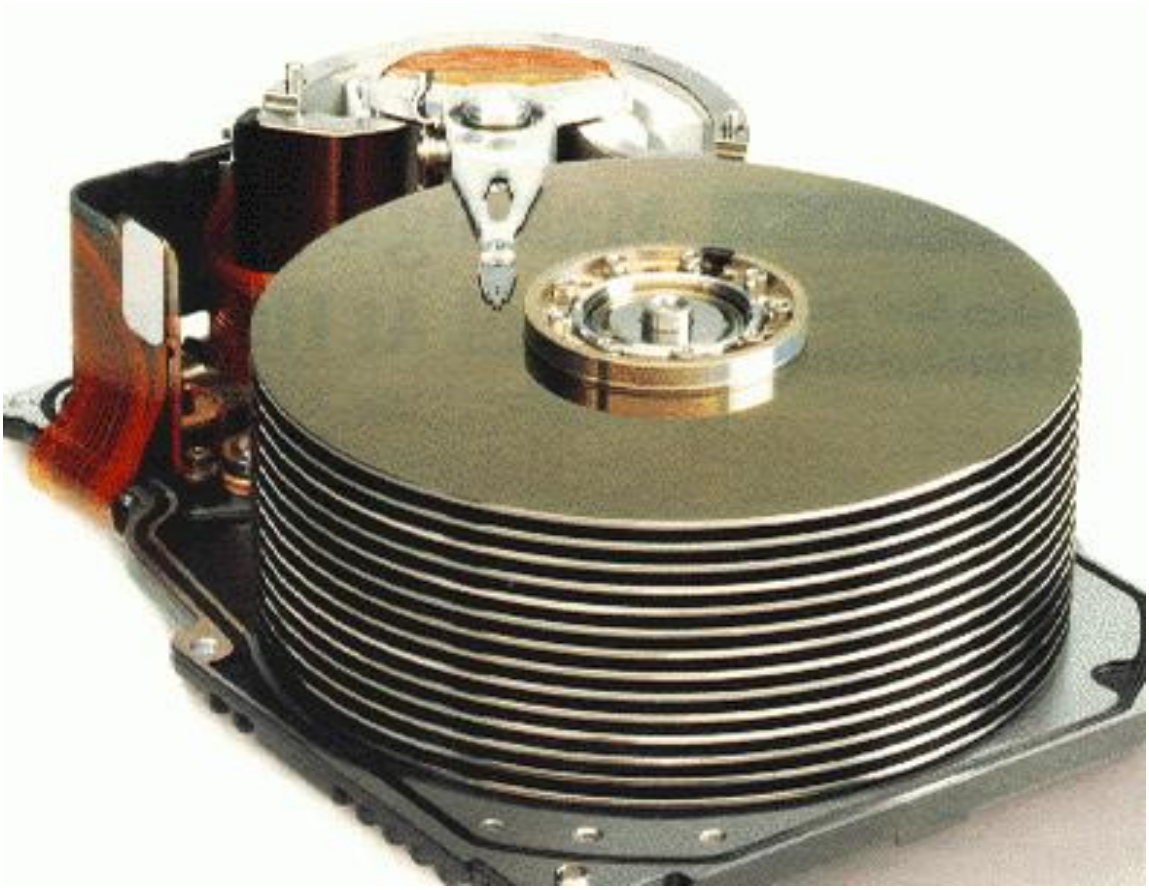
- Mechanical system to move the arm.
- Motor to rotate the platters.

b- Electronic Parts:

Is electronic panel on the down of hard disk, and responsibility of this section is

Control of reading and writing process on the hard disk and also control the motor which rotates platters.

Data storage on the platters, when an increases the number of platters, increases the storage Capacity of hard disk. The platters made of aluminum and the modern platter made from glass-reinforced ceramic which is the best performance to resistance of high temperatures. For each platter content two reading and writing heads, one for read and other for write which their location one of the down of platter and the other on the top of platter.



Hard Drive Platters

2-Compact Disc:

A compact disk (CD), also called an optical disc, is a flat round, portable storage. It is a small plastic disc used for the storage of digital data, covered with a transparent coating so that it can be read by a laser beam. It is popular because low cost and easy to use.

The capacity of a CD-ROM is 650 MB of data or about 70 minutes of audio.

A CD player has three major mechanical components:

a- Drive motor:

The drive motor rotates the disc between 200 and 500 revolutions per minute.

b- Laser and lens system:

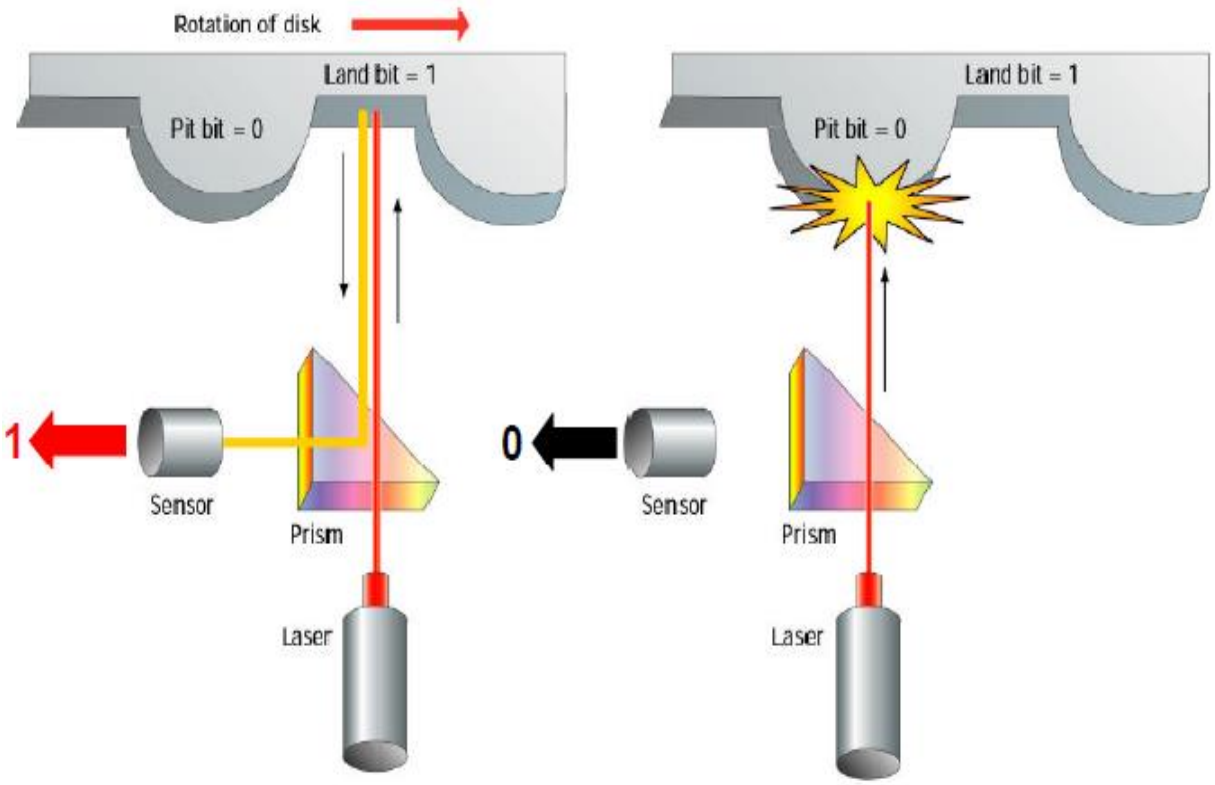
Reads the information using a laser beam.

c- **Tracking mechanism:**

Their function is to move the laser system so that the laser beam to trace the spiral path, and must be the accuracy of this system is very high so you can move the laser system.

The stored data on the disc is convert to the sets of data that can be handled, and then send it to Digital to Analogue Converter (DAC). The main function of the cylinder operator is the focus of the laser beam on the data path, when the laser

beam reaches to the cylinder passes through a layer of plastic and then reflected when it collides with a layer of aluminum. When reflected the laser beam 1, and when the laser beam not reflect 0.


Rotation of disk
Land bit = 1
Pit bit = 0
1
Sensor
Prism
Laser
0
Sensor
Land bit = 1
Pit bit = 0
Prism
Laser

CHAPTER FIVE

Input and Output Devices

Input and Output Devices

Input Devices

1-Keyboard*:*

The most commonly used input device is the keyboard, it ability to enter text and number. It has 101 keys arranged in five groups:

a- Alphanumeric keys

The alphanumeric keys –the part of the keyboard that looks like a typewriter arranged the same way on almost every keyboard. Sometimes this common arrangement is called the (QWERTY) because the first six keys on the top row of letters are Q, W, E, R, T and Y.

b- Modifier Keys

Besides the character keys, a keyboard incorporates special keys that do nothing by themselves but modify the functions of other keys. For example, the ⇧ Shift key can be used to alter the output of character keys, whereas the Ctrl (control) and Alt (alternate) keys trigger special operations when used in concert with other keys.

c- Numeric Key

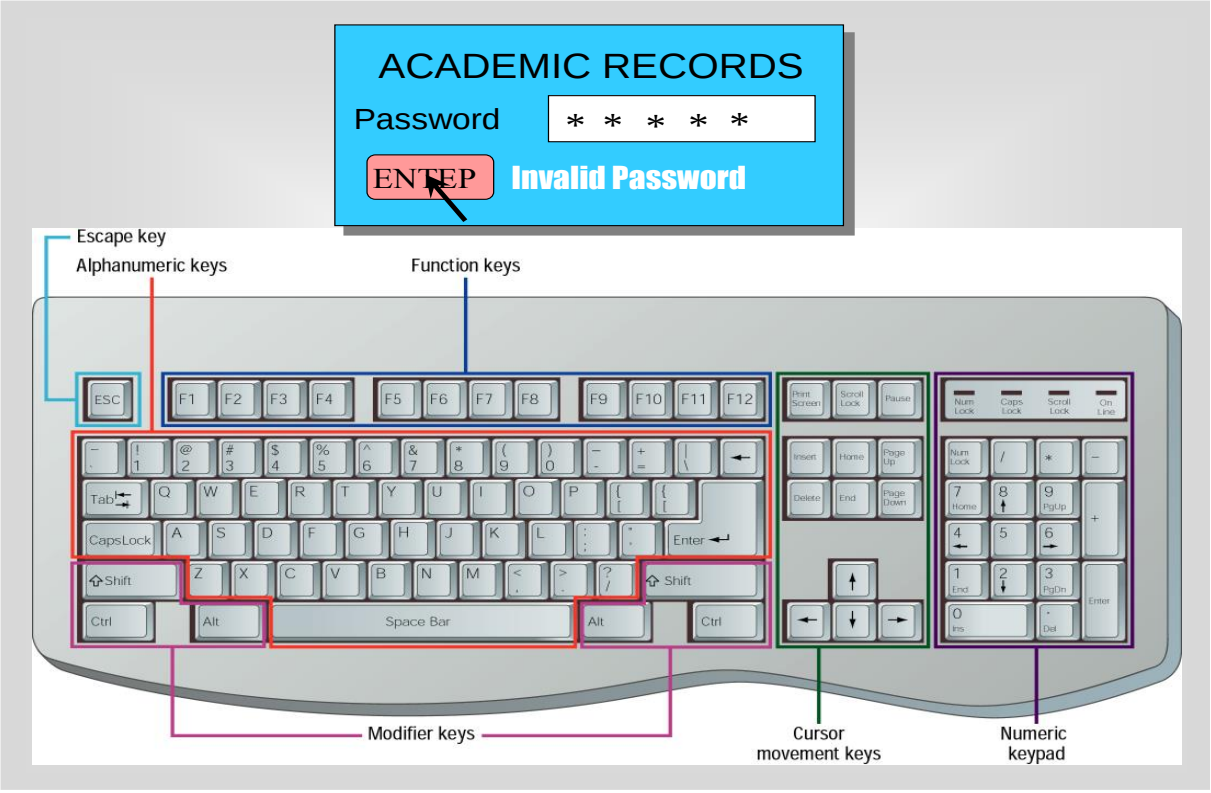
The Numeric Key located on the right of the keyboard, it is used for numeric data as well as mathematical operation (+, _, *, and /).

d- Function Key

The fourth parts of the keyboard, these keys (labeled F1, F2, and so on) are usually arranged in the top of keyboard. Each function key's purpose depends on the program you are using.

e- Cursor –Movement keys

The fifth part of the keyboard is the set of cursor –movement keys, which let you move around the screen without using a mouse such as → ↑ ← ↓ as well as print screen, home.



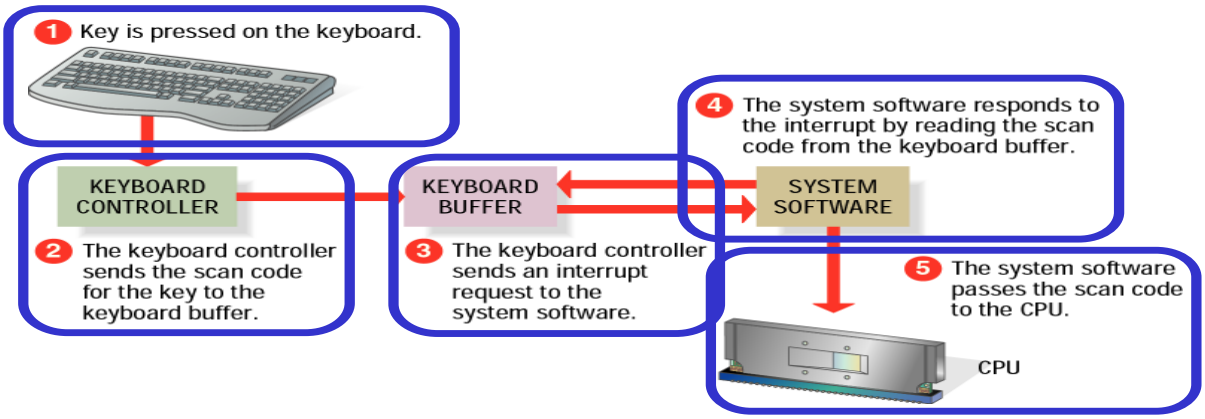
Parts of Keyboard Circuitry:

a- Keyboard controller b- Scan code

C- Keyboard buffer d- Interrupt request

How the Computer Accepts Input from the Keyboard

A tiny chip called the Key board controller notes the key has been pressed the Keyboard controller places a cod in to part of memory called the keyboard buffer, this code is called the key's Scan code, the Keyboard controller then send signals to the computer system software that something has happened at the key signal called interrupt request for read the code of the keyboard. When the system software receives an interrupt request to reads the code from the keyboard buffer .It then sends the key's scan code to the CPU.



How the Computer Accepts Input from the Keyboard

Touch Screens

- Touch-screen systems accept input directly through the monitor.
- Touch screens use sensors to detect the touch of a finger.

A touch screen has three main components:

1- touch screen sensor

A touch screen is the glass panel which generates an electrical current or signal accurately in the place where it presses directly.

2- touch screen controller

It takes information from the touch sensor and translates it into processor by cable USB**.**

3- software driver

It used for processing and run the screen.

Types of Touch Screen:

- *Resistive touch screen:*

Resistive-type screens the most common in use today , they tend to be very durable and can be used in a variety of environments.

- *Capacitive Touch Screen:*

It is charge storing screen that uses electric charges where stored by hand at the pressure, capacitive panels must be touched with a finger.

- *Surface Acoustic Wave Screen **(SAW):***

Surface wave touch panels are the more advanced of the two types, offering the highest clarity.

Mouse:

Is a pointing device that rolls around on a flat surface which is used to control the movement of a mouse pointer on the screen and also called mouse pointer that is used to select text, access menus, and interact with programs, files or data that appear on the screen.

Types of Mouse:

1- Mechanical mouse.
2- Opt. mechanical mouse.
3- Optical mouse.

Component of Opt. mechanical mouse:

- Container mouse.
- Function Buttons.
- Interconnect Cable.
- Internal Sensors.
- Tracking ball.
- Light Emitting Diode (LED).

How the Mouse work?

Mouse has the two units to generate light in each unit has light emitting diode and transistor to sensor light, the emitter is located on one of the aspects of wheel that sends light through the holes and when the wheel round, cutting the light passing through the holes and oriented sensor light on the other side of wheel, which receives light and converts it to electrical signals in the form of flashes. The computer processes these flashes to know the amount and direction movement of tracking ball in a mouse.

Output Devices (Monitors)

- *Cathode Ray Tube (CRT):*

CRT is the technology used in a computer monitors and televisions, works by moving an electron beam back and forth across the back of the screen. Each time the beam makes a pass across the screen, it lights up phosphor dots on the inside of the glass tube, thereby illuminating of the screen. The smallest number of phosphor dots (pixel). Common sizes are 15, 17, 19, 21, and 22 inches.

Resolution:

Is the term used to describe the number of dots, or pixels, used to display an image. Higher resolutions mean that more pixels are used to create the image, resulting in a cleaner image. The display, or resolution on a monitor, is composed of thousands of pixels or dots. This display is indicated by a number combination, such as 800 x 600. This indicates that there are 800 dots horizontally across the monitor, by 600 lines of dots vertically.

Compared between CRT and Liquid Crystal Displays (LCDs):

1- Cathode Ray Tubes (CRT) They are large, bulky and consume a lot of power**.** Liquid Crystal Displays more commonly known as (LCDs) to replace CRTs in most applications today, Consume screens (LCD) less energy compare to (C RT).

2- the important features in the LCD monitors is the viewing area, for example, screen size of 15 inches type LCD equals in the viewing area of a 17-inch screen of CRT.

3- Characterized of screens (LCD) that its image is sharp and accurate in addition to the clarity of colors compared to CRT.

Printers:

Device used to print the data or text on the pages, the output printed of computer called hard copy.

The printers different depending on:

1- The printer color (color or black).
2- Technique type (Thermal, Ink-Jet and Laser printers).
3- Printer accuracy.
4- The functions.
5- Printer speed.

There are two types of printers Depending on the work nature:

1- **Impact printers:**

Creates symbols by using small hammers or pins, by pressing on the tape against the paper. This type is limited speed, and the shape and size of letters are fixed.

2- Nonimpact Printers:

Creates symbols by heating effect or electrical effect and laser beam that prints more thousands of line per seconds.

There are many types of printers:

1- Thermal printers.

2- Ink-Jet printers.

3- Laser printers.

CHAPTER SIX

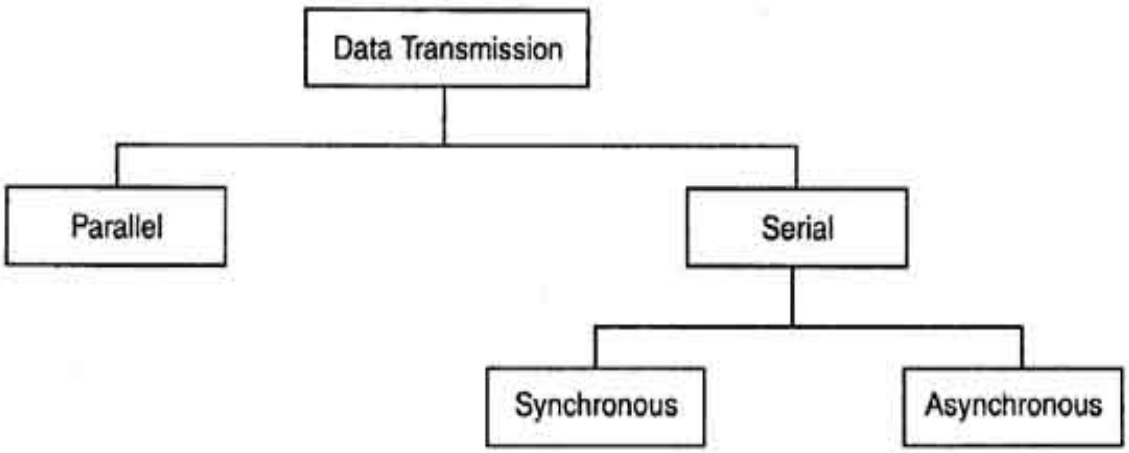
Data transmission

Definition Data transmission:

Data transmission is the transfer of data from point-to-point often represented as an electro-magnetic signal over a physical point-to-point or point-to-multipoint communication channel. Examples of such channels are copper wires, optical fibers, wireless communication channels, and storage media.

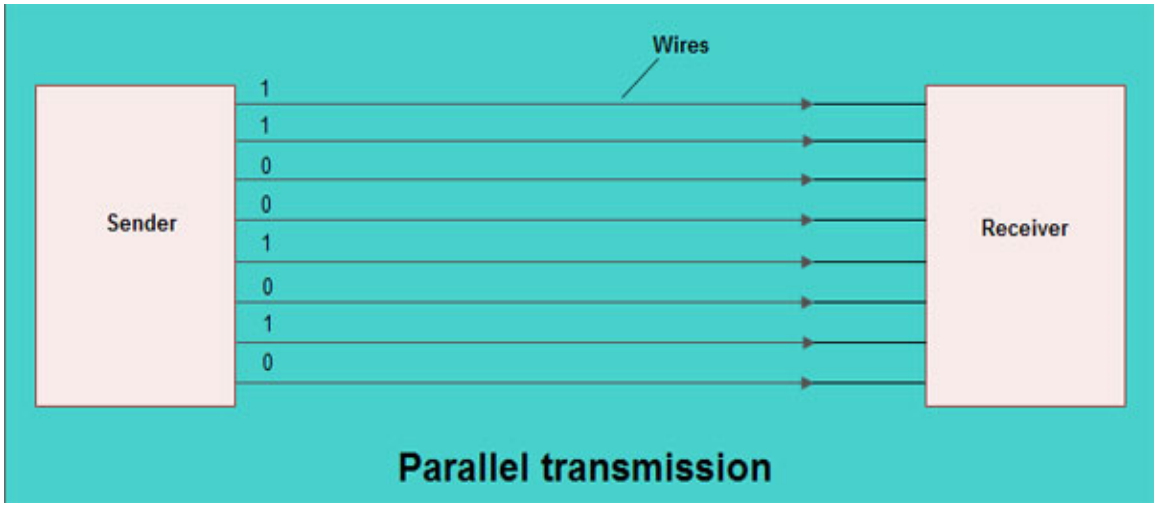
Data transmission is a subset of the field of data communication, which also includes computer networking or computer communication applications and networking protocols, for example routing. The term input refer, to data or software or instruction that you enter into the computer memory.

Types of data transmission:



1. Parallel transmission:

Within a computing or communication device, the distances between different subunits are too short. Thus, it is normal practice to transfer data between subunits using a separate wire to carry each bit of data. And data is exchanged using a parallel transfer mode. This mode of operation results in minimal delays in transferring each word. As shown in the fig. eight separate wires are used to transmit 8 bit data from sender to receiver.

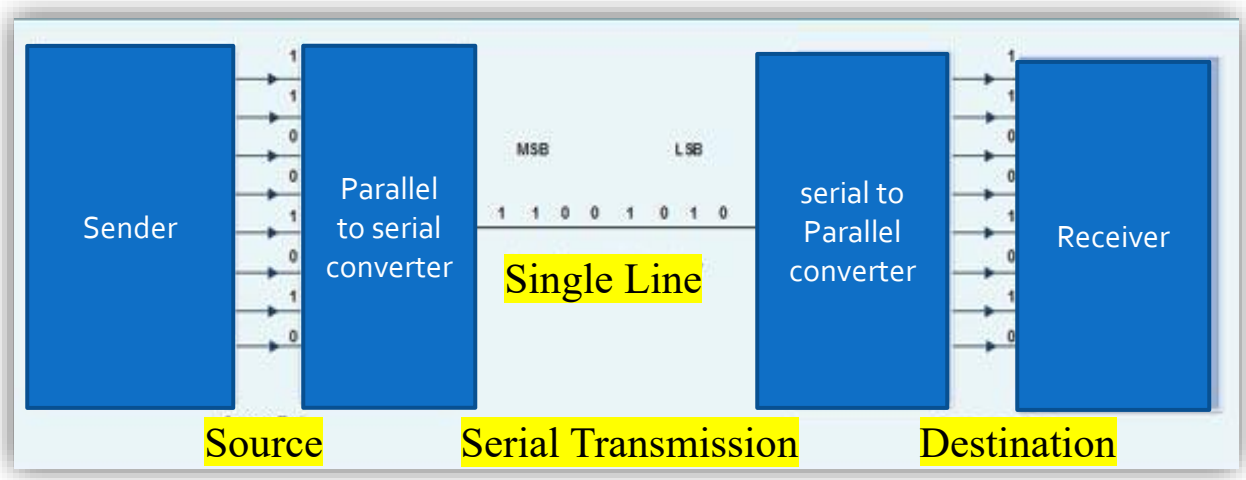


Parallel transmission is speedy way of transmitting data as multiple bits are transmitted simultaneously with a single clock pulse. But it is costly method of data transmission as it requires n lines to transmit n bits at the same time.

2. Serial Transmission:

When transferring data between two physically separate devices, especially if the separation is more than a few kilometers, for reasons of cost, it is more economical to use a single pair of lines. Data is transmitted as a single bit at a time using a fixed time interval for each bit. This mode of transmission is known as bit-serial transmission. The internal circuitry of computer transmits data in parallel fashion. So in order to change this parallel data into serial data, conversion devices are used.

These conversion devices convert the parallel data into serial data at the sender side so that it can be transmitted over single line. On receiver side, serial data received is again converted to parallel form so that the internal circuitry of computer can accept it.



Serial transmission is use a single communication line reduces the transmission line cost as compared to parallel transmission. But it use conversion devices at source and destination end may lead to increase in overall transmission cost. And this method is slower as compared to parallel transmission.

Types of Serial Transmission:

There are two types of serial transmission-synchronous and asynchronous both these transmissions use 'Bit Synchronization'

Bit Synchronization is a function that is required to determine when the beginning and end of the data transmission occurs. Therefore bit synchronization provides timing control.

1 – Asynchronous:

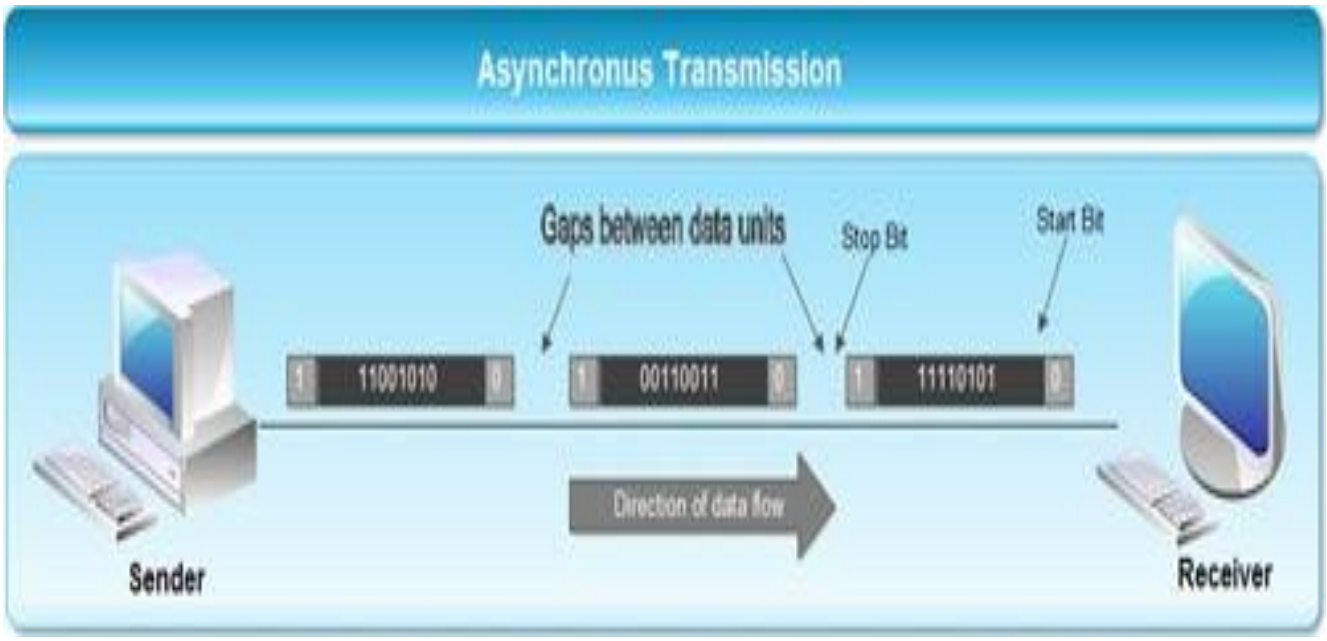
It sends only one character at a time where a character is either a letter of the alphabet or number or control character i.e. it sends one byte of data at a time.

Bit Synchronization between two devices is made possible using start bit and stop bit.

Start bit indicates the beginning of data i.e. alerts the receiver to the arrival of new group of bits.

A start bit usually 0 is added to the beginning of each byte. Stop bit indicates the end of data i.e. to let the receiver know that byte is finished, one or more additional bits are appended to the end of the byte. These bits, usually 1s are called stop bits.

The gap or idle time can be of varying intervals. This mechanism is called Asynchronous, because at byte level sender and receiver need not to be synchronized. But within each byte, receiver must be synchronized with the incoming bit stream.



2 – Synchronous:

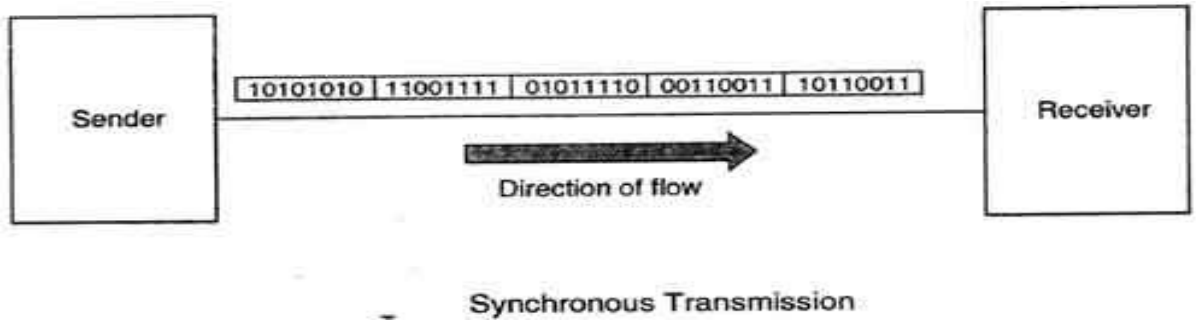
Synchronous transmission does not use start and stop bits. In this method bit stream is combined into longer frames that may contain multiple bytes.

There is no gap between the various bytes in the data stream.

In the absence of start & stop bits, bit synchronization is established between sender & receiver by 'timing' the transmission of each bit.

Since the various bytes are placed on the link without any gap, it is the responsibility of receiver to separate the bit stream into bytes so as to reconstruct the original information.

In order to receive the data error free, the receiver and sender operates at the same clock frequency.



Synchronous Transmission

Synchronous transmission is used for high speed communication between computers.

Compared between Synchronous and Asynchronous transmission:

1- Synchronous transmission method is faster as compared to asynchronous as there are no extra bits (start bit & stop bit) and also there is no gap between the individual data bytes.
2- But it is costly as compared to asynchronous method. It requires local buffer storage at the two ends of line to assemble blocks and it also requires accurately synchronized clocks at both ends. This lead to increase in the cost.
3- And the sender and receiver have to operate at the same clock frequency. This requires proper synchronization which makes the system complicated.

Sr. No.	Factor	Asynchronous	Synchronus
1.	Data send at one time	Usually 1 byte	Multiple bytes
2.	Start and Stop bit	Used	Not used
3.	Gap between Data units	Present	Not present
4.	Data transmission speed	Slow	Fast
5.	Cost	Low	High

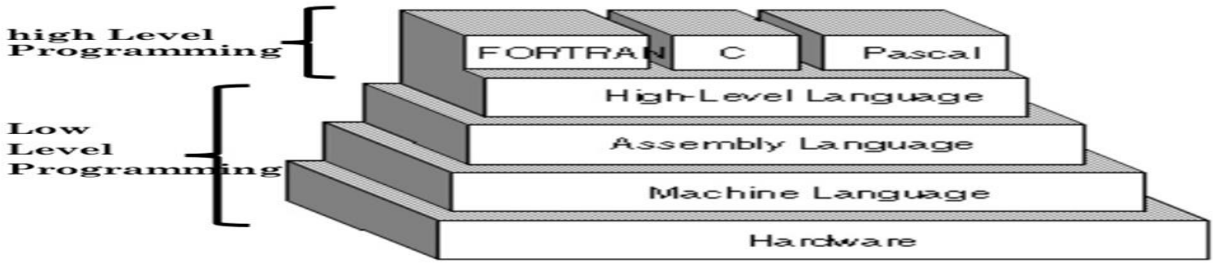
CHAPTER SEVEN

Programming Language

Programming languages:

Programming languages provide the basic building blocks for all systems and application software.

Programming languages allow people to tell computers what to do and are the means by which software systems are developed, we will describe the two-levels of programming languages:



Low level languages:

1 – Machine language:

Is the lowest-level computer language, consisting of the internal representation of the instructions and data. This machine code-the actual instructions understood and directly executable by the CPU is composed of binary digits. Machine language is the only programming language that the machine actually understands, therefore, machine language

is considered the first-generation language. All other languages must be translated into machine language before the computer can run the instructions

because computer's CPU is capable of executing only machine language programs.

Machine language is extremely difficult to understand and use by programmers. As a result, increasingly more user-friendly languages have been developed.

These user oriented languages make it much easier for people to program. But they are impossible for the computer to execute without first translating the program into machine language.

The set of instructions written in a user-oriented language is called a source program.

The set of instructions produced after translation into machine language is called the object program.

Programming in a higher-level language (i.e., a user oriented language) is easier and less time consuming but additional processor time is required to translate the program before it can be executed.

2 – Assembly language:

Assembly languages are considered second-generation languages it is more user-friendly because it represents machine language instructions and data locations in primary storage by using mnemonics, which people can more easily use.

Compared to machine language, assembly language eases the job of the programmers.

Translating an assembly language program into machine language is accomplished by system software program called an assembler.

High level languages:

1 – Procedural languages:

Called third-generation language

• Procedural language are much closer to natural language (the way we talk) and therefore, are easier to write, read.

• Procedural language use common words rather than abbreviated mnemonics.

• There are three examples of procedural languages FORTRAN, COBOL, and C.

2 – Nonprocedural languages:

- Called fourth-generation language.
- They can be used by non-technical users to carry out specific functional tasks.
- These languages simplify the programming process as well as reduce the number of coding errors.
- They are common in database applications as query languages, report generators.

Natural languages:

- Are called fifth –generation languages or" intelligent language".
- They are use mnemonics and tables.
- Most of these languages are still experimental because the programs that are translate natural language into machine–readable form are extremely complex and require a large amount of computer resources.

Newer programming languages:

1 – Visual programming languages:

- Are used within graphical environment.
- Are using a mouse, icons, and symbols on screen.
- Visual basic and visual C++ are examples of visual programming languages.

2 – Hypertext markup language (HTML):

- Is an approach to data management in which data are stored in a network of nodes connected by links(called hyperlinks).
- Users can access data through an interactive browsing system.
- The combination of nodes, links, and supporting indexes for any particular topic is a hypertext document.
- A hypertext document may contain text, images, and other types of information such as data files, audio, and video.
- World Wide Web (www) uses HTML for creating and recognizing hypertext document.

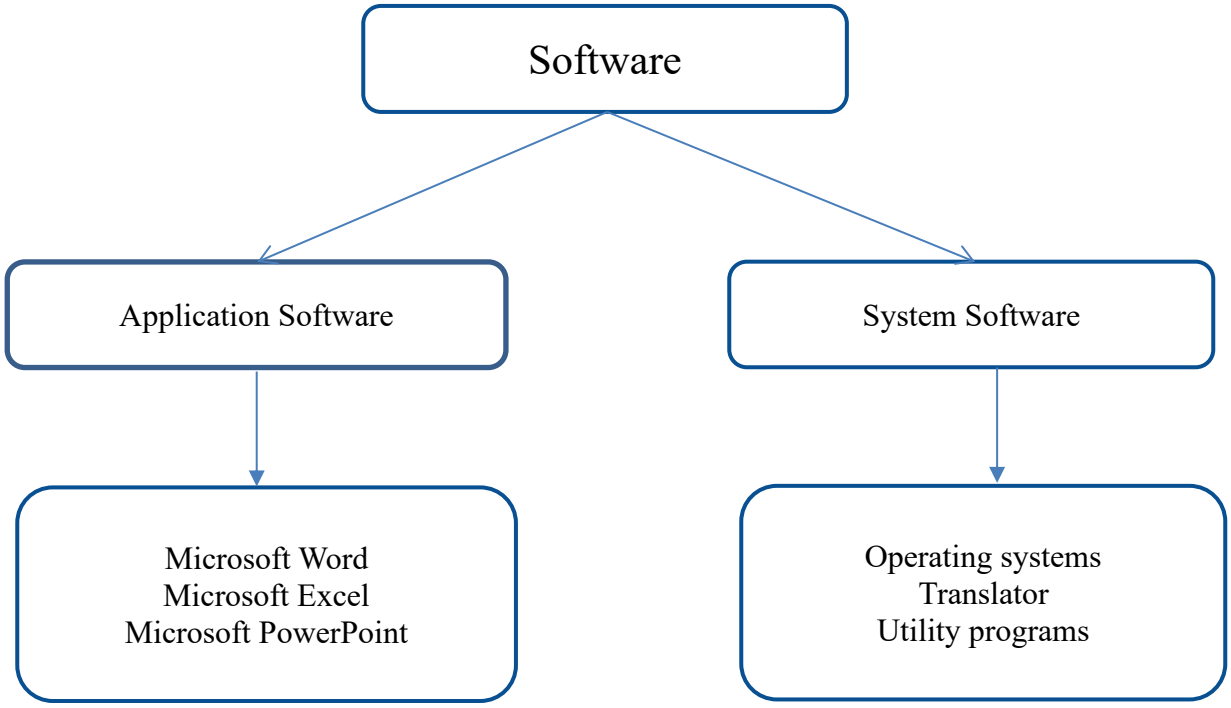
3 – Object-Oriented programming languages:

- Object-Oriented Programming (OOP) languages are based on the idea of taking a small amount of data and instructions about what to do with that data and putting both of them together into what is called an object.

- C++ and JAVA are examples of OOP languages.

CHAPTER EIGHT

Software - Translators



Application Software:

Programs that help the users to execution a specific task such as Microsoft Word and Microsoft Excel.

System Software

Is a set of programs that manage the resources of a computer system. System Software is a collection of system programs that perform a variety of functions.

Types of systems software:

A. Operating systems.

B. Translators.

C. Utility programs.

A. Operating systems

An operating system, also called an OS, is the important type of system software, which is designed for your computer system to ensure your system is working together smoothly and efficiently.

The operating system (OS) is the first thing loaded onto the computer, without the operating system, a computer is useless.

The primary functions of an operating system are:

1- Loads programs (such as word processing and spreadsheet programs, or game software) into the computer's memory that you can use them.

2- Provide a user interface.

3- Manage system memory.

4- Manage processing tasks.

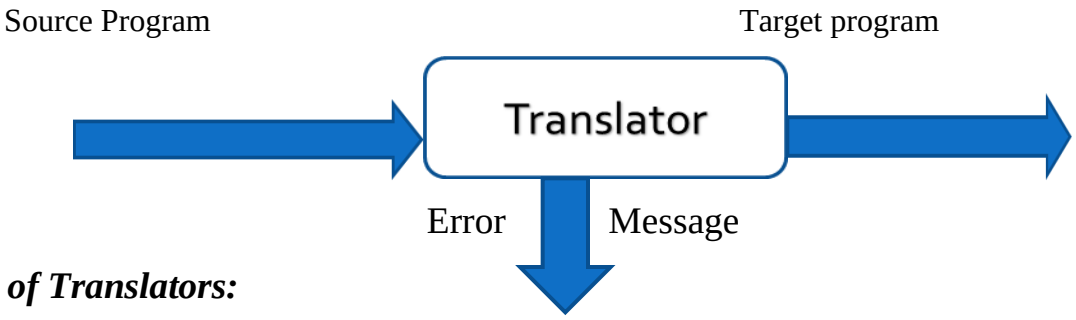
5- Coordinates how programs work with the computer's hardware and other software.

6- Provides ways to manage and organize the way information is stored on and retrieved from disks.

B. Translators

Is a computer program that converts a program written in high language such as Pascal or C++, into a machine language that can be directly executed by the computer.

The Program that written in high language called (Source program) and the program that convert to machine language called (target program or object program).



Types of Translators:

1 – Assemblers:

A computer program which translates from assembly language to machine language

2 – Compilers:

Is a special software program that converts source code into machine language. An important part of any compiler is the detection and reporting of errors.

- **Linker:**

Is a program that takes one or more objects generated by compilers into a single executable program.

- **Loader:**

Is the part of an operating system that is responsible for loading programs to memory, preparing them for execution.

The Phases of a Compiler:

1- Lexical Analysis.

2- Syntax Analysis.

3- Semantic Analysis.

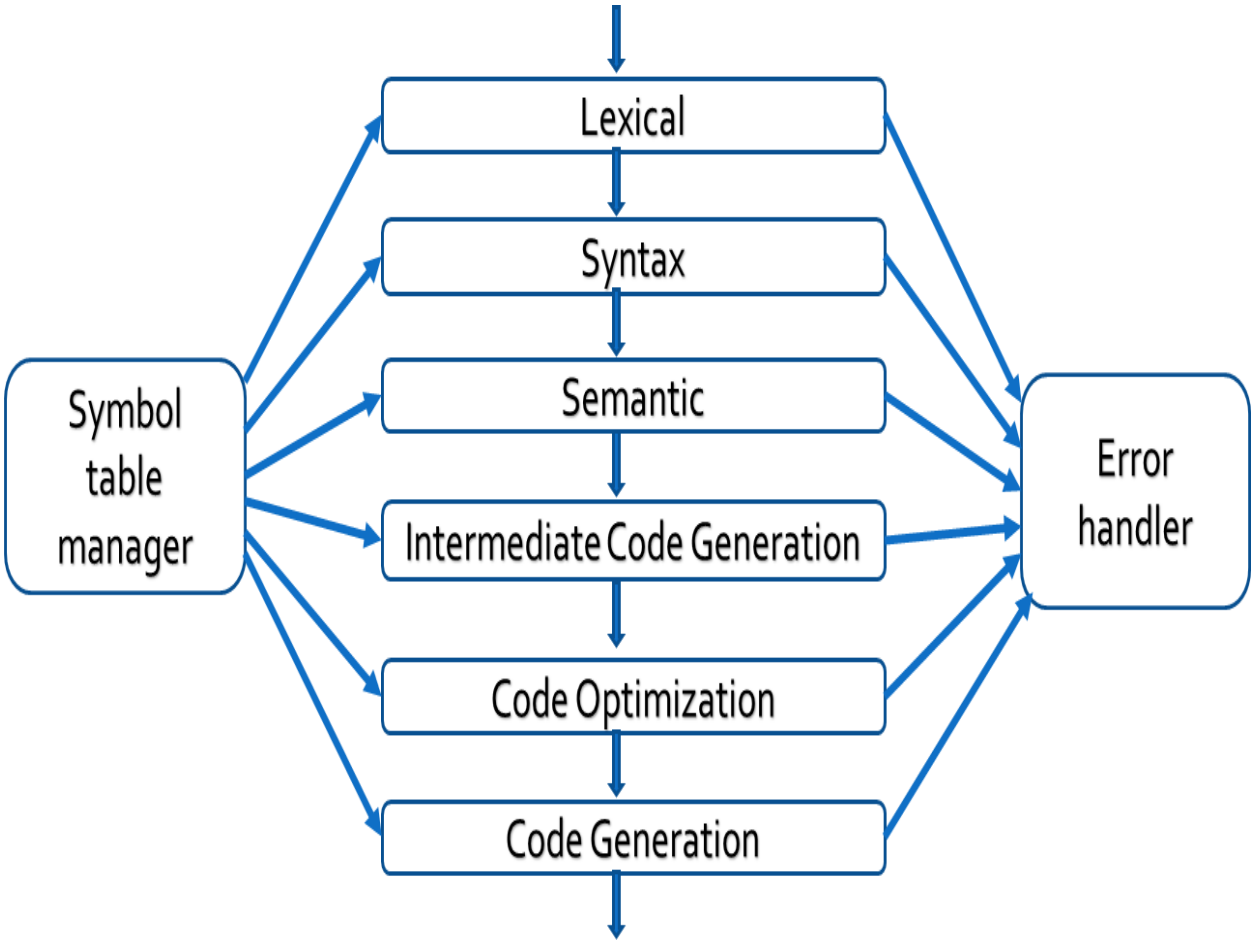
4- Intermediate Code Generation.

5- Code Optimization.

6- Code Generation

In each phase we need variables, which are taken from a table called (Symbol table manager) and in each phase may generate some errors so it must have a program to process these errors called (Error handler). Each stage in the compiler has two inputs and outputs, For example the first phase (lexical analysis) the first input is the source program, while the second input it is some of the variables that you need at that stage. The first output it is the errors that may generate and process in program called error handler, while the second output is the input for the next phase (syntax analysis).

Source



3 – Interpreters:

Is a program that translates a program line-by-line (statement-by-statement) and carries out the specified actions in sequence and executes the statement immediately before going on to translate the next statement.

C – Utility programs:

Programs that perform a specific task related to the management of computer functions, as password protection, memory management, virus protection, and file compression.

CHAPTER NINE

Software Model of the 8088\8086

The 8086 Microprocessor:

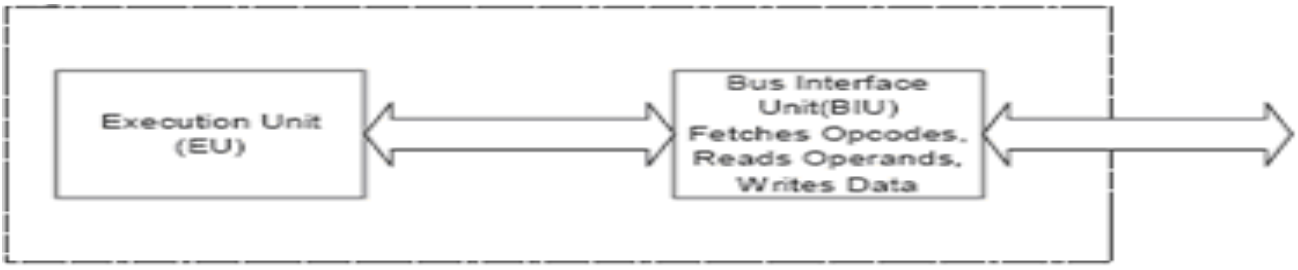
The 8086 have 16 bits internal and external data paths and a 20 bits address bus that can address up to l MB of memory.

The Internal Architecture of the 8086 MP:

The processor is partitioned into two logical units: an **Execution Unit** (EU) and **Bus Interface Unit** (BIU). The role of the EU is to execute instruction, whereas the BIU delivers instruction and data to EU.

The EU contains **ALU, CU** and number of registers. This feature enables the EU to execute instructions and perform arithmetic and logical operations.

The most important function of BIU is to manage the bus control unit, segment registers, and instruction queue. The BIU controls the busses that transfer data to the EU, to memory and to external input/output devices, whereas the segment registers control the memory addressing.



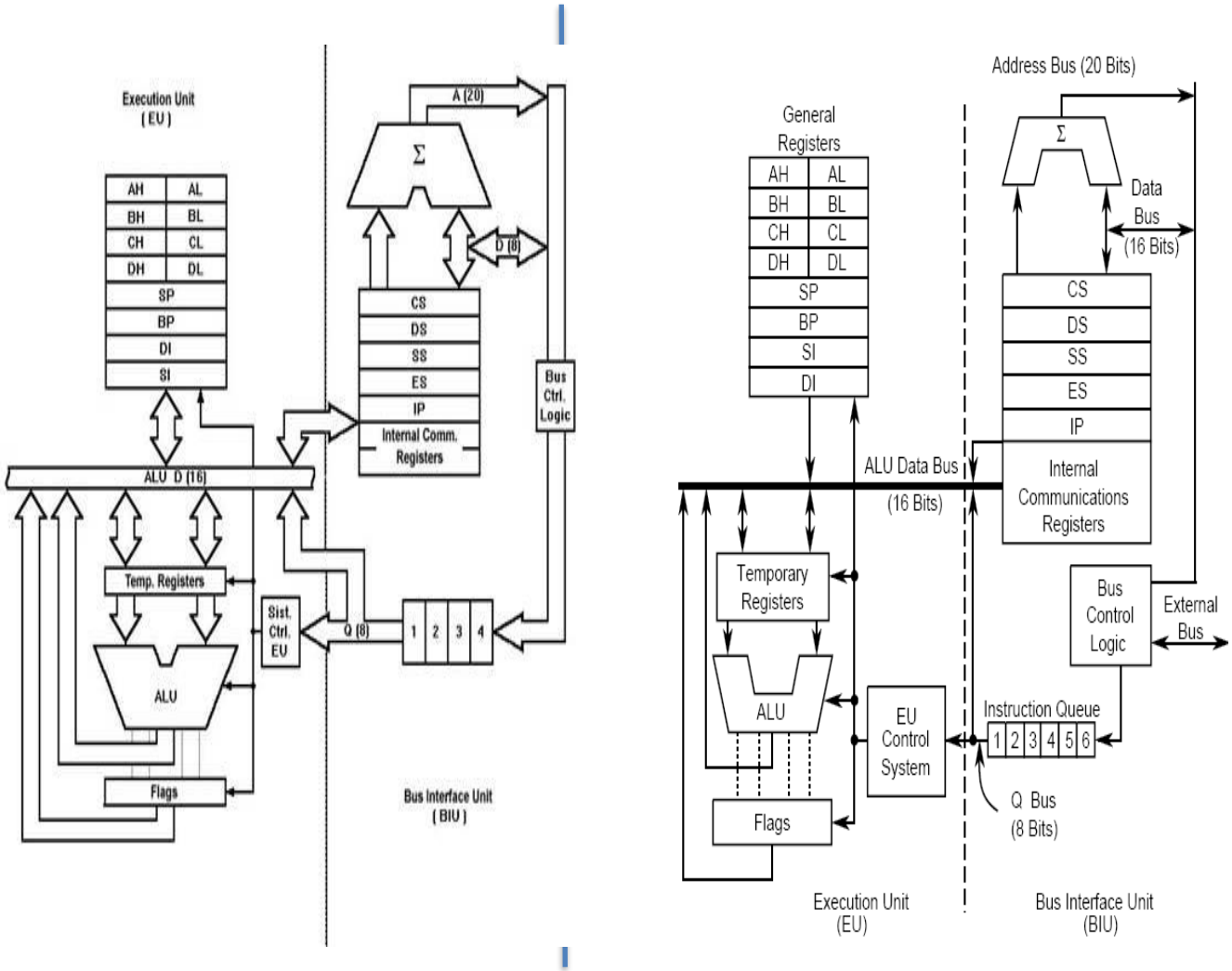
Another function of the BIU is to provide access to instructions. The instructions for a program which executing are kept in memory, the BIU must access it from memory and place them in an instruction queue, which varies in size depending on the processor (the size of instruction queue for 8086 MP is 6 bytes).

This feature enables the BIU to look ahead and pre-fetch instructions, so that there is always a queue of instructions ready to execute.

The EU and BIU work in parallel, with the BIU keeping one step ahead The EU notifies the BIU when it needs access to data in memory or I/O devices.

Also the EU request machine code instructions from the BIU instruction queue. The top instruction is the currently executable one, and while the EU is occupied executing an instruction, the BIU fetch another instruction from memory.

This fetching overlaps with execution and speeds up processing.



The only difference between an 8088 microprocessor and an 8086 microprocessor is the BIU.

In the 8088, the BIU data bus path is 8 bits wide versus the 8086's 16-bit data bus.

Another difference is that the 8088 instruction queue is four bytes long instead of six. The important point to note, however, is that because the EU is the same for each processor, the programming instructions are exactly the same for each. Programs written for the 8086 can be run on the 8088 without any changes.

Registers 8088\8086:

General Purpose Registers:

The 8086 microprocessor has a total of *fourteen* registers that are accessible to the programmer.

Eight of the registers are known as *general purpose registers* i.e. they can be used by the programmer for data manipulation.

Each of the registers is 16 bits long i.e. can contain a 16-bit binary number.

The first four registers are sometimes referred to as *data registers*. They are the AX, BX, CX, and DX registers.

The second four are referred to as *index/pointer registers* or *Offset Address Registers*. They are the SP, BP, SI and DI registers.

The data registers can be treated as 16-bit registers or they can each be treated as two 8-bit register.

As a 16-bit register, a general purpose registers name ends in an "X“ e.g. AX. As an 8-bit register, its name ends in either an "H", if it is the "high order" half of the 16-bits, or an "L", if it is the "low order" half; e.g. AH or Al, each 8-bit register can be used independently.

AX: *Accumulator* are registers that can be used for arithmetic, logical, shift, rotate, or other similar operations.

BX: *Base register* the only general purpose register whose contents can be used by instructions as a 16-bit offset address. This offset is paired by default with segment register "DS:", i.e. the memory reference [BX] actually means[DS:BX].

CX: ***Count register*** used by several instructions to control the number of times a looping process is performed.

DX: ***Data register*** often used to bold single-byte character data and referenced as DH or DL; also combines with AX to form a 32-bit register for some operations (e.g. MULtiply).

SI: S***ource index register:*** Offset address relative to DS.

DI: ***Destination index register:*** Offset address relative to ES.

BP: ***Base pointer:*** Offset address relative to SS.

SP: ***Stack pointer:*** A very important register. Maintains the program stack, and so should be used carefully, Offset address relative to SS.

Special Purpose Registers:

There are two special purpose registers on the 8086 the *instruction painter* (IP) and the *flag register*. The IP is sometimes referred to as the pc (*program counter*). These registers cannot be accessed directly; rather they are modified by the CPU during execution.

IP or PC - (Instruction Pointer Register) or (program counter):

This is a crucially important register which is used to control which instruction the CPU executes. The IP, or program counter, is used to store the memory location of the next instruction to be executed. The CPU checks the program counter to ascertain which instruction to carry out next. It then updates the program counter to point to the next instruction. Thus the program counter will always point to the next instruction to be executed. This address is the offset within the code segment. specified by the CS register.

Status (Flags) Register:

Flags register determines the current state of the processor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program.

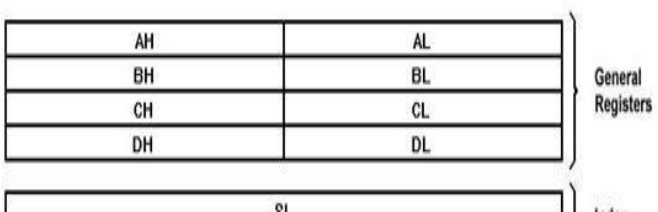
Nine individual bits of the status register are used as control flags (3 of them) and status flags (6 of them). The remaining 7 are not used. For most practical purposes, only 4 flags are used: zero, carry, sign and overflow.

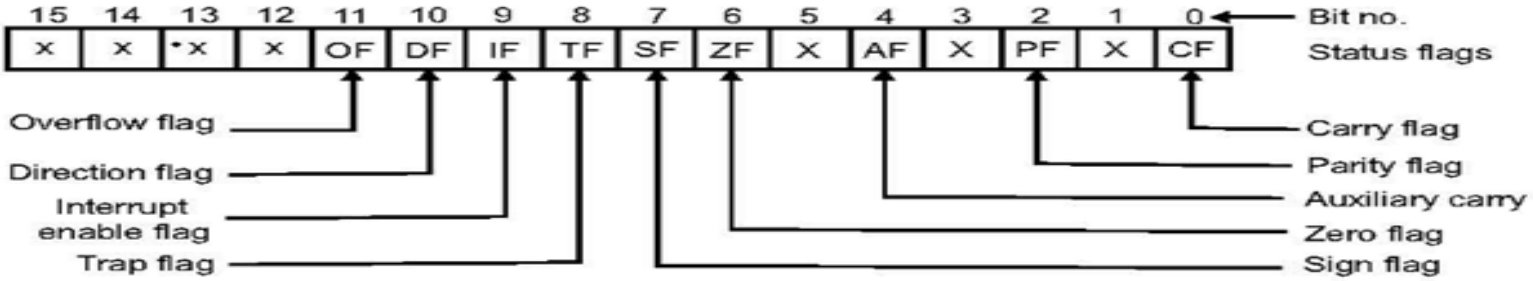
A flag can only take on the values o and 1. We say a flag is set if it has the value 1.

1. **Carry Flag (CF)** - this flag is set to 1 when there is an unsigned overflow. For example when you add bytes ***255+1*** (***result is not in range 0…255***). When there is no overflow this flag is set to0.
2. **Parity Flag (PF)** - this flag is set to 1 when there is even number of one bits in result, and to 0 when there is odd number of one bits.
3. **Auxiliary Flag (AF)** - set to 1 when there is an unsigned overflow for low nibble (4 bits).
4. **Zero Flag (ZF)** - set to 1 when result is zero. For non-zero result this flag is set to 0.
5. **Sign Flag (SF)** - set to 1 when result is negative. When is positive it is set to 0. (This flag takes the value of the most significant bit.)
6. **Trap Flag (TF)**
7. ***Interrupt enable Flag (IF)***
8. ***Direction Flag(DF)***
9. ***Overflow Flag (OF)***

Registers 8088\8086:

Registers 8088\8086:




15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit no.
X X X X OF DF IF TF SF ZF X AF X PF X CF
Status flags
Overflow flag
Direction flag
Interrupt enable flag
Trap flag
Carry flag
Parity flag
Auxiliary carry
Zero flag
Sign flag

CHAPTER TEN

Memory Address Space & Data Organization

OUTLINE

- **Memory Address Space & Data Organization**
  - Memory Segmentation.
  - Segment Registers.
  - Generating a Memory Address Space (Logical & physical Address).

Memory Segmentation:

The total memory size is divided into segments of various sizes. A segment is just an area in memory; the process of dividing memory this way is called ***Segmentation.*** In memory, data is stored as byte; "Byte-addressable" means that each byte has its own unique address.

Intel 8086 has 20 lines address bus, with 20 address lines, the memory that can be addressed is

(2^20) bytes = 1,048,576 = (1 MB)

Addresses always start at zero and go up in steps of one; so every number from zero up to 1"Meg" is an identifier or "address" of unique 8-bit storage location. Each byte has a specific address, 8086 can access memory with address ranging from 0000H to FFFFH.

Segment Registers:

In 8086, memory has four different types of segments are:

1- Code Segment

2- Data Segment

3- Stack Segment

4- Extra Segment

Each of these segments are addressed by an address stored in corresponding segment register.

These registers are 16-bit in size.

Each register stores the base address (starting address) of the corresponding segment. Because the segment registers cannot store 20 bits, they only store the upper 16 bits.

Generating a Memory Address Space (Logical & physical Address):

How is a 20-bit address obtained if there are only 16-bit registers?

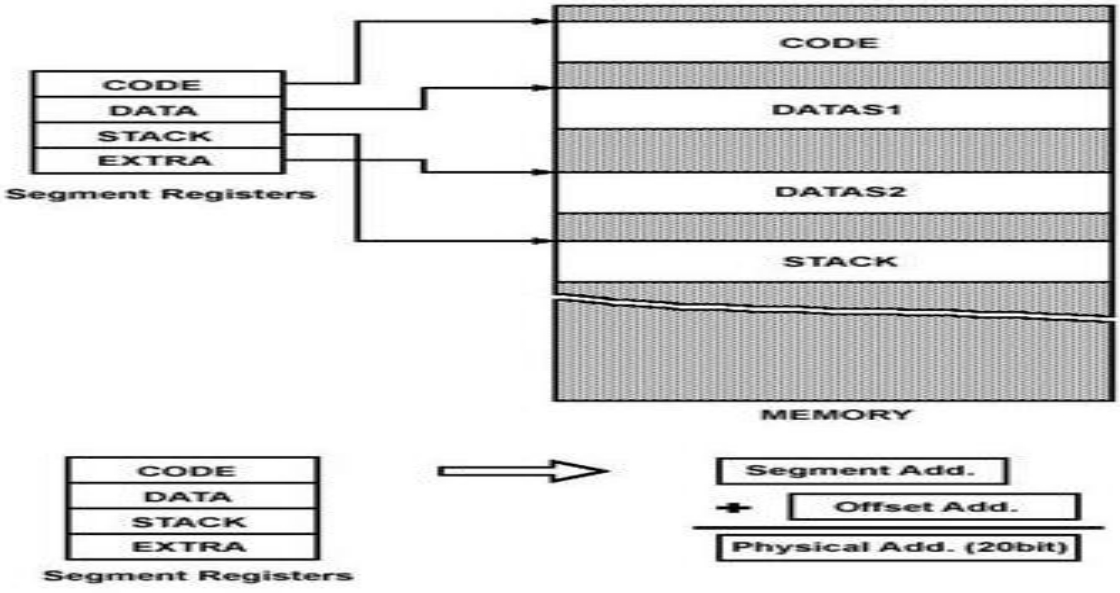
The 20-bit address of a byte is called its ***Physical Address***.

But, it is specified as a ***Logical Address.***

Logical address is in the form of:

Base Address: Offset

Offset is the displacement of the memory location from the starting location of the segment.



Example 1:

The value of Data Segment Register (DS) is 2222H.

To convert this 16-bit address into 20-bit, the BIU appends 0H to the LSBs of the address. After appending, the starting address of the Data segment becomes 22220H.

Example 2:

If the data at any location has a logical address specified as:

2222H : 0016H

Then, the number 0016H is the offset.

2222H is the value of DS.

Example 3:

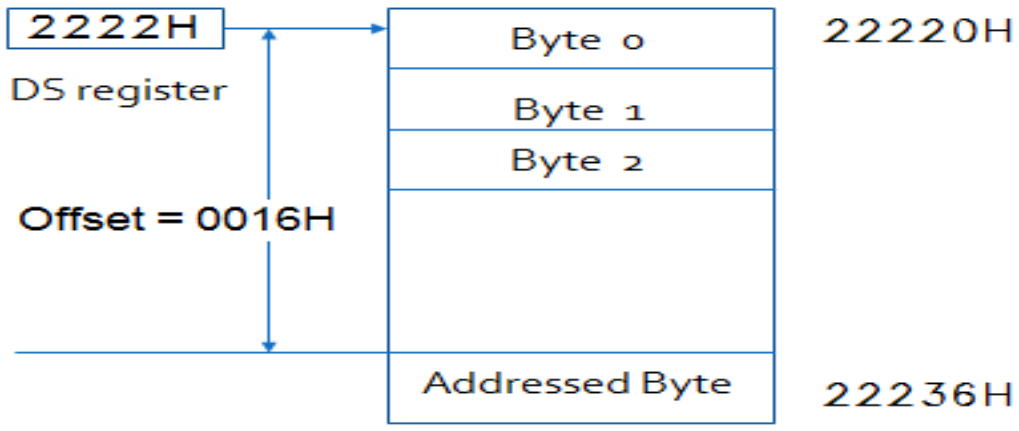
To calculate the effective address of the memory, BIU uses the following formula:

Effective Address = Starting Address of Segment*10 + Offset

To find the starting address of the segment, BIU appends the contents of Segment Register with 0H. then, it adds offset to it.

Therefore:

EA=22220H
0016H+
22236H



All offsets are limited to 16-bits. It means that the maximum size possible for segment is

2^16 = 65,535 bytes(64 KB).

The offset of the first location within the segment is 0000H.

The offset of the last location in the segment is FFFFH.

Segment	Offset Register	Function
CS	IP	Address of the next instruction
DS	BX, DI, SI	Address of data
SS	SP, BP	Address in the stack
ES	BX, DI, SI	Address of destination data (for string operations)

Question:

The contents of the following registers are:

CS = 1111H

DS = 3333H

SS = 2526H

IP = 1232H

SP = 1100H

DI = 0020H

Calculate the corresponding physical addresses for the address bytes in CS, DS and SS?

CS = 1111H

The base address of the code segment is I1110H.

Effective address of memory is given by 11110H + 1232H = 12342H.

DS = 3333H

The base address of the data segment is 33330H.

Effective address of memory is given by 33330H + 0020H = 33350H.

SS 2526H

The base address of the stack segment is 25260H.

Effective address of memory is given by 25260H + 1100H = 26360H

CHAPTER ELEVEN

The 8086 Addressing Mode

The 8086 Addressing Mode

When the 8086 executes an instruction, it performs the specified function on data. Thee data are called its **operands** and may be part of the instruction reside in one of the internal registers of the 8086, stored at an address in memory, or held at an I/O port. To access these different types of operands, the 8086 is provided with various addressing modes:

1. Register Addressing Mode

With the register addressing mode, the operand to be accessed is specified as residing in an internal register of the 8086, an example of an instruction that uses this addressing mode is

MOV AX, BX

This stands for move the contents of BX, the source operand, to AX, the destination operand. Both the source and destination operands have been specified as the content of the internal registers of the 8086. See Figure 14 (a, b).

2. Immediate Addressing Mode

If a source operand is part of the instruction instead of the contents of a register or memory location, it represents what is called an immediate operand and is accessed

using the immediate addressing mode. Typically, immediate operands represent constant data. Immediate operands can be either a byte or word of data. In the instruction

MOV AL, 015H The source operand 15H is an example of a byte-wide immediate source operand. Note that the value of the immediate operand must always be preceded by a zero. See Figure 15(a, b).

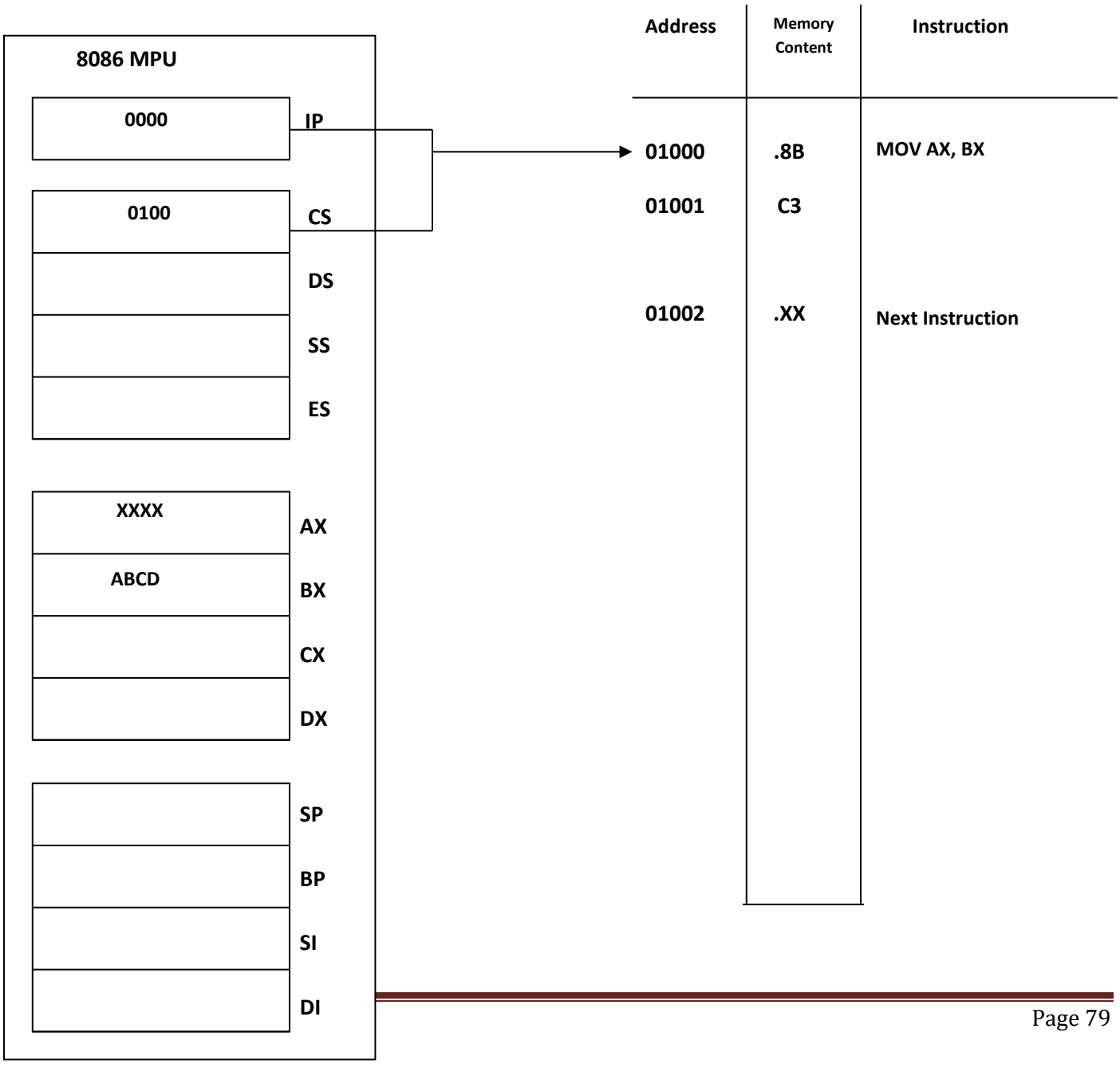


Figure 14(a): Register addressing mode before execution.

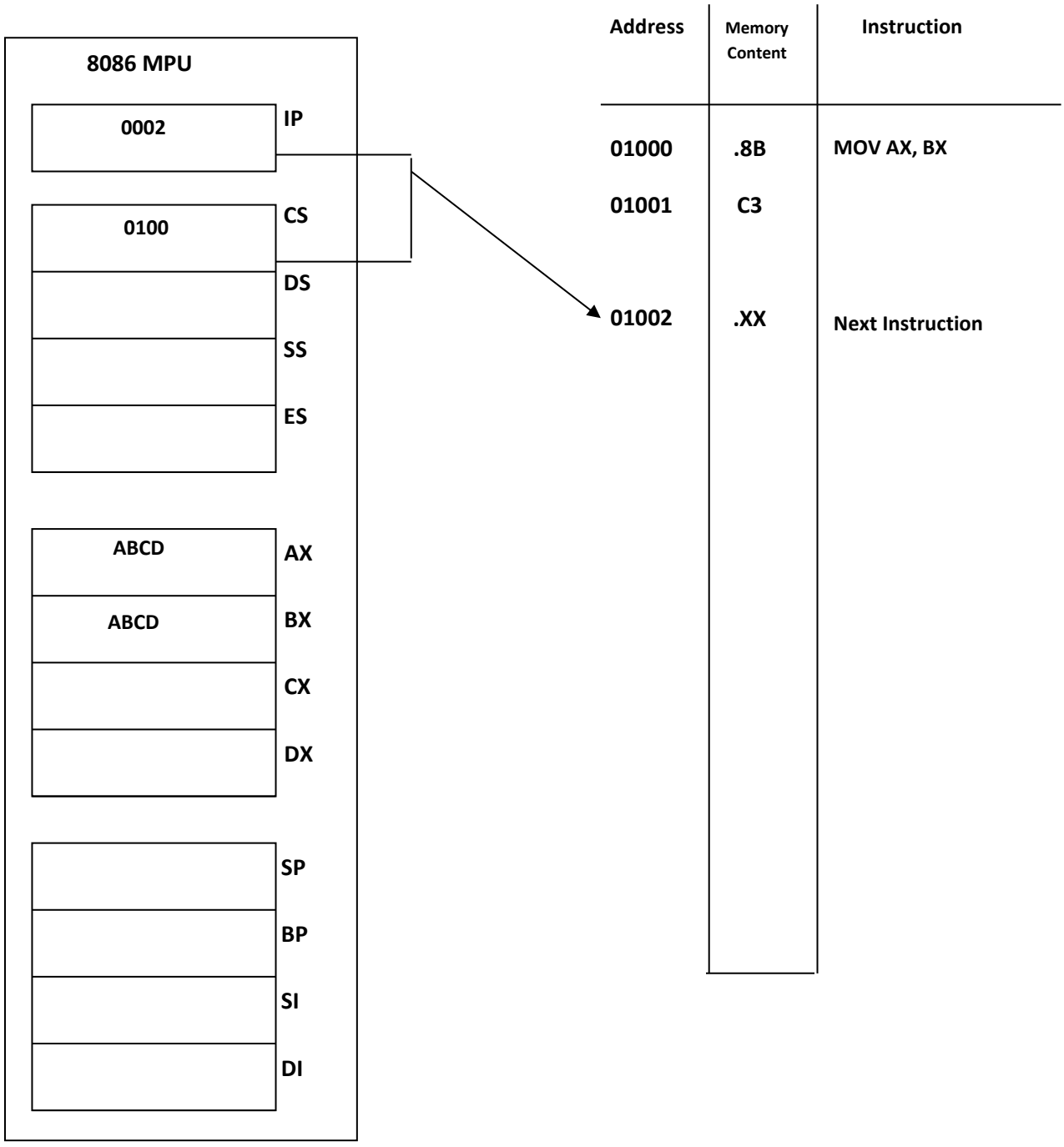


Figure 14(b): Register addressing mode after execution.

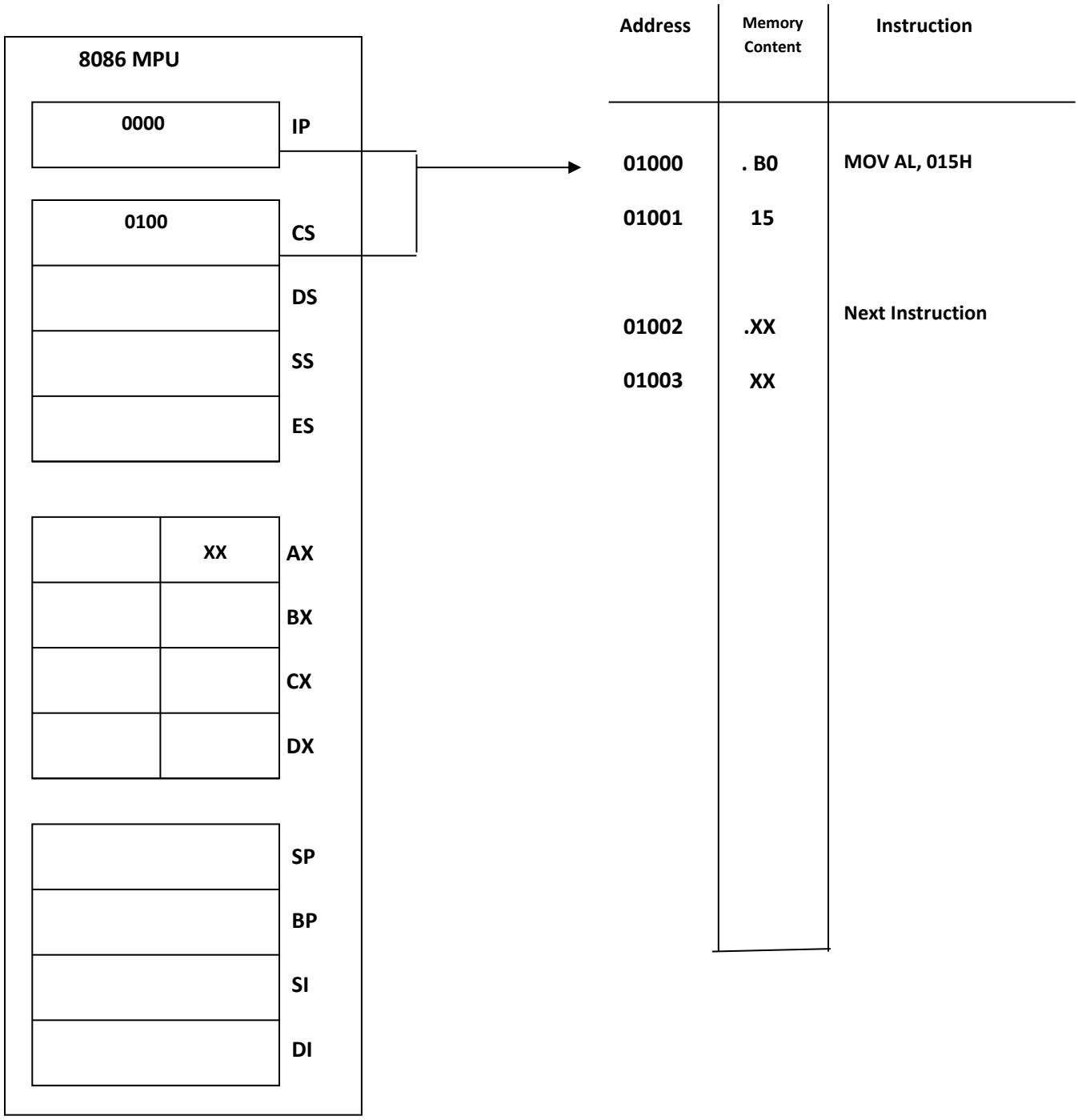


Figure 15(a): Immediate addressing mode before execution.

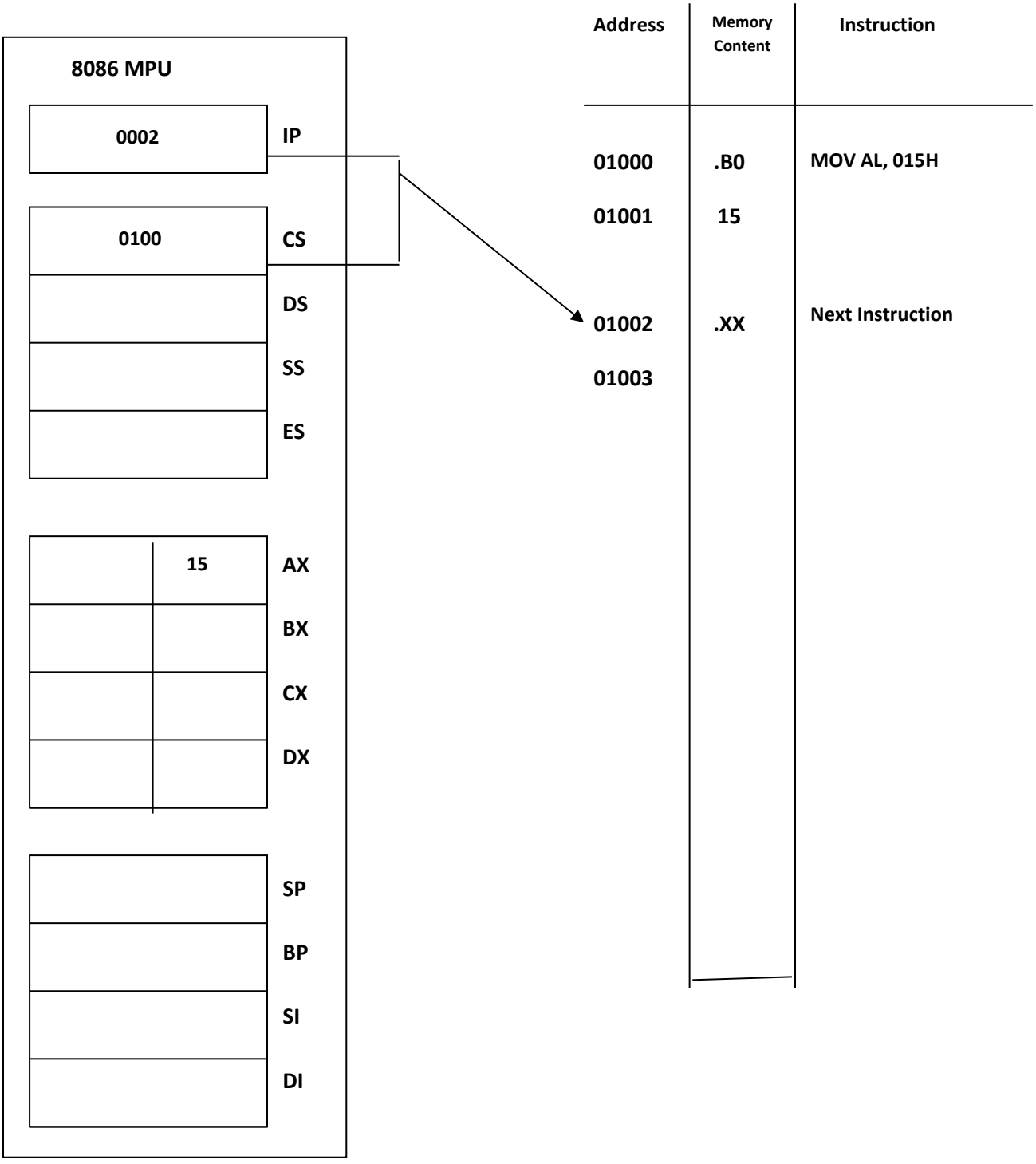


Figure 15(b): Immediate addressing mode after execution.

3. Direct Addressing Mode

Direct addressing differs from immediate addressing in that the locations following the instruction opecode hold an **effected memory address (EA)** instead of data. This effective address is a 16-bit offset of the storage location of the operand from the current value in the data segment (DS) register. EA is combined with the contents of DS in the BIU to produce the **physical address** for its source operand is

MOV CX, BETAThis stands for move the contents of the memory location which is offset by BETA from the current value in DS into internal register CX. See Figure 16(a, b). Notice that the value assigned to constant BETA is 1234_H.

$PA = 02000_H + 1234_H$

$= 03234_H$

4. Register Indirect Addressing Mode

Register indirect addressing is similar to direct addressing in that an effective address is combined with the contents of DS to obtain a physical address. However, it differs in the way the offset is specified. This time EA resides in either a pointer register or index register within the 8086. The pointer register can be either BX or BP and the index register can be SI or DI.

MOV AX, [SI] This instruction moves the contents of the memory location offset by the value of EA in SI from the current value in DS to the AX register. See Figure 17(a, b). SI contains 1234H and DS contains 0200H.

PA = 02000H + 1234H = 03234H

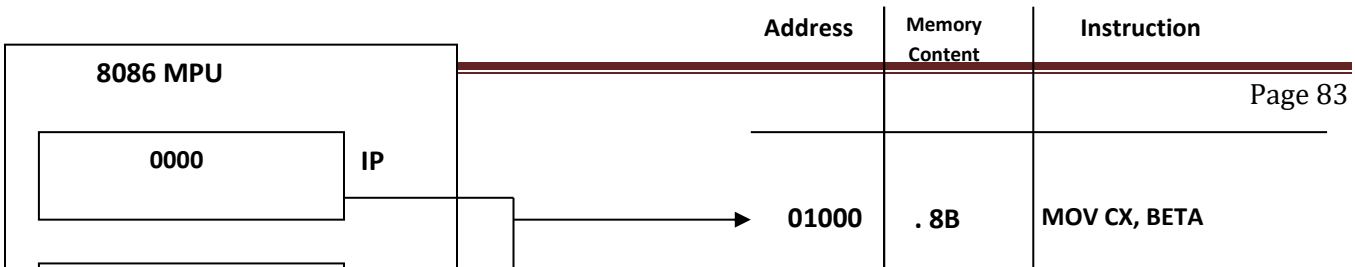


Figure 16 (a): Direct Addressing mode before execution.

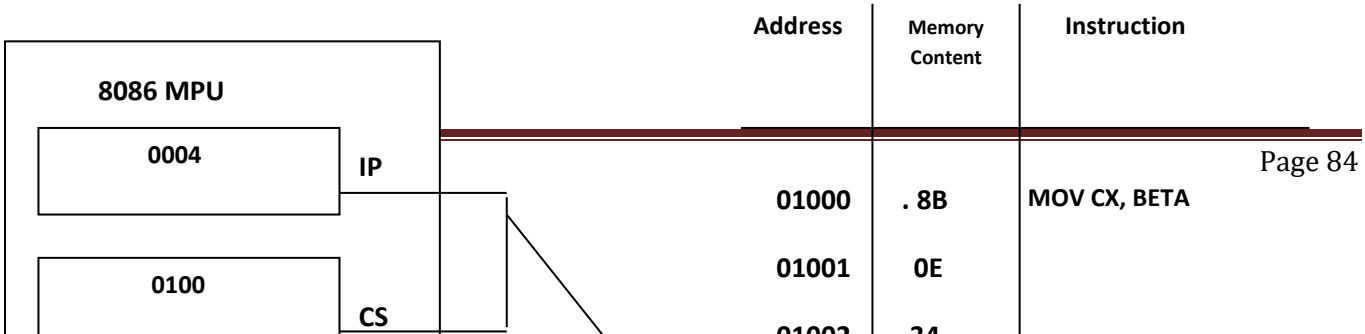


Figure 16 (b): Direct Addressing mode after execution.

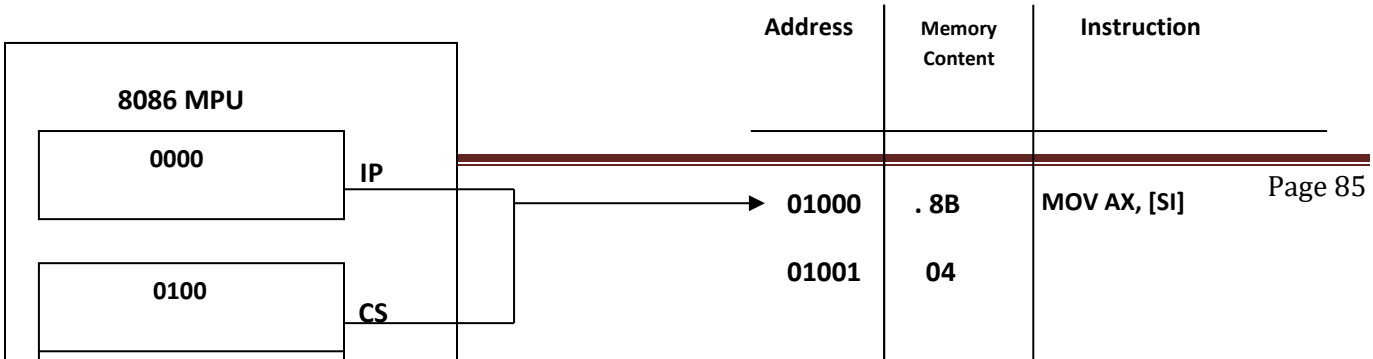


Figure 17(a): Register Indirect Addressing before execution.

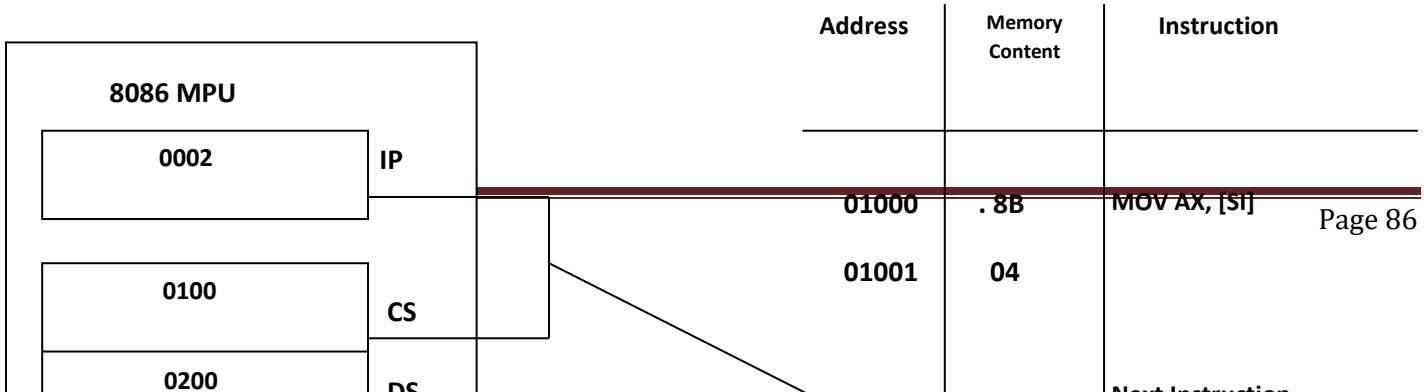


Figure 17(b): Register Indirect Addressing mode after execution.

5. Based Addressing Mode

In the based addressing mode, the physical address of the operand is obtained by adding a direct or indirect displacement to the contents of either BX or BP and the

current value in DS and SS, respectively. A MOV instruction that uses based addressing to specify the location of its destination operand is as follows:

MOV [BX].BETA, AL

As shown in Figure 18(a,b) the fetch and execution of this instruction causes the BIU to calculate the physical address of the destination operand from the contents of DS, BX, and the direct displacement. The result is

$PA = 02000_H + 1000_H + 1234_H$

$= 04234_H$

6. Indexed Addressing Mode

Indexed addressing works identically to the based addressing, it uses the contents of one of the index registers, instead of BX or BP, in the generation of the physical address, here is an example:

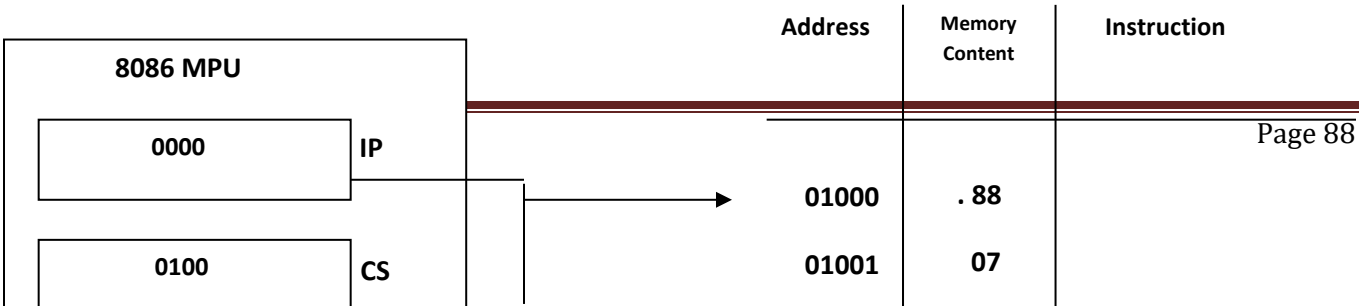
MOV AL, ARRAY[SI]

The example in Figure 19(a,b) shows the result of executing the MOV instruction. First the physical address for the source operand is calculated from DS, SI, and the direct displacement.

$PA = 02000_H + 2000_H + 1234_H$

$= 05234_H$

Then the byte of data stored at this location, which is BEH is read into lower byte AL of the accumulator register.



MOV [BX].BETA, AL

Next Instruction

Source Operand

Figure 18(a): Based Addressing before execution.

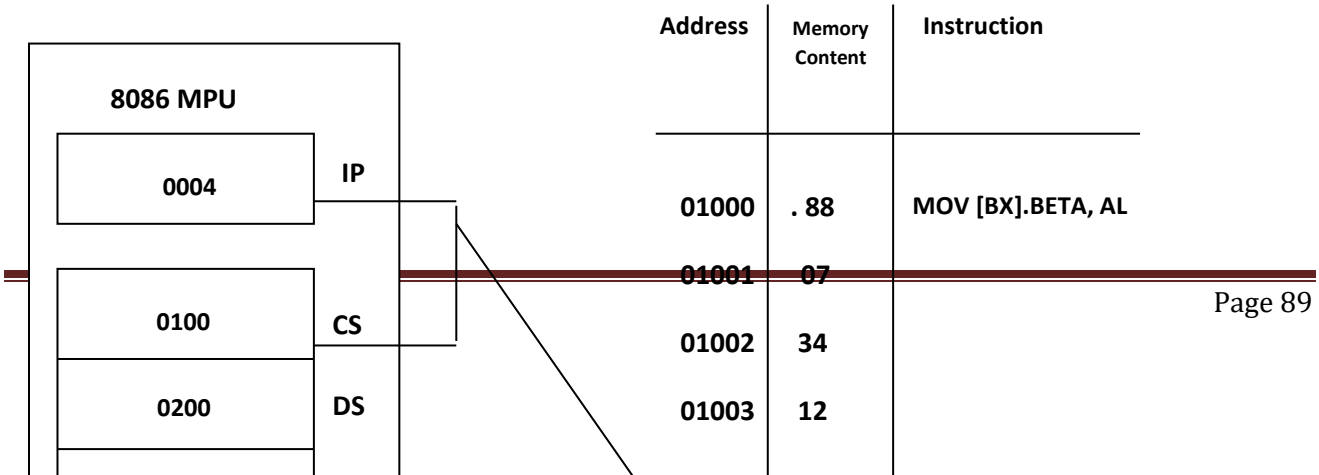


Figure 18(b): Based Addressing mode after execution.

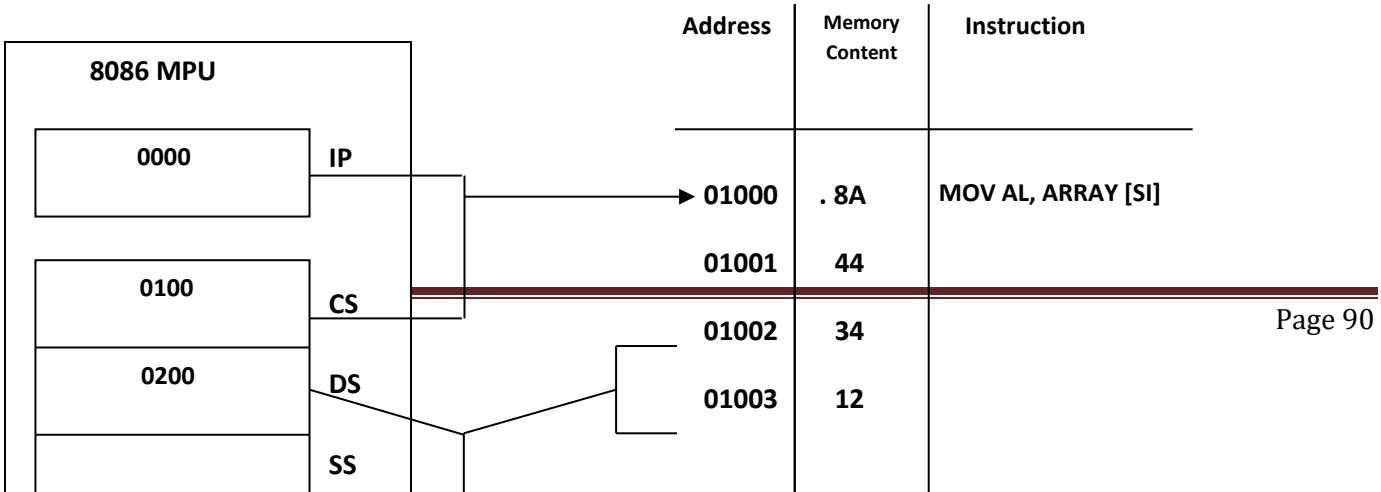


Figure 19(a): Direct Indexed Addressing before execution.

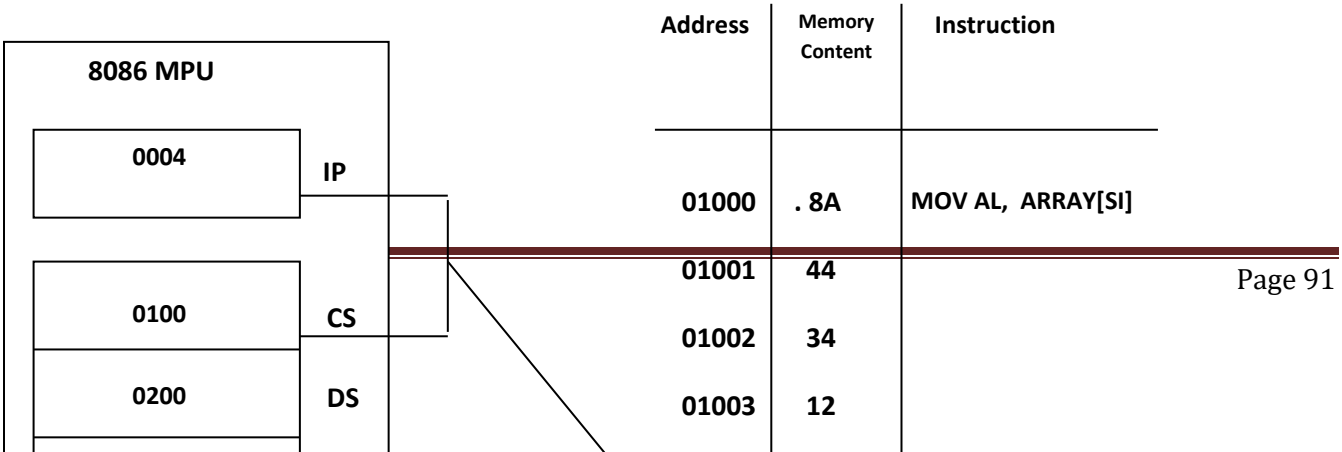


Figure 19(b): Direct Indexed Addressing mode after execution.

7. Based Indexed Addressing Mode

Combining the based addressing mode and the indexed addressing mode together results in a new, more powerful mode known as based indexed addressing. Let us consider an example of a MOV instruction using this type of addressing.

MOV AH, [BX].BETA[SI]

An example of executing this instruction is illustrated in Figure 20(a,b). The address of the source operand is calculated as

$PA = 02000_H + 1000_H + 1234_H + 2000_H$

$= 06234_H$

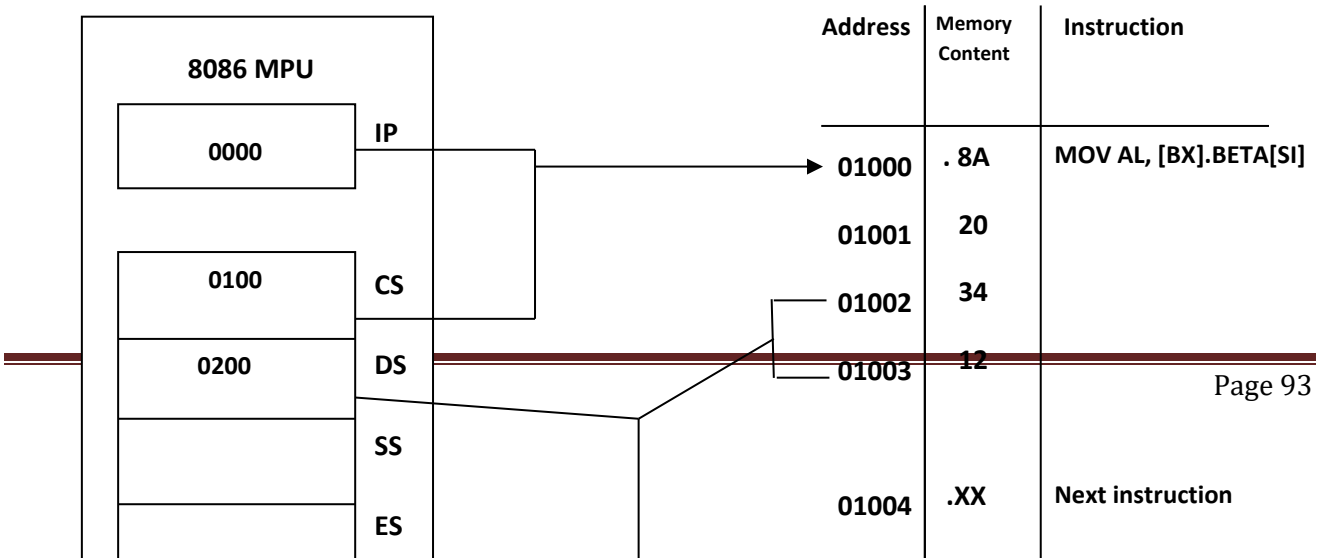
Execution of this instruction causes the Value stored at this location to be written into AH.

8. String Addressing Mode

The string instructions of the 8086's instruction set automatically use the source and destination index registers to specify the effective addresses of the source and destination operands, respectively. The move string instruction

MOVS

is an example. Notice that neither SI nor DI appears in the string instruction, but both are used during its execution.



10 00

Figure 20(a): Based Indexed Addressing before execution.

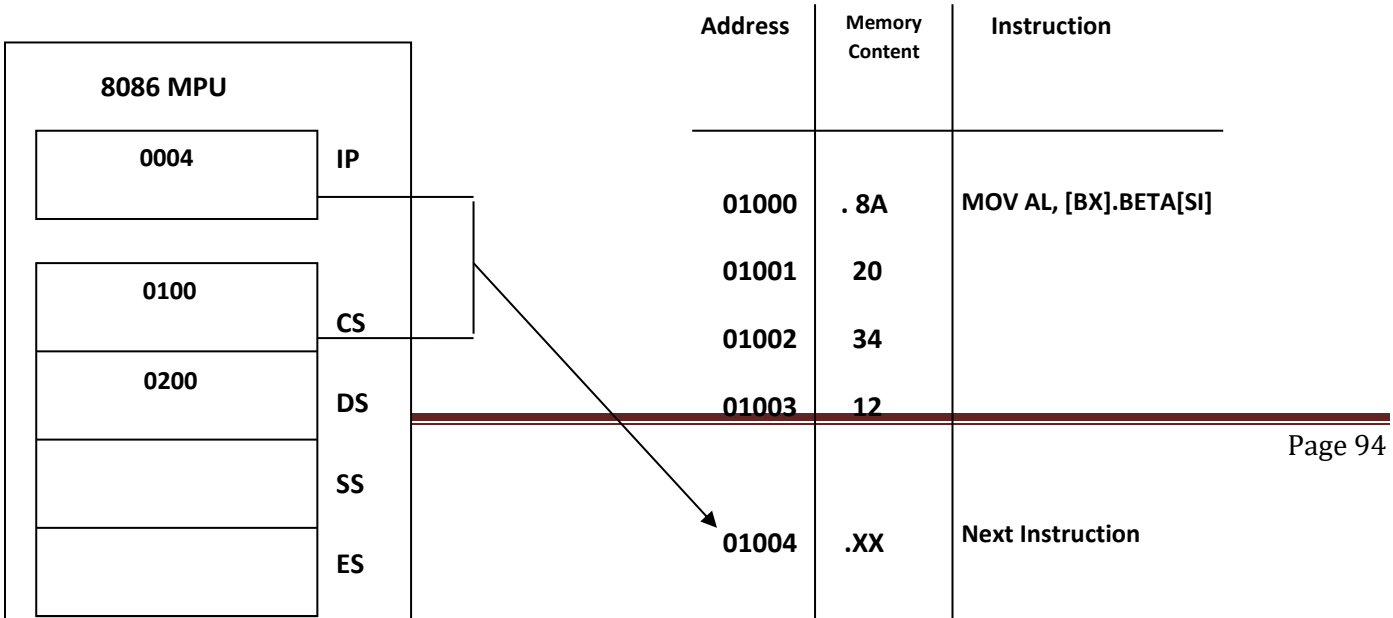


Figure 20(b): Based Indexed Addressing mode after execution.

9. Port Addressing Mode

Port addressing is used in conjunction with the IN and OUT instructions to access input and output ports. Any of the memory addressing modes can be used for the port address for memory mapped ports. For ports in the I/O address space, only the **Direct addressing mode** and an **Indirect addressing mode** using DX are available. For example, **Direct addressing** of an input port is used in the instruction

IN AL, 15_H

This stands for input the data from the byte wide input port at address 15_H of the I/O address space to register AL.

Next, let us consider another example. Using **Indirect port addressing** for the source operand in an IN instruction, we get:

IN AL, DX

It means input the data from the byte wide input port whose address is specified by the contents of register DX. For instance, if DX equals 1234_H the contents of the port at this I/O address are loaded into AL.

CHAPTER TWELVE

Input / Output programming

Input/output Programming:

Input/output Problems:

1- Wide variety of peripherals.

2- Delivering different amounts of data.

a- At different speeds.

b- In different formats.

3- All slower than CPU and RAM.

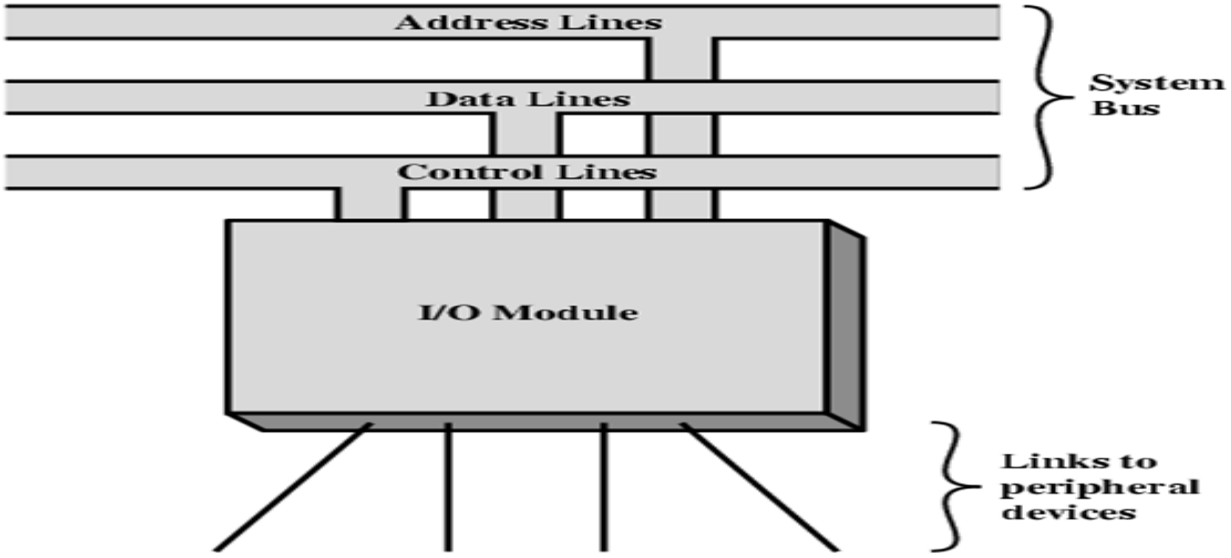
4- Need I/O modules.

Input/output Module:

*Purpose of I/O Modules***:**

a- Interface to the processor and memory via the system bus or control switch.

b- Interface to one or more peripheral devices.



External device categories:

1- Human readable: communicate with the computer user-CRT.

2- Machine readable: communicate with equipment-disk drive or ta drive.

3- Communication: communicate with remote devices may be human readable or machine readable.

Basic structure of an external device:

1- Control signals: determine the function that will be performed.

2- Data: set of bits to be sent of received.

3- Status signals: indicate the state o e device.

4- Control logic: controls the device's operations.

5- Transducer: converts data from electrical to other forms of energy.

6- Buffer: temporarily holds data being transferred.

Major faction requirement I/O Modules categories:

***Control and Timing**:*

1- Coordinates the flow of traffic between internal resources and external devices.

2- Cooperation with bus arbitration.

CPU Communication:

1- Command Decoding.

2- Data.

3- Status Reporting.

4- Address Recognition.

Device Communication:

1- Commands.

2- Status Information.

3- Data.

Data Buffering:

1- Rate of data transfer to/from CPU is orders of magnitude faster than to/from external devices.

2- I/O module buffers data so that peripheral can send/receive at its rate, and CPU can send/receive at its rate.

Error Detection:

1- Must detect and correct or report errors that occur.
2- Types of errors.
3- Mechanical/electrical malfunctions.
4- Data errors during transmission.

I/O control steps:

1- Processor checks I/O module for external device status.
2- I/O module returns status.
3- If device ready, processor gives I/O module command to request dat transfer.
4- I/O module gets a unit of data from device.
5- Data transferred from the I/O module to the processor.

I/O Module Decisions:

1- Hide or reveal device properties to CPU.
2- Support multiple or single device.
3- Control device functions or leave for CPU.

Input Output Techniques:

1- Programmed (I/O).
2- Interrupt driven.
3- Direct Memory Access (DMA).

1- Programmed I/O

Properties:

- CPU has direct control over I/O.
  1- Sensing status.
  2- Read/write commands.
  3- Transferring data.
- CPU waits for I/O module to complete operation.
- Problem with programmed I/O is CPU has to wait for I/O module to be ready for either reception or transmission of data (Wastes CPU time).

Programmed I/O - detail

- CPU requests I/O operation.
- I/O module performs operation.
- I/O module sets status bits.
- CPU checks status bits periodically.
- I/O module does not interrupt CPU directly.
- I/O module does not interrupt CPU.
- CPU may wait or come back later.

2- Interrupt Driven I/O

Properties:

- Overcomes CPU waiting.
- No repeated CPU checking of device.
- I/O module interrupts when ready.

Interrupt Driven I/O Basic Operation:

- CPU issues read command.
- I/O module gets data from peripheral whilst CPU does other work.
- I/O module interrupts CPU.
- CPU requests data.
- I/O module transfers data.

I/O module view point:

- I/O module receives a READ command form the processor.
- I/O module reads data from desired peripheral into data register.
- I/O module interrupts the processor.
- I/O module waits until data is requested by the processor.
- I/O module places data on the data bus when requested.

Processor view point:

- The processor issues a READ command.
- The processor performs some other useful work.
- The processor checks for interrupts at the end of the instruction cycle.
- The processor saves the current context when interrupted by the I/O module.

- The processor read the data from the I/O module and stores it in.
- The processor the restores the saved context and resumes execution.

Design issues:

How does the CPU determine which device issued the interrupt?

1. Multiple Interrupt Lines:

a- Most straight forward solution.

b- Impractical to dedicate many lines.

c- Multiple I/O modules are likely attached to each line.

2. Software Poll:

a- Interrupt service routine polls each device to see which caused the interrupt.
b- Time consuming.

3. Daisy Chain (hardware poll, vectored):

a- Interrupt Acknowledge sent down a chain.

b- Module responsible places vector on bus.

c- CPU uses vector to identify handler routine.

4. Bus Arbitration (vectored):

a- Module must claim the bus before it can raise Interrupt.

b- Each interrupt line has a priority.

c- Higher priority lines can interrupt lower priority lines.

d- If bus mastering only current master can Interrupt.

IF multiple interrupts have occurred, how does the CPU decide which one to process?

Multiple lines- assign priorities to lines, and pick the one with highest priority.

Software polling- order in which modules are polled determines priority.

Daisy chain- order of modules on chain determines priority.

Bus arbitration- can employ a priority scheme through the arbiter or arbitration algorithm.

3- Direct Memory Access (DMA)

Drawback of Programmed and Interrupt-Driven I/O

Interrupt driven and programmed I/O require active CPU intervention

a- Transfer rate is limited.

b- CPU is tied up, DMA is the answer.

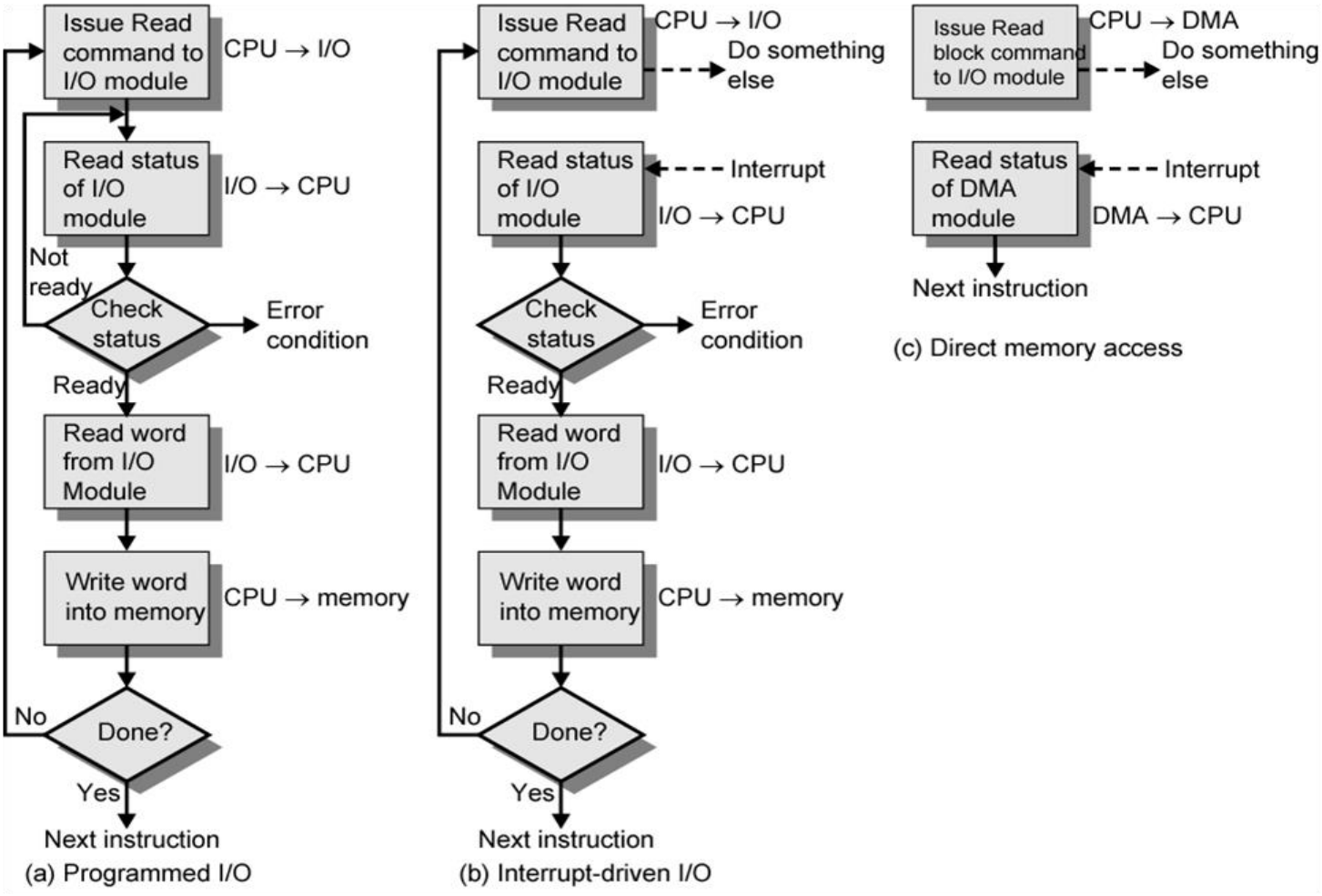
DMA Operation

- The processor issues a command to DMA module
  - ✓ Read or write.
  - ✓ I/O device address using data lines.
  - ✓ Starting memory address using data lines-stored in address.
  - ✓ Number of words to be transferred using data lines-stored in data register.
- The processor then continues with other work
- DMA module transfers the entire block of data (one word at a time directly to or from memory without going through the processor)
- DMA module sends an interrupt to the processor when complete

I/O Commands

The processor issues an address, specifying I/O module and device, and an I/O command. The commands are:

- **Control:** activate a peripheral and tell it what to do.
- **Test:** test various status conditions associated with an I/O module and its peripherals.
- **Read:** causes the I/O module to obtain an item of data from the peripheral and place it into an internal register.
- **Write:** causes the I/O module to take a unit of data from the data bus and transmit it to the peripheral.


Issue Read command to I/O module
CPU → I/O
Read status of I/O module
I/O → CPU
Not ready
Check status
Error condition
Ready
Read word from I/O Module
I/O → CPU
Write word into memory
CPU → memory
No
Done?
Yes
Next instruction
(a) Programmed I/O
Issue Read command to I/O module
CPU → I/O
Do something else
Read status of I/O module
Interrupt
I/O → CPU
Check status
Error condition
Ready
Read word from I/O Module
I/O → CPU
Write word into memory
CPU → memory
No
Done?
Yes
Next instruction
(b) Interrupt-driven I/O
Issue Read block command to I/O module
CPU → DMA
Do something else
Read status of DMA module
Interrupt
DMA → CPU
Next instruction
(c) Direct memory access

REFERENCES

1- Introduction to computers for Peter Norton 2003.

2- THE INTEL MICROPROCESSORS 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-Bit Extensions Architecture, Programming, and Interfacing Eighth Edition for BARRY B. BREY 2006.

3- "Fundamentals of computer organization and architecture" By John wiley & Sons, 2005.

4- "Fundamentals of computer organization and architecture" By John wiley & Sons, 2015.

5- Introduction to computers for Peter Norton's 2014.