UNIT-I
Generations

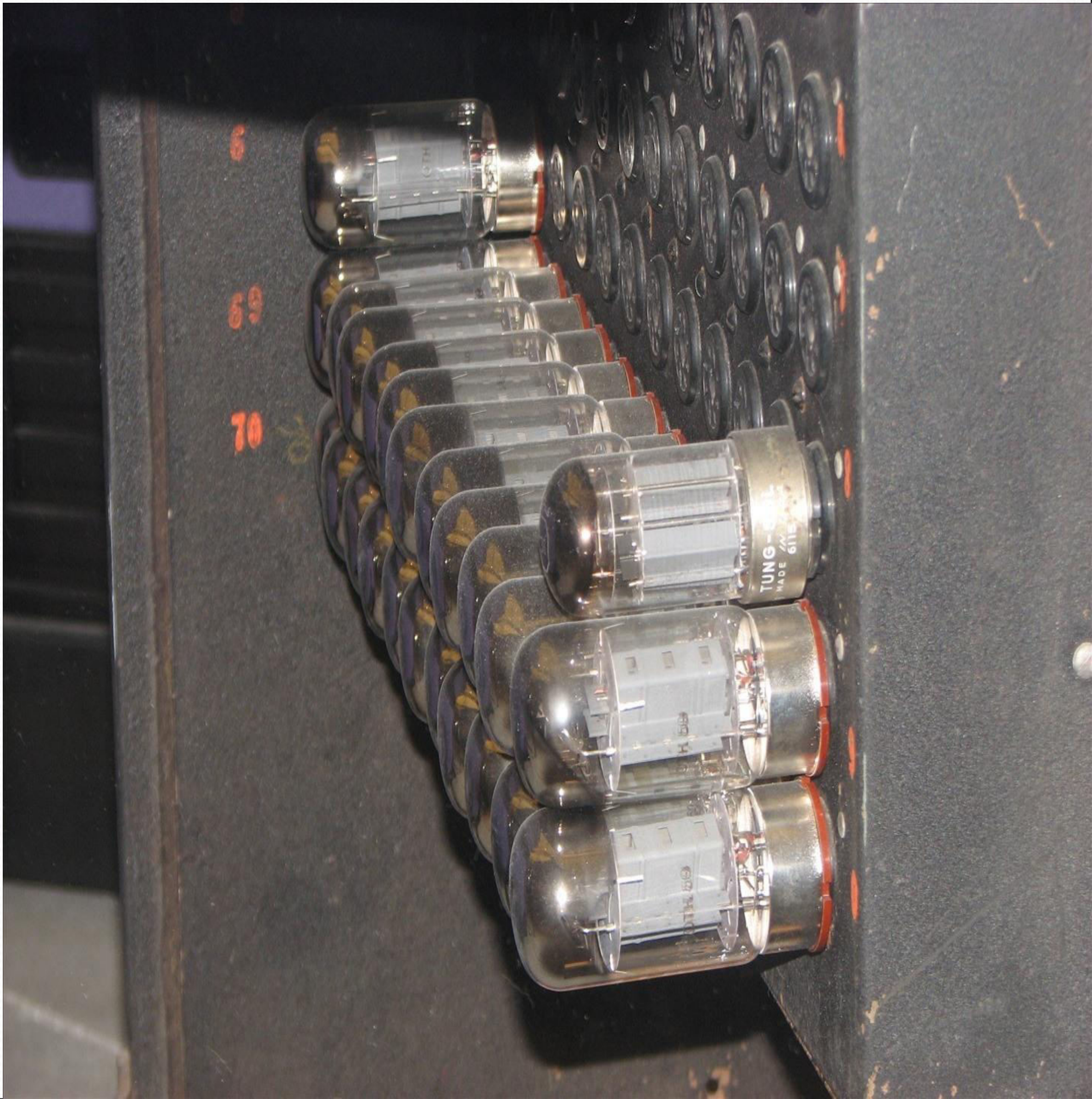
Dr. Devaraj Verma C,
Associate Professor,
Dept of CSE – SP,
Jain University

The Five Generations of Computers

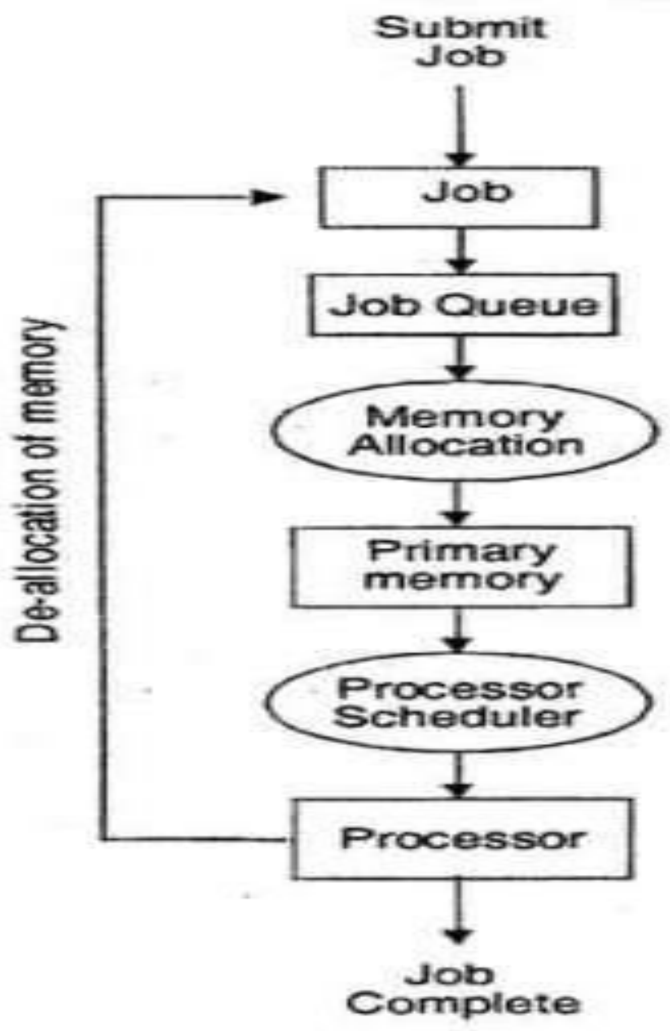
First generation computers

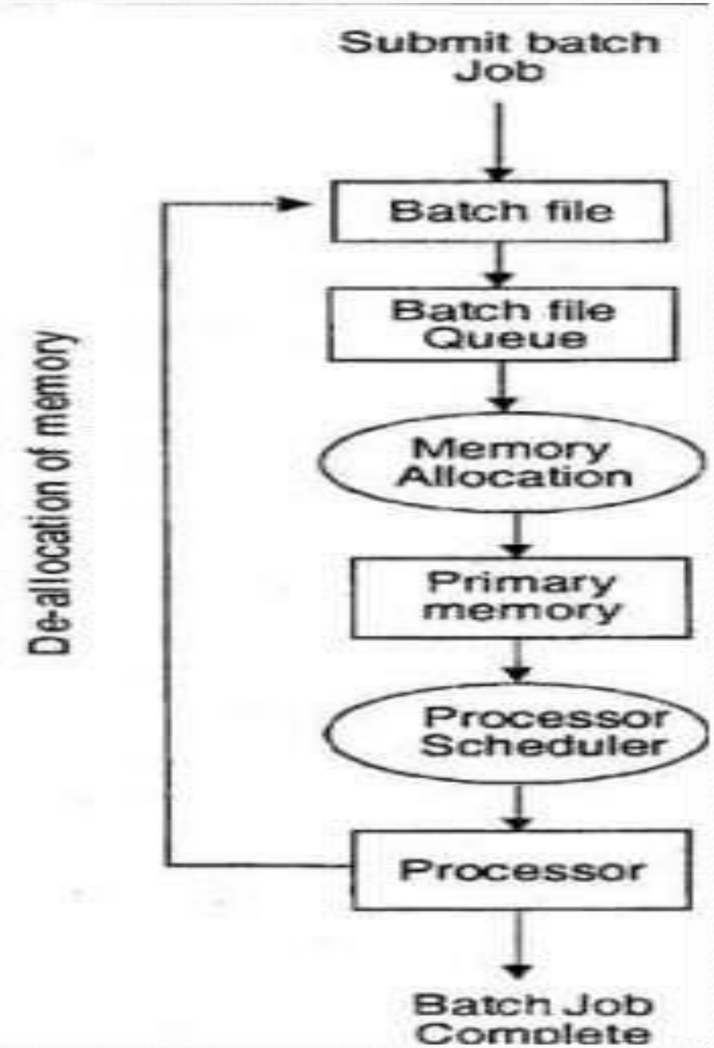
- *The period of first generation was* ***1946-1959.***
- *First generation of computers started with using* ***vacuum tubes*** *as the basic components for memory and circuitry for CPU (Central Processing Unit).*
- *These tubes like electric bulbs* ***produced a lot of heat*** *and were prone to frequent fusing of the installations.*
- *Very expensive and could be afforded only by very large organizations.*
- *In this generation, mainly* ***batch processing operating systems*** *were used.*
- *In this generation,* ***Punched cards, Paper tape, Magnetic tape*** *Input & Output device were used.*

Vacuum Tubes

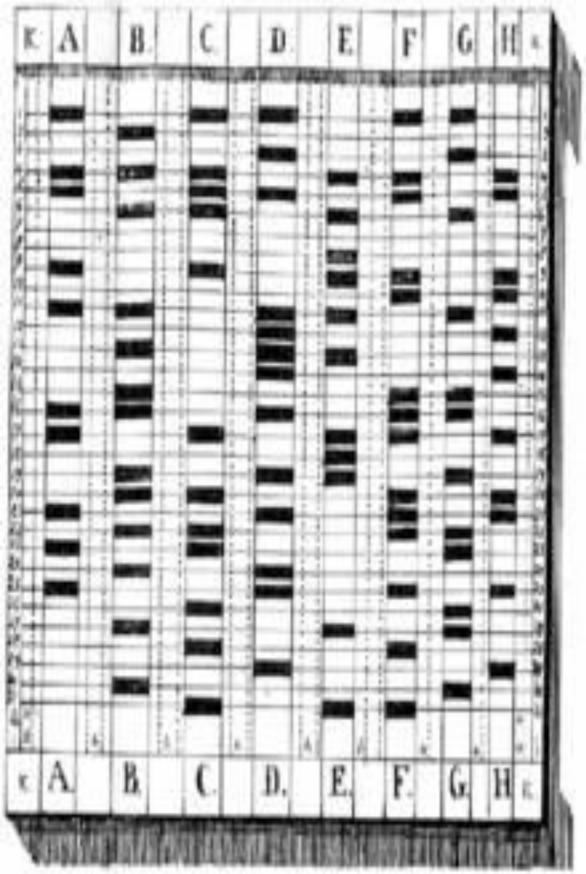


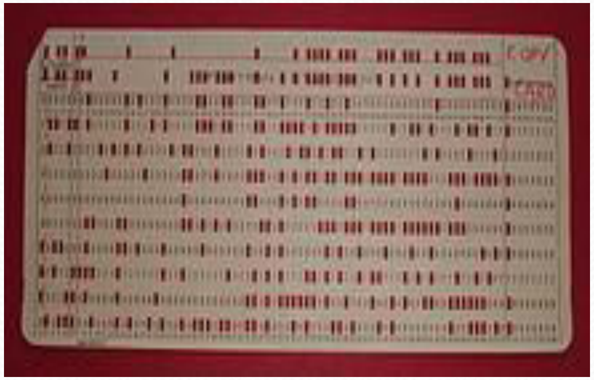
Batch Processing Operating System





Punch Card And Magnetic Tape





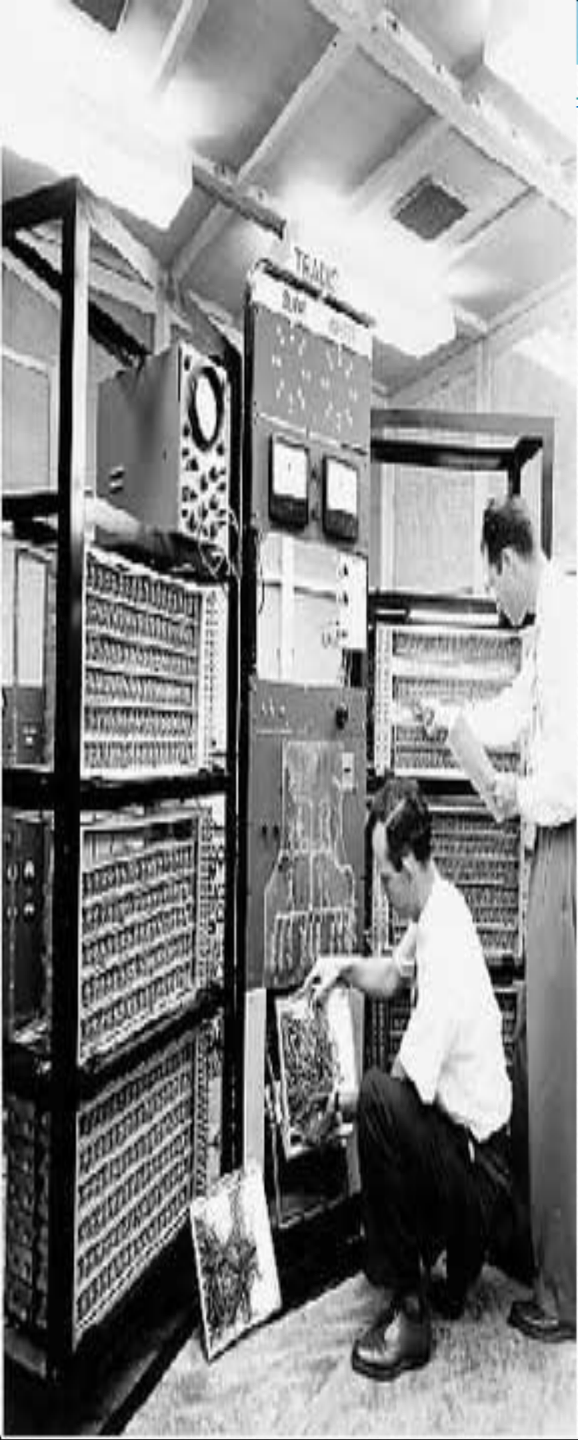
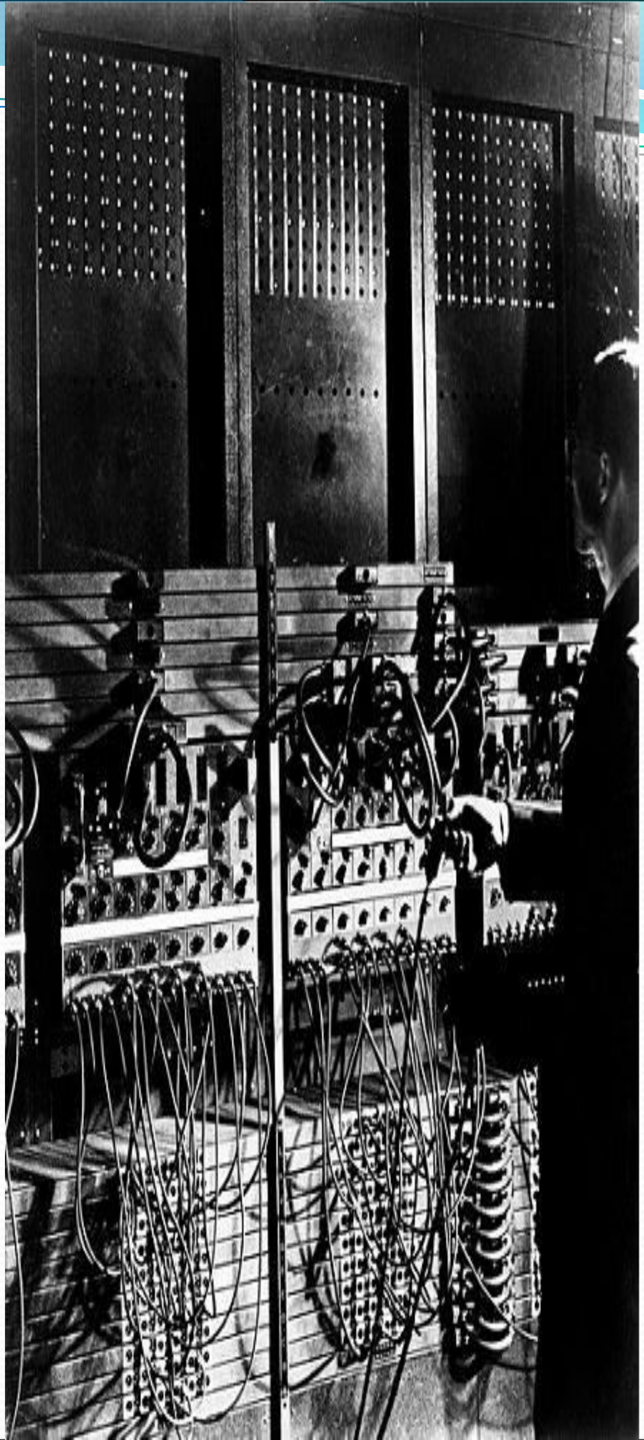
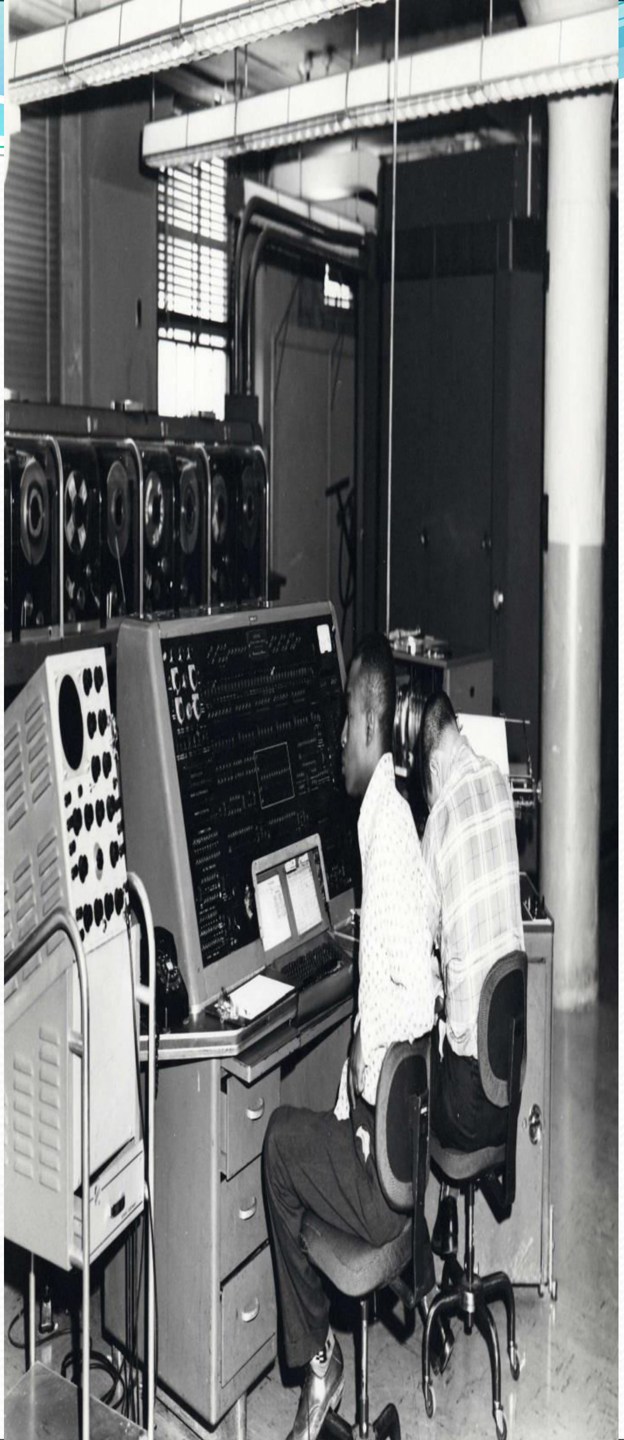
Paper Punch Card

The main features of First Generation are:

- *Vacuum tube technology*
- *Unreliable*
- *Supported Machine language only*
- *Very costly*
- *Generate lot of heat*
- *Slow Input/output device*
- *Huge size*
- *Need of A.C.*
- *Non-portable*
- *Consumed lot of electricity*

Some computers of this generation were:

- **ENIAC-Electronic Numerical Integrator And Computer** [The ENIAC contained 17,468 vacuum tubes, along with 70,000 resistors, 10,000 capacitors, 1,500 relays, 6,000 manual switches and 5 million soldered joints. It covered 1800 square feet (167 square meters) of floor space, weighed 30 tons, consumed 160 kilowatts of electrical power.]
- **EDVAC- EDVAC (*Electronic Discrete Variable Automatic Computer*)** was one of the earliest electronic computers. Unlike its predecessor the ENIAC, it was binary rather than decimal, and was a stored program computer.
- **UNIVAC- (UNIVersal Automatic Computer).** It is the first general purpose computer for commercial use.
- **IBM-701**- The IBM 701, known as the Defense Calculator while in development, was announced to the public on April 29, 1952, and was IBM's first commercial scientific computer.[1] Its business computer siblings were the IBM 702 and IBM 650.
- IBM-650


TRADIC



Second generation computers

- *The period of second generation was* ***1959-1965.***
- *This generation using the* ***transistor*** *were cheaper, consumed less power, more compact in size, more reliable and faster .*
- ***Magnetic cores*** *were used as primary memory and* ***magnetic tape and magnetic disks*** *as secondary storage devices.*
- *Assembly language and high-level programming language like FORTRAN, COBOL were used.*
- *There were* ***Batch processing and Multiprogramming Operating system*** *used.*

The main features of Second Generation are:

- *Use of transistors*
- *Reliable as compared to First generation computers*
- *Smaller size as compared to First generation computers*
- *Generate less heat as compared to First generation computers*
- *Consumed less electricity as compared to First generation computers*
- *Faster than first generation computers*
- *Still very costly*
- *A.C. needed*
- *Support machine and assembly languages*

Some computers of this generation were:

- *IBM 1620*
- *IBM 7094*
- *CDC 1604*
- *CDC 3600*
- *UNIVAC 1108*

Advantages of transistors over vacuum tubes

Smaller

Lighter

Less power consumption

More rugged

Lower voltages

Less heat

Greater reliability

Transistors: much simpler, much smaller, much cheaper, more reliable, no warm up, much faster.

Vacuum tubes: slow, expensive, fragile

Third generation computers (1965-1975)

- The development of the integrated circuit was the hallmark of the third generation of computers.
- Transistors were miniaturized and placed on siliconchips, called semiconductors.
- SSIC – 10 transistors per chip, MSIC -100 transistors per chip.
- Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system.
- Allowed the device to run many different applications at one time.

Third generation computers (1965-1975)

- Switching speed, realibilty increased by factor 10.
- Power dissipation & size reduced by factor 10.
- Overall CPU is powerful carry out 1 millons of instructions per second.
- New CPU Architectur e's ideas Virtual memory, base registers, interrupt processing ..etc., introduced.
- Improvement in Memory – main memory size- 4MB, secondary memory 100MB.
- Time shared operating system concept came.

Third generation computers (1965-1975)

- Many online applications became feasible.
- Control systems, airline reservation systems, interactive query systems , real time closed loop process control system & integrated database systems.
- Great improvement in HLLs- FORTAN, COBOL 68 PL/1.

Third generation computers

Fourth generation computers First Decade(1976-1985)

- The microprocessor brought the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip (LSIC) & millons of integrated circuits were built onto a single silicon chip (VLSI).
- The Intel 4004 chip, developed in 1971, located all the components of the computer.
- From the central processing unit and memory to input/output controls—on a single chip.
- . Fourth generation computers also saw the development of GUIs, the mouse and handheld devices.
- Magentic core memories replaced by semiconductor memories- size 16MB with cycle time-200ns.

Fourth generation computers First Decade(1976-1985)

- Powerful personal computers came & computer cost came down drastically.
- IBM's Decision -IPM PC is open & use Intel 8008 processor. OS of IBM PC used to built PCs by microsoft called as clones
- Development decentralized- individual processor controls for terminals and peripherals.
- Computer networks and distributed systems were developed. Concurrent programming languages like ADA support such systems were developed.
- UNIX OS and time shared interactive systems evolved.

Fourth generation computers Second Phase(1986-2000)

- Increase in speed of microprocessors and the size of the main memory-by factor 4 by every 3 years.
- Alpha microprocessor chip designed by DEC in 1994 packed 9.3 million transistors in single chip.(driven by 300MHz clock & can carry out billion operations per sec. 64-bit data and address bus).
- Intel designed Pentium and pentium II processors (clock speed 466MHz) and Celeron (300MHz).
- Drastic improvements in storage- hard disk(1GB). RAID techology(100GB HD).
- DVDs, CDs, Optical disks developed. Computer networks- LAN , Internet, WWW came

Fourth generation computers Second Phase(1986-2000)

- HLL Languages-C, C++, Java .
- Logic oriented Language-PROLOG.
- Functional oriented Language- HASKEL, ML

Fourth generation computers

Fifth generation computers (present and beyond)

- Fifth generation computing devices, based on artificial intelligence.
- Are still in development, though there are some applications, such as voice recognition.
- The use of parallel processing and superconductors is helping to make artificial intelligence a reality.
- The goal of fifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.
- Grid computing, cloud computing & utility computing technologies are under development.

Fifth generation computers

TABLE 12.1 Computer generations — A comparison

Generation	*Years*	*Switching device*	*Storage device*	*Switching time*	*MTBF**	*Software*	*Applications*
First	1949–55	Vacuum tubes	Acoustic delay lines and later magnetic drum. 1 KB memory	0.1 to 1 milli-second	30 mts. to 1 hour	Machine and assembly languages. Simple monitors	Mostly scientific. Later simple business systems
Second	1956–65	Transistors	Magnetic core main memory, tapes and disk peripheral memory. 100 KB main memory	1 to 10 micro-seconds	About 10 hr	High level languages. FORTRAN, COBOL, Algol, Batch operating systems	Extensive business applications. Engineering design optimization, scientific research
Third	1966–75	Integrated Circuits (IC)	High speed magnetic cores. Large disks (100 MB). 1 MB main memory	0.1 to 1 micro-second	About 100 hr	FORTRAN IV, COBOL 68, PL/1. Timeshared operating system	Database management systems. On-line systems
Fourth — First phase	1975–84	Large-scale integrated circuits. Microprocessors (LSI)	Semiconductor memory. Winchester disk. 10 MB main memory. 1000 MB disks	10 to 100 nano-seconds	About 1000 hr	FORTRAN 77, Pascal, ADA, COBOL-74, Concurrent Pascal	Personal computers. Distributed systems. Integrated CAD/CAM Real time control. Graphics oriented systems
Fourth — Second phase	1985–2000	Very large-scale integrated circuits. Over 100 million transistors per chip	Semiconductor memory. 1 GB main memory. 100 GB disk	1 to 10 nano-seconds	About 10,000 hr	C, C++, JAVA, PROLOG, Haskell FORTRAN 90/95	Simulation, Visualization, Parallel computing, Virtual reality, Multimedia

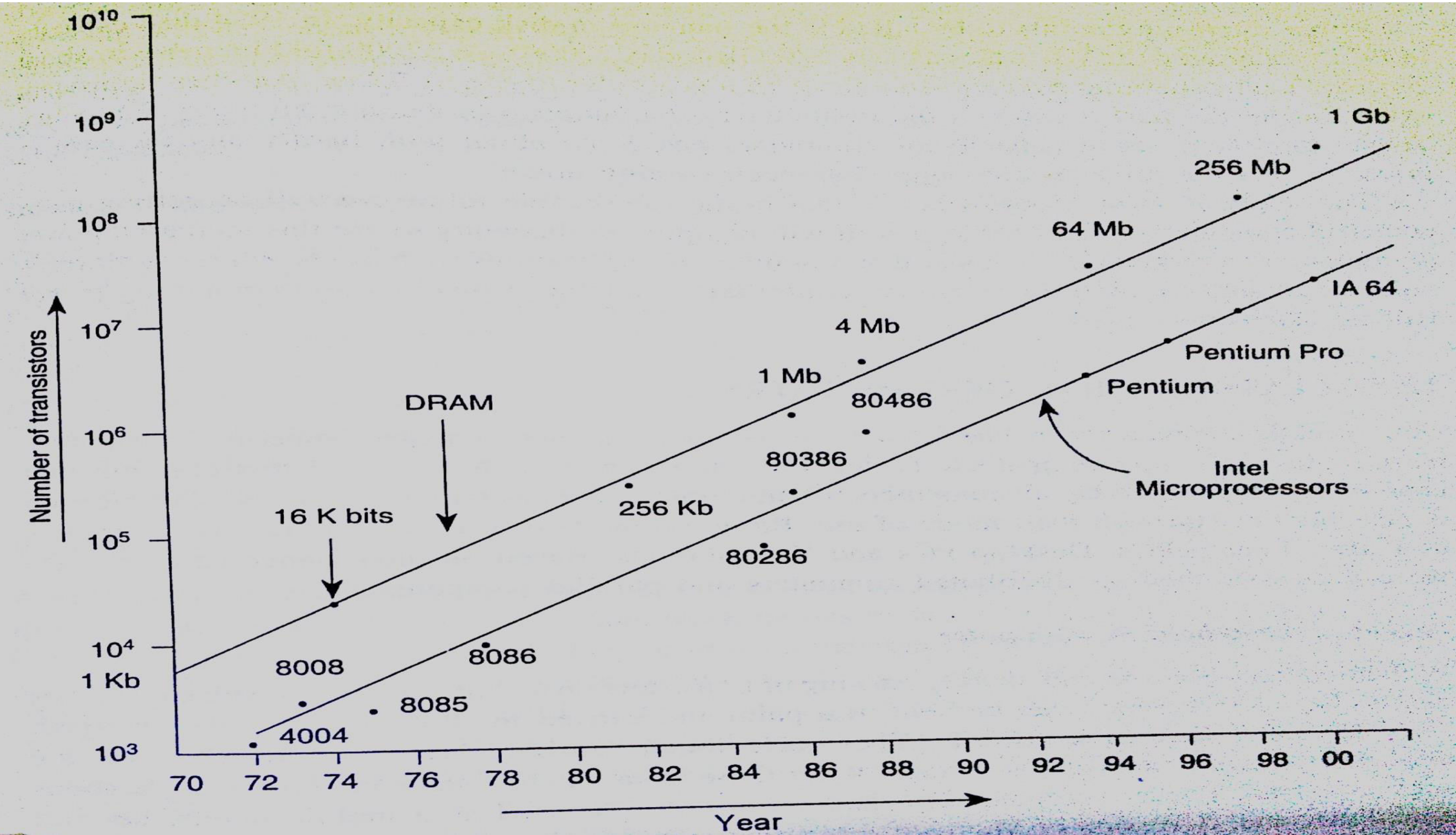
UNIT-I
Moore's Law
Classification of Digital Computers
&
Applications of Computers

Mr. Devaraj Verma C,
Associate Professor,
Department of CSE,
Jain University.

MOORE’S LAW

- The density of transistors in IC’s will double at regular intervals of around 2 years.
- Based on the experience from 1965 to date, it has been found that his prediction has been surprisingly accurate.
- In fact , the number of transistors per integrated circuit chip has approximately doubled every 18 months.
- Growth has sustained over 30 years.
- By extrapolating Moore’s law it is expected that by the year 2010, DRAMs will have nearly 100 billion bits .
- The current generation of microprocessor are 64 bit processors with clock in the range of 3GHz.
- Multicore processors emerged.(Max 16 processor in a chip).
- Increase in disk capacity.(20 MB – 2TB)@ rate 14 months.
- From past 5 years doubling of Communication bandwidth every 9 months.

Moores Law which shows the growth in the number of components in Integrated Circuits



Classification

- **Pocket or handled PCs/Simputer**
- **Personal Computers**
- **Laptops**
- **Network Computers**
- **Mini and Microcomputers**
- **PDAs**
- **Workstations**
- **Servers**
- **Thin clients**
- **Mainframes**
- **Supercomputers**

Pocket or handled PCs/Simputer

- high density of transistors on chip
- Held in a palm and shirt pockets
- Have full fledged keyboards.
- Touch sensitive screens with icons to invoke applications such as e-mail, using stylus.
- Simputer is a mobile handheld computer with input through icons on touch sensitive overlay LCD display panel.
- Uses free open source OS(GNU/Linux), no s/w cost for os. Smart card reader/writer increases the functionality of simputer.

simputer

Pocket PC

Personal Computers (PCs)

- The most popular PCs are desktop machines.
- Early PCs had Intel 8088 microprocessors as their CPU. Currently (2009), Intel Dual Core is the most popular processor.
- The machines made by IBM are called IBM PCs. Other manufacturers use IBM's specifications and design their own PCs. They are known as IBM compatible PCs.
- IBM PCs mostly use MS-Windows, WINDOWS –XP or GNU/Linux as Operating System.
- IBM PCs, nowadays (2004) have 64 to 256 MB main memory, 40 to 80 GB of Hard Disk and a floppy disk or flash ROM.
- Besides these a 650 MB CDROM is also provided in PCs intended for multimedia use.
- Another company called Apple also makes pCs known as Apple Macintosh.
- Apple Macintosh machines used Motorola 68030 microprocessors but now use Power PC 603 processor.

Personal Computers

- Used at home
- Used in Educations Institutions and Organizations as nodes
- Can perform typical activities like documentation, playing games, surfing web to more complex activities as programming, design, DTP etc.

PC

Laptops

- Same as PCs in functionality
- More Compact, Portable
- Less Power requirement, operated on battery/s
- Less capacity compared to PCs
- Maintenance and actually are costlier compared to PCs
- Theft prone

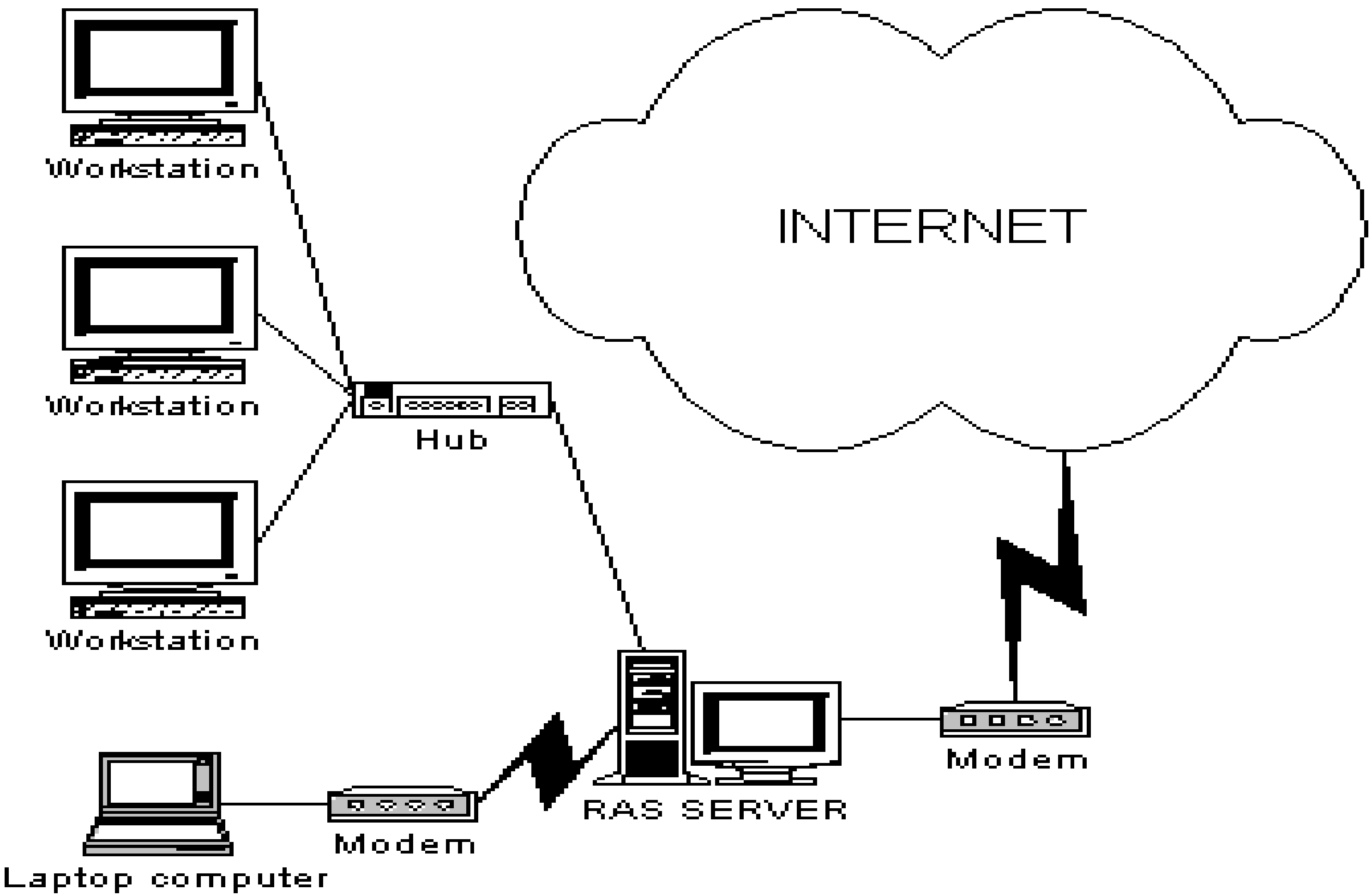
Laptop PCs:

- Laptop PCs (also known as notebook computers) are portable computers weighing around 2 kgs.
- They have a keyboard, flat screen liquid crystal display, and a Pentium or Power PC processor.
- They normally run using WINDOWS OS.
- Laptops come with hard disk (around 40 GB), CDROM and floppy disk.
- They should run with batteries and are thus designed to conserve energy by using power efficient chips.
- Many Laptops can be connected to a network.
- The most common use of Laptop computers is used for word processing, and spreadsheet computing.
- As Laptops use miniature components which have to consume low power and have to be packaged in small volumes.
- Cost is 3 to 4 times of table PCs.


PC PRO
reviews

Network Computers

- Collection of PCs
- Additional Network Interface Card
- Can share the information, work from anywhere environment
- LAN – WAN – MAN – Internet
- Uniting the World


INTERNET
Workstation
Workstation
Workstation
Hub
RAS SERVER
Modem
Modem
Laptop computer

Mini and Microcomputers

- Microcomputer is also called Personal Computer
- Minicomputers are in between mainframes and Microcomputers. They are also called midrange computers
- They are maintained by some organization

Ex. PDP - 11

PDA – Personal Digital Assistant

- Like Mini computer in a general sense
- Smaller than laptops (can be called Palmtops)
- Used to store information used frequently wherever you go
- Nowadays work with smart cards which has all the information of a user and his/her transactions


Pocket PC
Internet Explor
3:21
http://216.115.228.129:9081/wps/
Account/Contact Summary
Account Name:
ABC Electronic Controls
Address Line-1:
54672 Mac Arthur Blvd
Address Line-2:
Building 4W
City:
Irvine
State:
CA
Account Telephone:
View Tools


INDIA GATE
TAJ MAHAL
QUTUB MINAR
GATEWAY OF INDIA
CHARMINAR

Workstations

- They are similar to PCs but with more memory and a high speed processor
- They are intended to support network operating systems and network applications.
- They are used in architectural design, video editing, animations etc.

Workstations:

- Workstations are also desktop machines.
- More powerful processorspeeds about 10 times that of PCs.
- Most workstations have a large colour video display unit (19 inch monitors).
- Normally they have main memory of around 256 MB to 4 GB and Hard Disk of 80 to 320 GB.
- Workstations normally use RISC processors such as MIPS (SIG), RIOS (IBM), SPARC (SUN), or PA-RISC (HP).
- Some manufacturers of Workstations are Silicon Graphics (SIG), IBM, SUN Microsystems and Hewlett Packard (HP).

Workstations:

- The standard Operating System of Workstations is UNIX and its derivatives such as AIX (IBM), Solaris (SUN), and HP-UX (HP).
- Very good graphics facilities and large video screens are provided by most Workstations.
- A system called X WINDOWS is provided by Workstations to display the status of multiple processes during their execution.
- Most Workstations have built-in hardware to connect to a Local Area Network (LAN).
- Workstations are used for executing numeric and graphic intensive applications such as those, which arise in Computer Aided Design, simulation of complex systems and visualizing the results of simulation.

Servers

- A server is one for which many PCs are connected.
- It has large capacity secondary storage and more memory
- They host, like workstations, network servers and operating systems
- They avoid duplicate installation of applications and all users will access to a common copy of the program

Compaq Server

Servers

- While manufacturers such as IBM, SUN and Silicon Graphics have been manufacturing high performance workstations.
- The speed of Intel Pentium Processors has been going up by 2009, Pentium with clock speed 3 GHz are available.
- They can support several GB main memories.
- Thus the difference between high end PCs and Workstations is vanishing.
- Now servers are used for specific purpose such as high performance numerical computing (called compute server).
- web page hosting, database store, printing etc.
- Compute servers have performance processors with large main memory,
- database servers have big on-line disk storage (100s of GB) and print servers support several high speed printers.

Thin clients

- An alternate to desktop PCs to access servers connected to the network. Destktop PCs disadvantages are: -they rapidly become absolete, consume too much power, each DPCs need OS and Other s/w such as msoffice, virus protection ..etc
- A thin client is terminal to access a server(s) connected to the network. povides icons on the screen to invoke applicatons on the server. Results are sent back to the thin client. Advantages are:
- Cheaper and not absolete.
- System administration simple (as only servers need to be attended)
- s/w cost is lower (as thin clients do not have native s/ws such as msoffice, virus protection ..etc
- Disadvantages are: Dependency on network which may get congested, cannot be used as independent computers.

Mainframes

- Large computers both in terms of physical size as well as computations
- They support huge numbers of users
- Basically used to store and process huge amount of data
- Not all organizations can offered to maintain one mainframe. Take service of one vendor

Mainframes Computers

- There are organizations such as banks and insurance companies process large number of transactions on-line.
- They require computers with very large disks to store several Terabytes of data and transfer data from disk to main memory at several hundred Megabytes/sec.
- The processing power needed for such computers is hundred million transactions per second.
- These computers are much bigger and faster than workstations and several hundred times more expensive.
- They normally use proprietary operating systems, which usually provide high expensive services such as user accounting, file security and control.
- They are normally much more reliable when compared to Operating System on PCs. These types of computers are called mainframes.
- These are a few manufacturers of mainframes (e.g., IBM and Hitachi)


Supercomputers

- Used in scientific and engineering applications those handling huge data and do a great amount of computation.
- Extremely fast in operation (@ 1 trillion operations/second)
- Fastest, costliest and powerful computer available today
- Application involves, weather forecasting, military applications, electronic design etc.

Supercomputers

- Supercomputers are the fastest computers available at any given time and are normally used to solve problems, which require intensive numerical computations.
- Examples of such problems are numerical weather prediction, designing supersonic aircrafts, design of drugs and modeling complex molecules.
- All of these problems require around 10^16 calculations to be performed.
- These problems will be solved in about 3 hours by a computer, which can carry out a trillion floating point calculations per second.
- Such a computer is classifieds as supercomputer today .
- By about the year 2006 computers which can carry out 10^15 floating point operations per second on 64 bit floating point numbers would be available and would be the ones which be called supercomputers.
- Interconnecting several high speed computers and programming them to work cooperatively to solve problems build supercomputers.


NASA

Characteristics of Computers

- Speed – MIPS/BIPS
- Accuracy
- Reliability – No human Intervention
- Storage Capacity
- Diligence – same result forever

Applications

- At Home
  - Mostly to check mails
  - Small documentation
  - Gaming
  - Music and Video
  - To solve homework
  - Photo Printouts using Good Printers
  - Work from Home concept

Applications..

- In Education
  - Schools to Universities
  - To Educate necessary skills demanded by Industries
  - To give a demo or training
  - Server the purpose of Teaching Aids
  - To convey messages using Internet

Applications…

- In Science
  - To analyze large data acquired over a period of time
  - To do complex floating point arithmetic
  - Image Processing
  - Research

Applications….

- In Industry
  - To develop software, mostly to automate the manual work
  - To provide necessary solution to clients' needs
  - Software is developed for the needs of networking, banking, business, retail etc

Applications…..

- Entertainment
  - Music Industry
  - Games
  - Movies – to watch and create – 200 Linux Machines in parallel to create visualization in Titanic, the movie
  - III^{ly} Cartoons, special effects
  - Nowadays to promote theirs productions

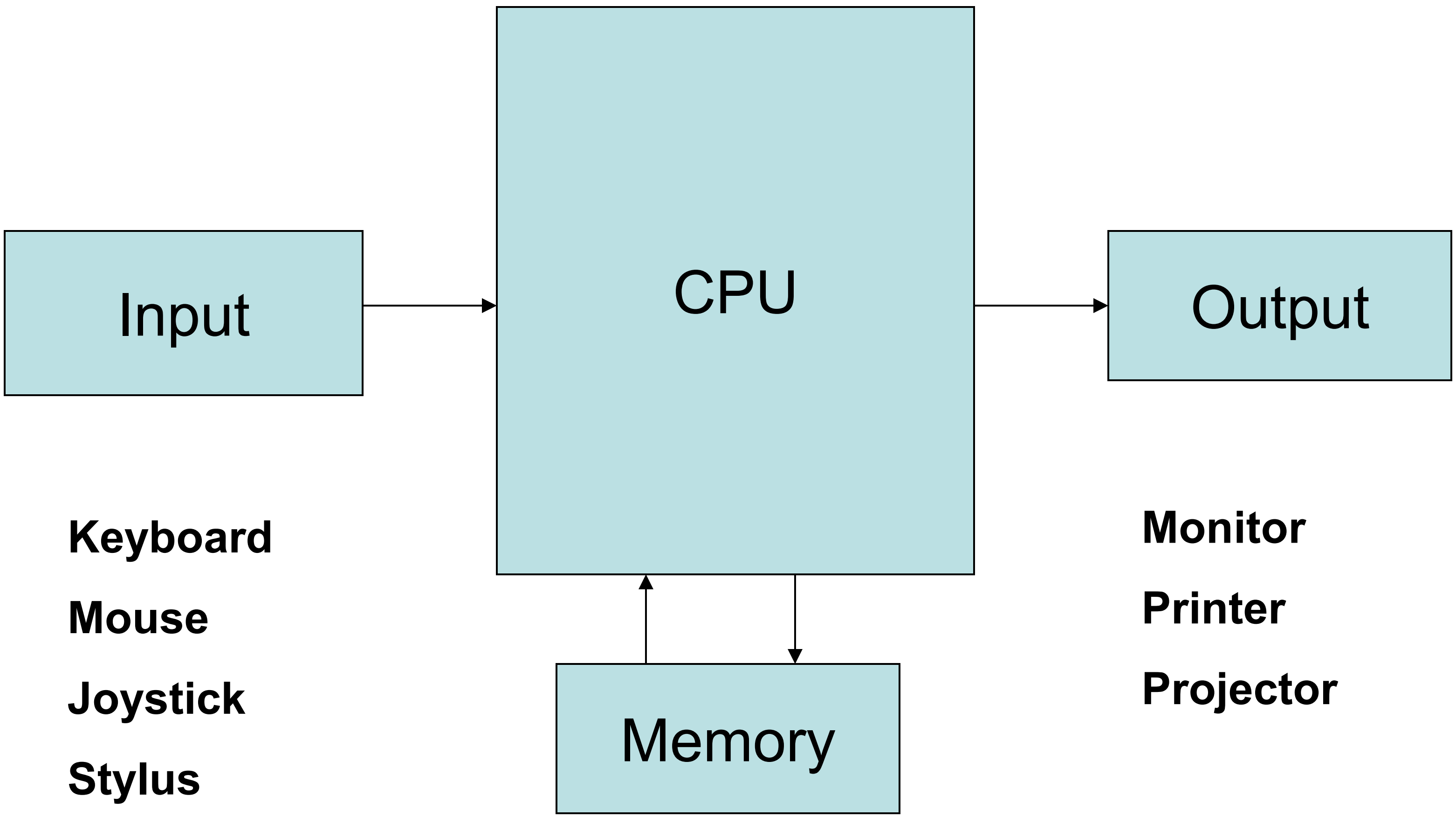
Business

- Banking
  - To store, access and modify huge amounts of data
  - Online business called e-business is becoming popular with a small amount of limitations
  - Paying bills become easy and time saving
  - online promotions

Applications……

- Government
  - “Biometrics Attendance Monitoring”
  - Weather Forecasting and military applications
  - E- governance
  - Online payment of taxes, Insurances
  - Send Messages to virtually unreachable places at present
  - Wireless communication

A typical computer



Keyboard

Mouse

Joystick

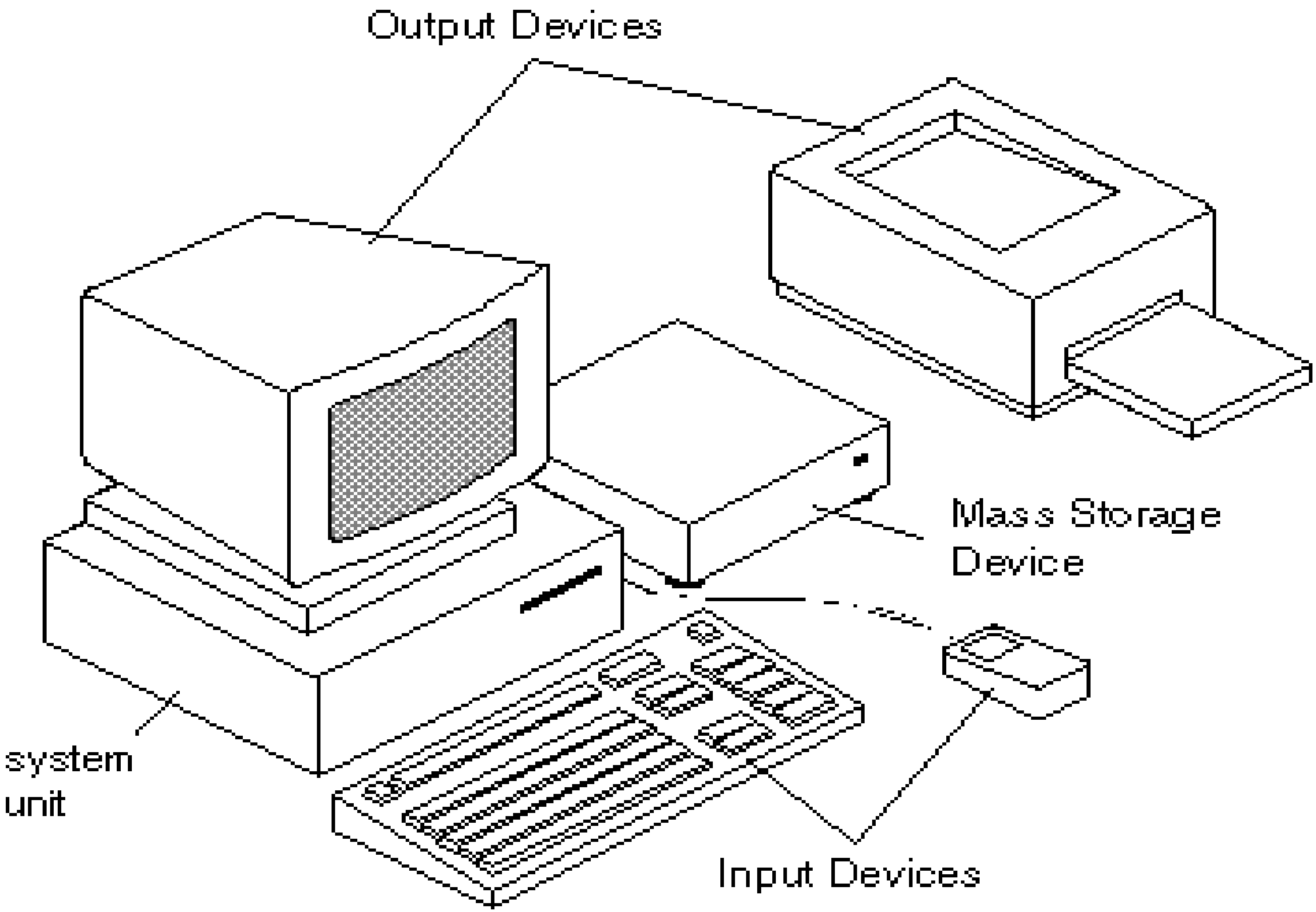
Stylus

Monitor

Printer

Projector

Primary, Secondary & fixed, portable


Output Devices
Mass Storage
Device
system
unit
Input Devices

Hardware and Software

- Hardware
  - Whatever we see physically
- Software
  - Set of instructions written using a language
  - Application Vs System software
  - Natural Vs Artificial Languages

Criticality of an Application

- Science – Solve a problem using computer
- Commercial – Payroll management
- OLTP – Ticketing Reservation Systems
- Process Control Applications – Boiler Pressure Control System
- Satellite Communication

Thank You

What is a computer?

A computer is an **electronic device**, operating under the control of instructions (i.e. **software**) stored in its own **memory** unit, that can:

- accept data (input)
- manipulate data (process)
- produce results (output)
- store the results (storage)

Computer system is used to describe a **collection of devices** that function together as a system.

What are the collection of devices?

Devices that comprise a computer system

Data and Information

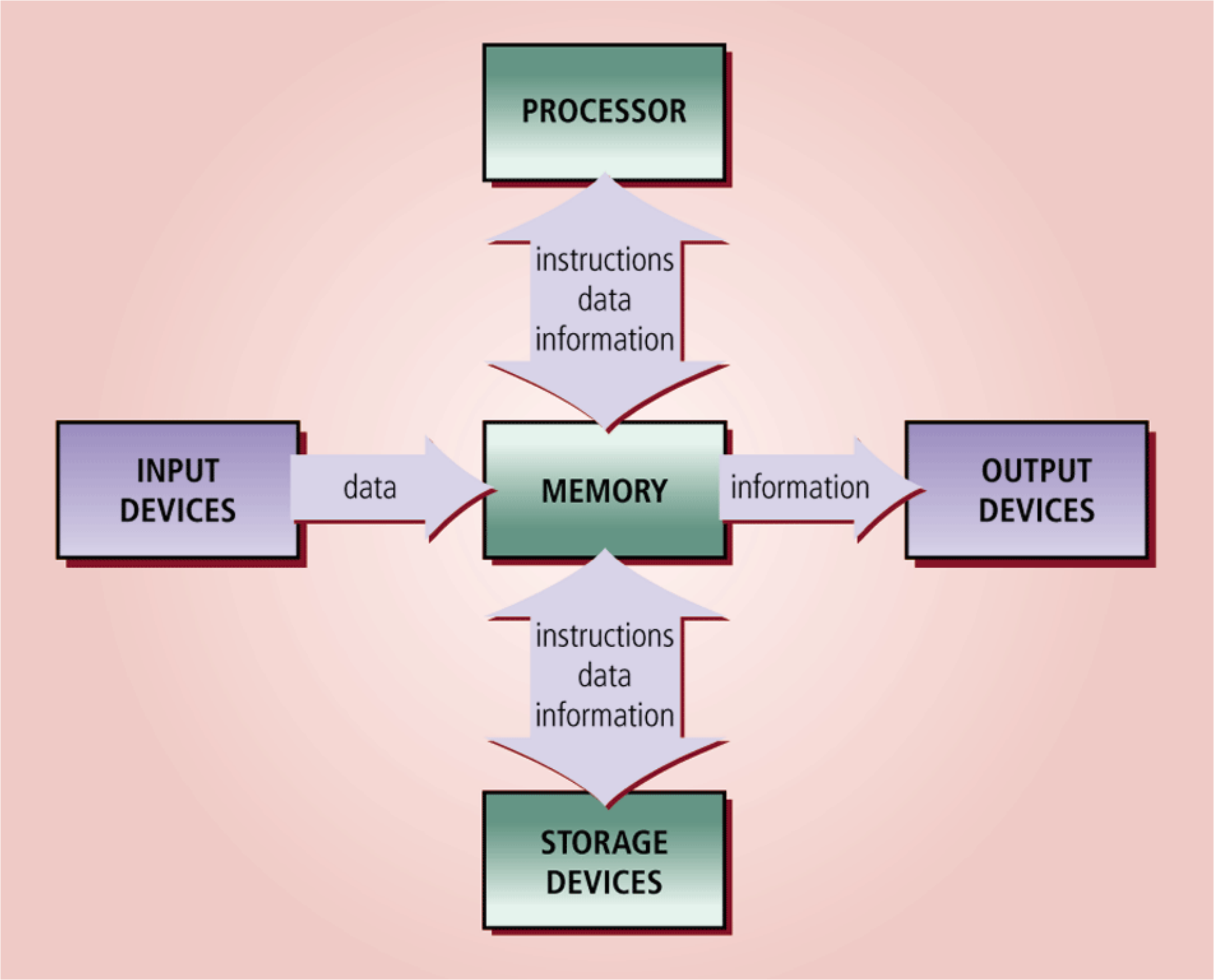
- All computer processing requires data, which is a collection of raw facts, figures and symbols, such as numbers, words, images, video and sound, given to the computer during the input phase.

- Computers manipulate data to create information. Information is data that is organized, meaningful, and useful.

- During the output Phase, the information that has been created is put into some form, such as a printed report.

- The information can also be put in computer storage for future use.

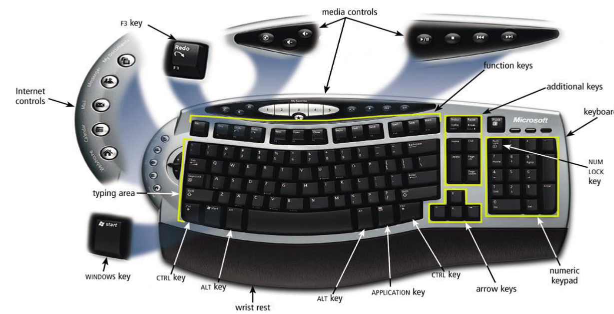
Computer system organization

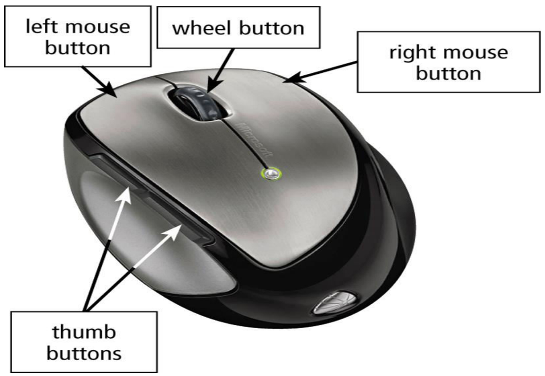


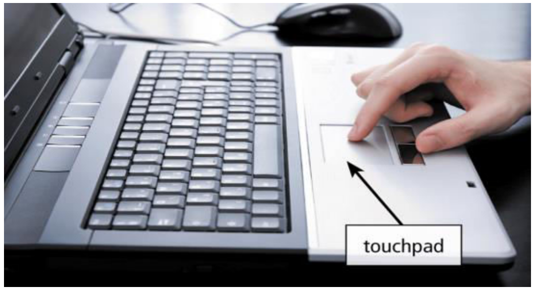
Input Device

An input device is any hardware component that allows you to enter data and instructions into a computer.

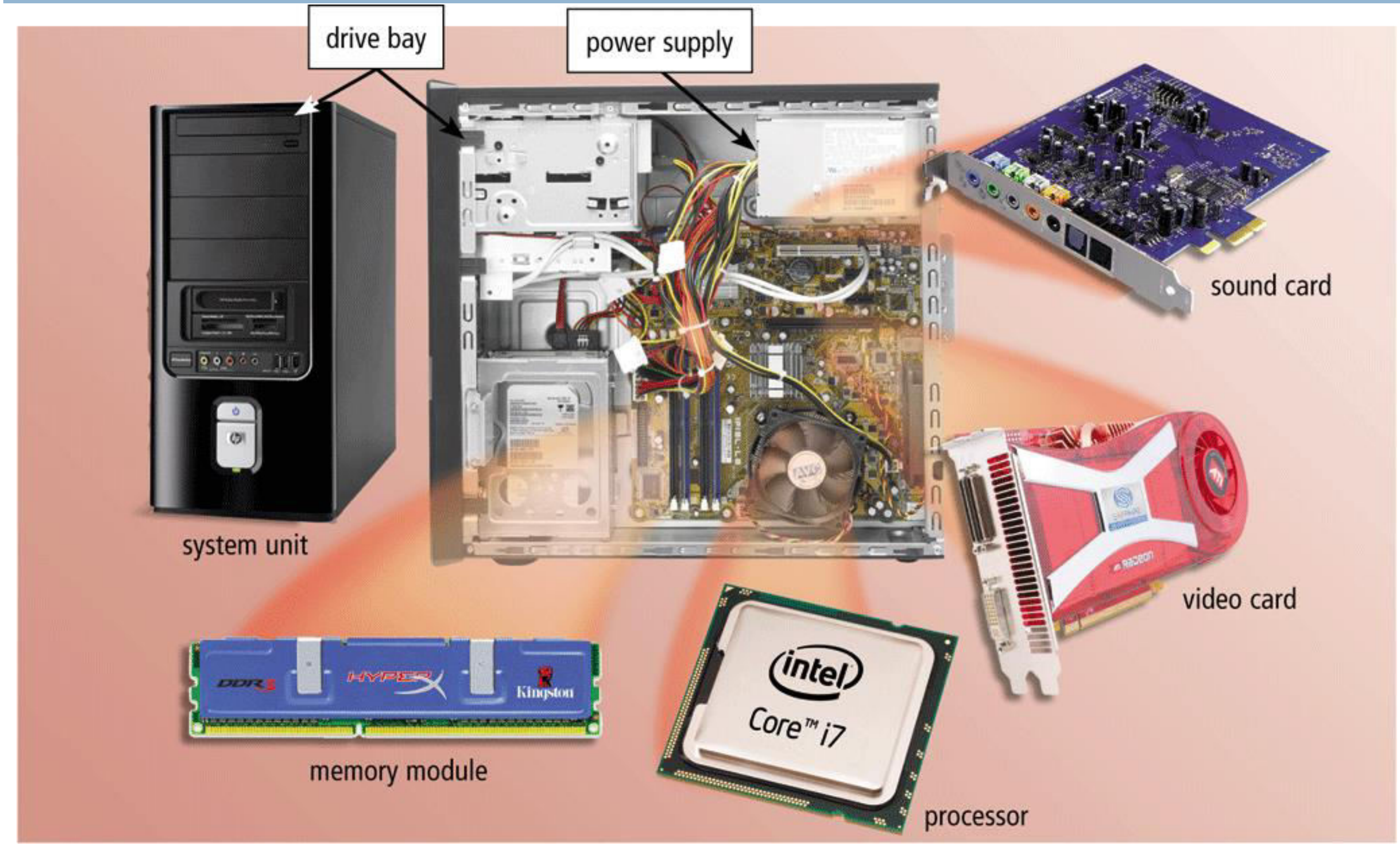
- A keyboard is an input device that contains keys users press to enter data and instructions into the computer
- A mouse is a pointing device that fits comfortably under the palm of your hand
- Most notebook computers have a touchpad, a small, flat, rectangular pointing device







System Unit


drive bay
power supply
system unit
sound card
video card
memory module
processor
intel
Core™ i7
Kingston

Output Devices

Output devices are hardware components that convey information to one or more people. A printer is an output device that produces text and graphics on a physical medium such as paper

- Ink-jet printer
- Photo printer
- Laser printer
- Multifunction peripheral (all-in-one device)

Display Device

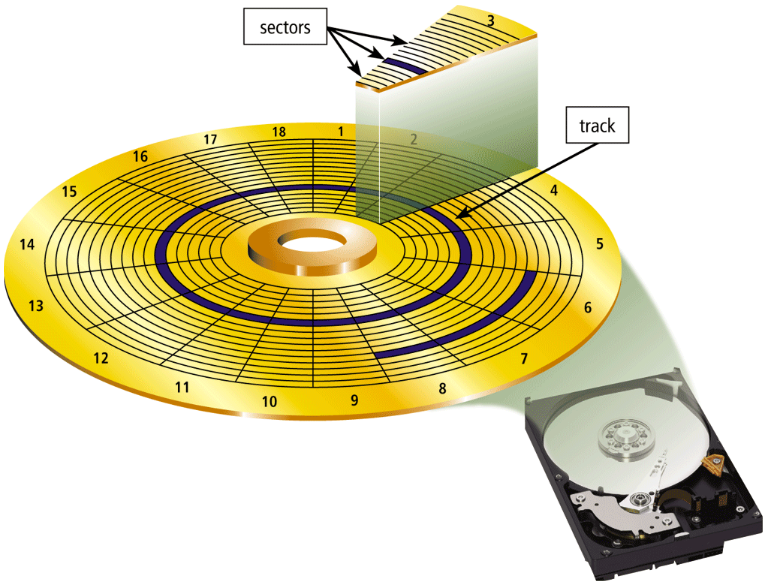
A **display device is an output device** that visually conveys text, graphics, and video information.

- LCD monitors use a liquid crystal display to produce images on the screen

Storage Devices

A storage device is the computer hardware that records and/or retrieves items to and from storage media.

A hard disk is a storage device that contains one or more inflexible, circular platters that use magnetic particles to store data, instructions, and information in tracks and sectors



Storage Devices cont…

An **external hard disk** is a separate freestanding hard disk that connects with a cable to a port on the system unit and communicates

A **removable hard disk** is a hard disk that you insert and remove from a drive.

Storage Devices cont…

Flash memory is a type of memory that can be erased electronically and rewritten

A memory card is a removable flash memory device, usually no bigger than 1.5 inches in height or width, that you insert and remove from a slot in a computer, mobile device, or card reader/writer

Storage Devices

A USB flash drive, sometimes called a thumb drive, is a flash memory storage device that plugs into a USB port on a computer or mobile device

Communication Devices

A communications device is a hardware component that enables a computer to send (transmit) and receive data, instructions, and information to and from one or more computers

Software

Software consists of a series of instructions, organized for a common purpose, that tells the computer hardware what tasks to perform and how to perform them. This set of instructions is sequenced and organized in a computer program. Therefore, a program is a series of instructions which is intended to direct a computer to perform certain functions and executed by the processors.

So in broader sense

Software can be described as a set of related programs and it is more than a collection of programs.

Software cont…

Software

System software is designed to facilitate and coordinate the use of the computer by making hardware operational. It interacts with the computer system at low level. Examples are operating system, loader, linker etc

Application software is designed to perform specific usages of the users. Examples are Microsoft Word, Microsoft Excel, Microsoft Power Point, Microsoft Access, Page Maker, Coral Draw, Photoshop, Tally, AutoCAD, Acrobat, Win Amp, Micro Media Flash, iLeap, Xing MP3 Player etc

Hardware

Hardware is **the physical components** of a computer that includes all **mechanical, electrical, electronic and magnetic** parts attached to it.

A computer consists of the following major hardware components:

- I/O i.e. Input/output devices
- CPU i.e. central processing unit
- Memory unit and storage devices
- Interface unit

Hardware cont…

Hardware is **the physical components** of a computer that includes all **mechanical, electrical, electronic and magnetic** parts attached to it.

A computer consists of the following major hardware components:

- I/O i.e. Input/output devices
- CPU i.e. central processing unit
- Memory unit and storage devices
- Interface unit

Input devices

The data and instructions are typed, submitted, Or transmitted to a computer through input devices. Most common input devices are :

- Keyboard
- Mouse
- Scanner
- Touchpad

Output devices

Output devices display the output results of the operations on the input data or to print the data. Most common output devices are :

- Monitor
- Printer

CPU

CPU can be thought of as the brain of the computer where most of the processing takes place. During processing, it locates and executes the program instructions and fetches data from memory and input/output devices and sends back the computed data. It is divided into following functional units

- **Registers** - These are high-speed storage devices. In most CPUs, some registers are reserved for special purposes.
- **ALU** (Arithmetic logic unit) - part of the CPU that performs arithmetic operations, such as addition and subtraction as well as logical operations, such as comparing two numbers to see if they are the equal or greater or less.
- **CU** (Control unit) - coordinates the processing by controlling the transfer of data and instructions between main memory and the registers in the CPU.

Memory Unit

Components such as the I/O devices and CPU are not sufficient for the working of the computer and hence the storage area is needed to store instructions and data either temporarily or permanently so that subsequent retrieval of the instructions and data can be possible on demand. Data are stored in the memory as binary digits, called bits. Data of various types are encoded as series of bits and stored in consecutive memory locations. Each memory location is comprises of a single byte which is equals to 8 bits and has a unique address so that the contents of the desired memory locations can be accessed independently by referring to its address. A single data item is stored in one or more consecutive bytes of memory. The address of the first byte is used as the address of the entire memory location.

Unit of memory

bit- 0 or 1

8 bits = 1 Byte

1024 Bytes = 1 KiloByte

1024 KiloBytes = 1 MegaByte

1024 MegaBytes = 1 GigaByte

1024 GigaBytes = 1 TeraByte

1024 TeraBytes = 1 PetaByte

1024 PetaByte = 1 ExaByte

1024 ExaByte = 1 ZettaByte

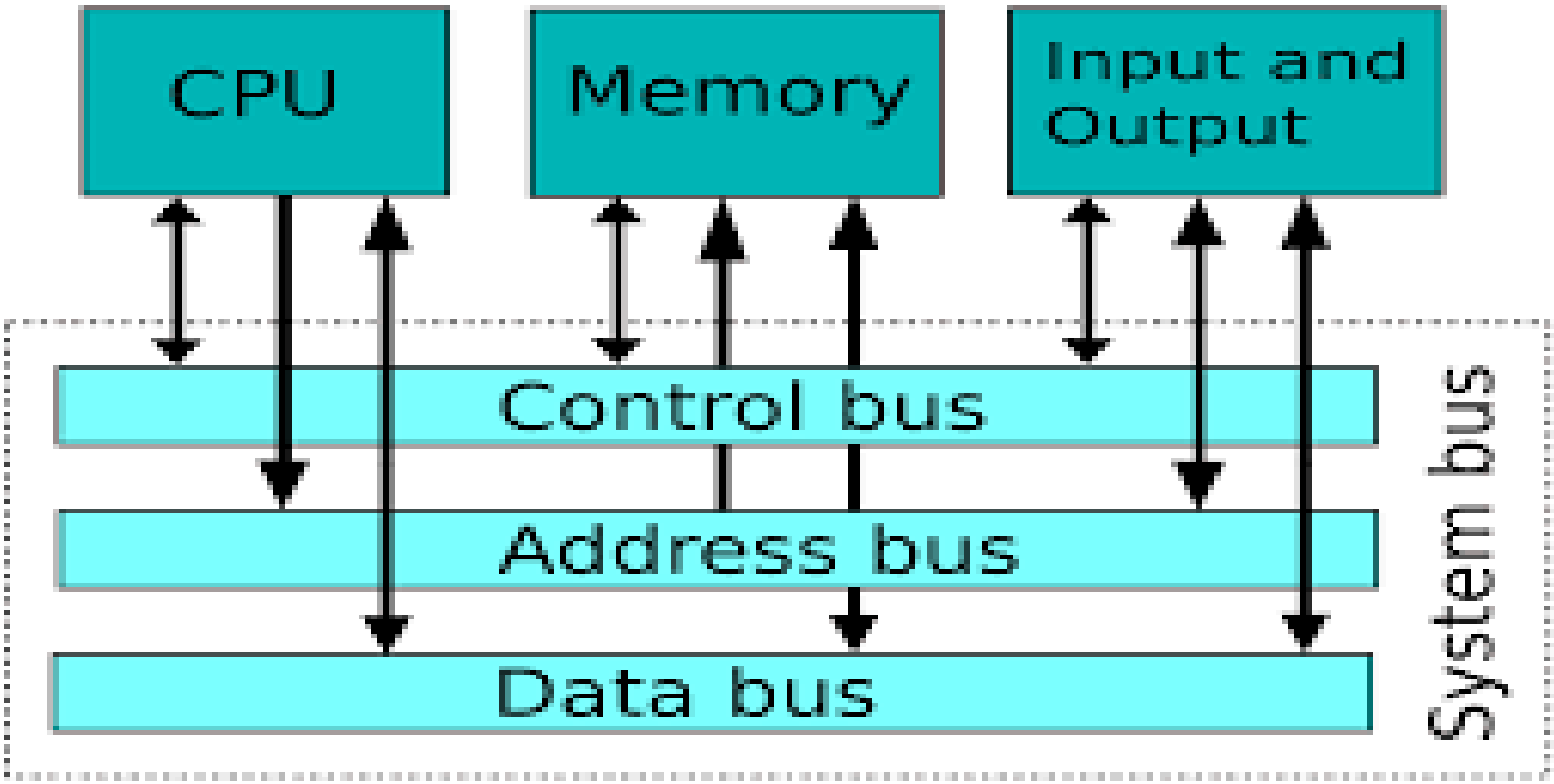
1024 ZettaByte = 1 YottaByte

Interface Unit

The interface unit interconnects the CPU with memory and also with the various input/output (I/O) devices and hence the instructions and data move between the CPU and other hardware components through it. It is a set of parallel wires or lines which connects all the internal computer components to the CPU and main memory. Depending on the type of data transmitted, a bus can be classified into the following three types:

- **Data bus :** The bus used to carry actual data
- **Address bus memory or Input/output device :** Addresses travel via the address bus.
- **Control bus:** This bus carries control information between the CPU and other devices within the computer.

Bus based computer organization



Computer Memory

Primary memory – It is the area where data and programs are stored while the program is being executed along with the data. This is also known as main memory forms the working area of the program and is accessed directly by the processor.

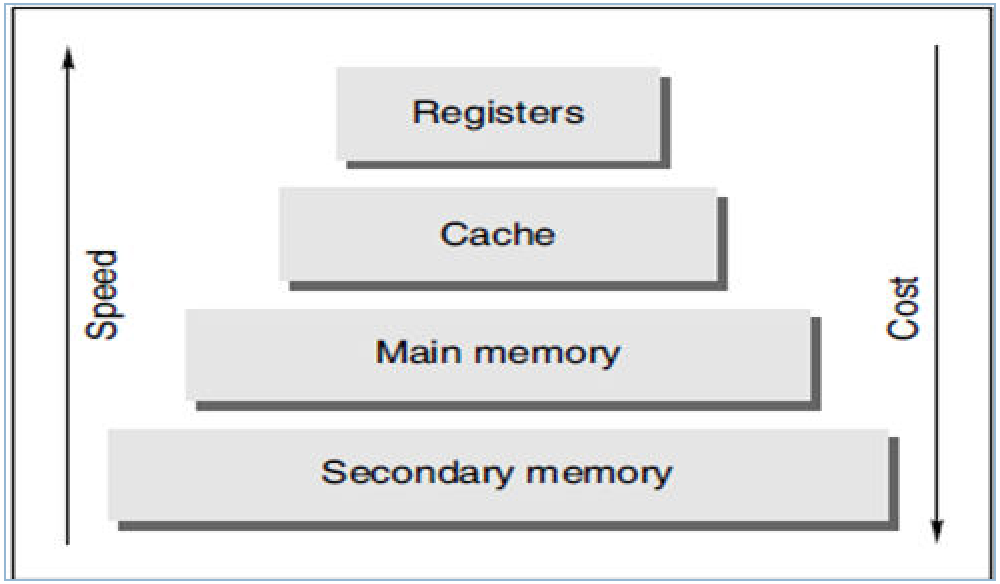
Secondary memory – It is where programs and data are kept on a long-term basis. Common secondary storage devices are the hard disk and optical disks. The hard disk has enormous storage capacity compared to main memory. It is also known as auxiliary memory that stores huge volume of data at a less cost than primary memory devices

Computer Memory cont…

Cache – Cache memory is a small-sized type of volatile computer memory that provides high-speed data access to a processor and stores frequently used computer programs, applications and data. It stores and retains data only until a computer is powered up.

Registers - A special, high-speed storage area within the CPU. All data must be represented in a register before it can be processed. For example, if two numbers are to be multiplied, both numbers must be in registers, and the result is also placed in a register.

Memory hierarchy



Layered view of the Computer

Application Programs

Word-Processors, Spreadsheets,
Database Software, IDEs,
etc…

System Software

Compilers, Interpreters, Preprocessors,
etc.
Operating System, Device Drivers

Machine with all its hardware

Operating System (OS)

Provides several essential services:

- Loading & running application programs
- Allocating memory & processor time
- Providing input & output facilities
- Managing files of information

Programming Languages

Programs are written in programming languages (i.e. PL). A PL is

- A special purpose language
- A set of rules and symbols used to construct a computer program
- A language used to interact with the computer

Computer Languages

- Machine Language
  - Uses binary code
  - Machine-dependent
  - Not portable
- Assembly Language
  - Uses mnemonics
  - Machine-dependent
  - Not usually portable

Low Level Language

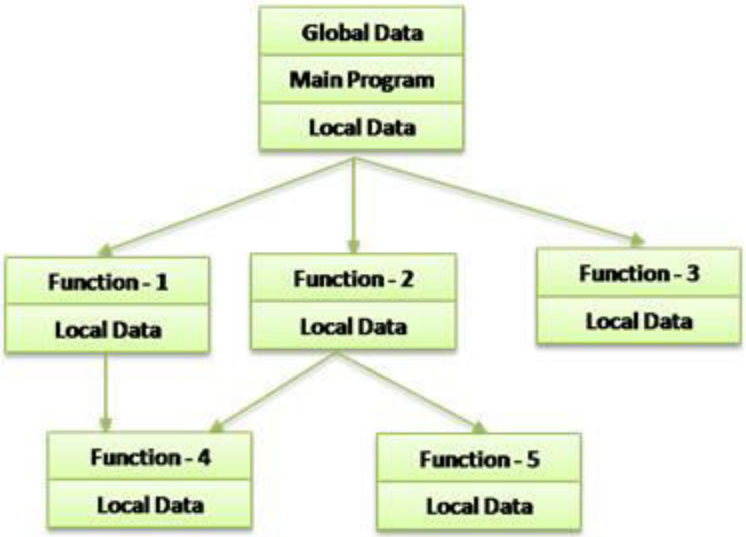
- High Level Language
  - Uses English-like language
  - Machine independent
  - Portable (but must be compiled for different platforms)
  - Examples: Pascal, C, C++, Java, Fortran, . . .

Programming Languages

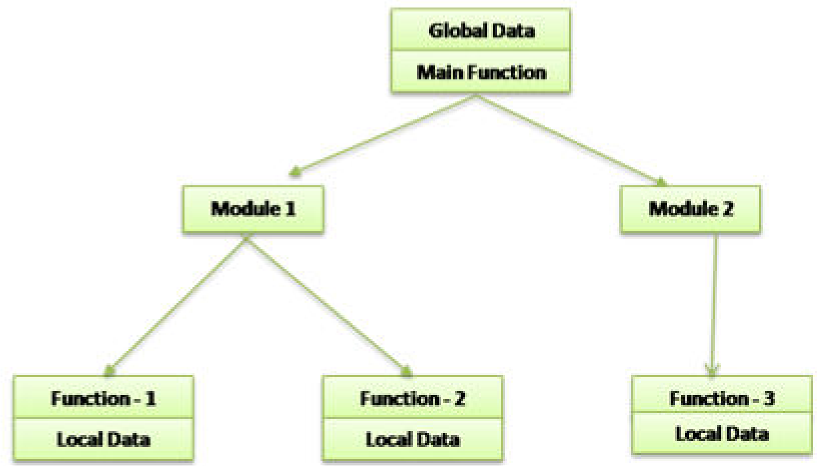
- Procedural Language
  - The given problem is divided in to a number of sub problems depending upon its functionality.
  - The sub problems are called procedures or Methods.
  - Any procedure can be called at any point during the program execution.
  - The program has global and local variables.
  - E.g. Fortran, COBOL

- Structural Language
  - The program is divided into modules and the modules are then divided into functions.
  - The usage of **goto** statement is removed or reduced.
  - Each module can work independent of one another.
  - E.g. Algol, Pascal, C

Programming Languages cont..



Procedural Programming



Structural Programming

Algorithm

A step by step procedure for solving a particular problem is an Algorithm. To be an algorithm, a set of rules must be unambiguous and have a clear stopping point. Algorithms can be expressed in any language, from natural languages like English or French to programming languages like C++, C.

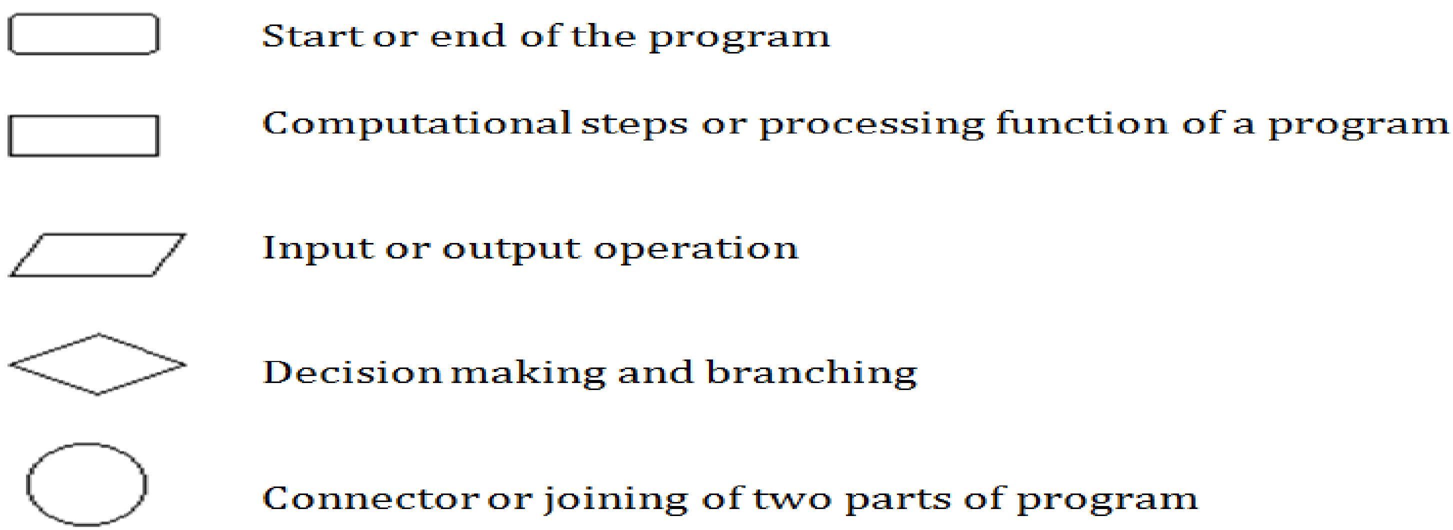
Problem - Design an algorithm to add two numbers and display the average.

Step 1 – START
Step 2 – input x
Step 3 – input y
Step 4 – sum = a + b
Step 5 – average = sum / 2
Step 5 – display average
Step 6 – END

Flowchart

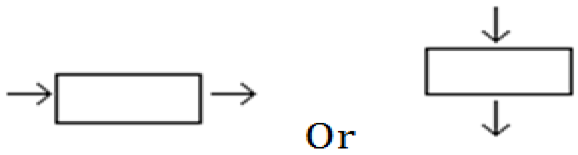
A flowchart is a graphical representation of an algorithm. These flowcharts play a vital role in the programming of a problem and are quite helpful in understanding the logic of complicated and lengthy problems. Once the flowchart is drawn, it becomes easy to write the program in any high level language. Often we see how flowcharts are helpful in explaining the program to others. Hence, it is correct to say that a flowchart is a must for the better documentation of a complex program.

Standard Symbols:



Flowchart Guidelines

- In drawing a proper flowchart, all necessary requirements/steps should be listed out in logical order.
- The flowchart should be clear, neat and easy to follow. There should not be any room for ambiguity in understanding the flowchart.
- The usual direction of the flow of a procedure or system is from left to right or top to bottom
- Only one flow line should come out from a process symbol



- Only one flow line should enter a decision symbol, but two or three flow lines, one for each possible answer, should leave the decision symbol.

Flowchart Guidelines

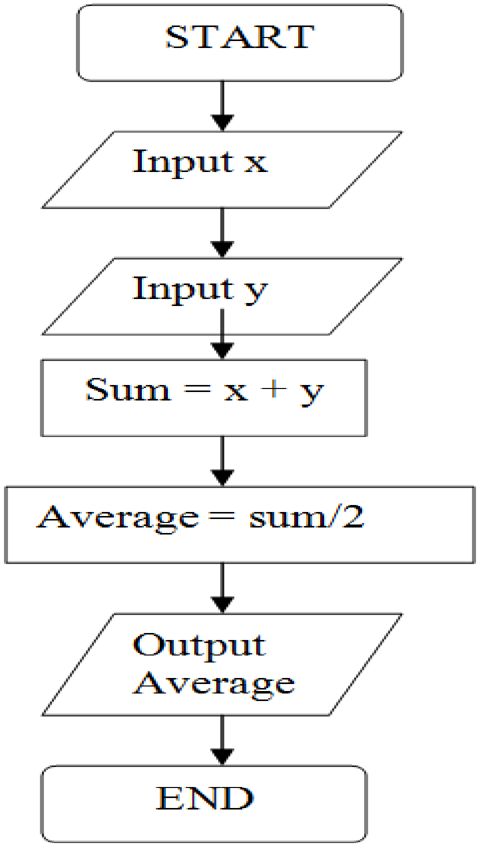
- If the flowchart becomes complex, it is better to use connector symbols to reduce the number of flow lines. Avoid the intersection of flow lines if you want to make it more effective and better way of communication.

- Ensure that the flowchart has a logical start and finish.

- It is useful to test the validity of the flowchart by passing through it with a simple test data.

Flowchart

Below is the flowchart to add two numbers and display the average

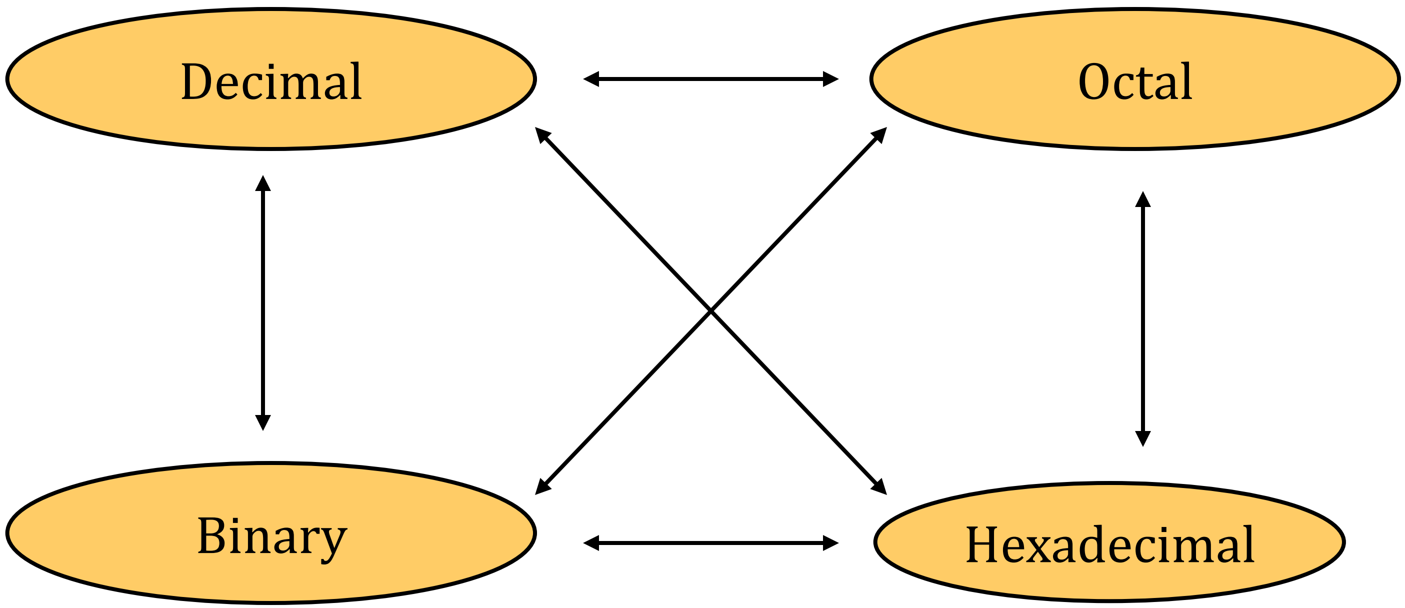


Number System

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, … 7	No	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

Number System Conversion

The possibilities:



Binary to Decimal

Technique:

- Multiply each bit by 2^n, where n is the "weight" of the bit
- The weight is the position of the bit, starting from 0 on the right
- Add the results

101011_2 =>

1	x	2^0	=	1
1	x	2^1	=	2
0	x	2^2	=	0
1	x	2^3	=	8
0	x	2^4	=	0
1	x	2^5	=	32
				43_{10}

Bit "0"

Decimal to Binary

Technique:

- Continuously divide by two to give a result and keep track of the reminder until the final result equals to 0

$125_{10} = ?_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

$125_{10} = 1111101_2$

Thank You

Algorithm and flowchart

Algorithm

- **Step by step procedure to solve a problem**
- **Formally: “An algorithm is an ordered set of unambiguous executable steps, defining a terminating process.”**

Characteristics of algorithm

- Finiteness – terminates after a finite number of steps
- Definiteness – rigorously and unambiguously specified
- Input – valid inputs are clearly specified
- Output – can be proved to produce the correct output given a valid input
- Effectiveness – steps are sufficiently simple and basic

Representation of Algorithms

- A single algorithm can be represented in many ways:
  - Formulas: $F = (9/5)C + 32$
  - Words: Multiply the Celsius by 9/5 and add 32.
  - Flow Charts.
- In each case, the algorithm stays the same;
- the implementation differs!

Example:

Design an algorithm for adding the test scores as given below:

26, 49, 98

<u>Algorithm in words:</u>

Step 1: Start

Step 2: Get three numbers26,49,98

Step 3: Add three numbers and store the result in sum

Step 4: Display sum

Step 5: Stop

Algorithm:

- Is a planned set of steps to solve certain problem

Ex.:

- **Assume for instance that the gross pay of an employee is to be calculated and then 10 percent of the gross is to remove as tax. The yield being the net pay. Write down the algorithm for this problem.**

Sol.:

1. Begin
2. input name, hours_worked, and wage par hour
3. calculate gross_pay = hours_worked * wage par hour
4. calculate tax = (10/100) * gross_pay
5. calculate net_pay = gross_pay – tax
6. print name, net_pay
7. End

Algorithm:

Ex.:

- **Write an algorithm to read the name and the mark (one) for one student then add to his mark 5 marks**

Sol.:

1. Begin
2. input name, mark
3. new_mark = mark+5
4. print name, new_mark
5. End

Pseudo code:

- English-like description of each step of algorithm
- Not computer code
- pseudo code is written in a format that is closely related to high level programming language structures.
- But pseudo code does not use specific programming language syntax

Example

To find sum of 3 numbers:

Start

Use variables: sum, number1, number2, number3 of type integer

Accept number1, number2, number3

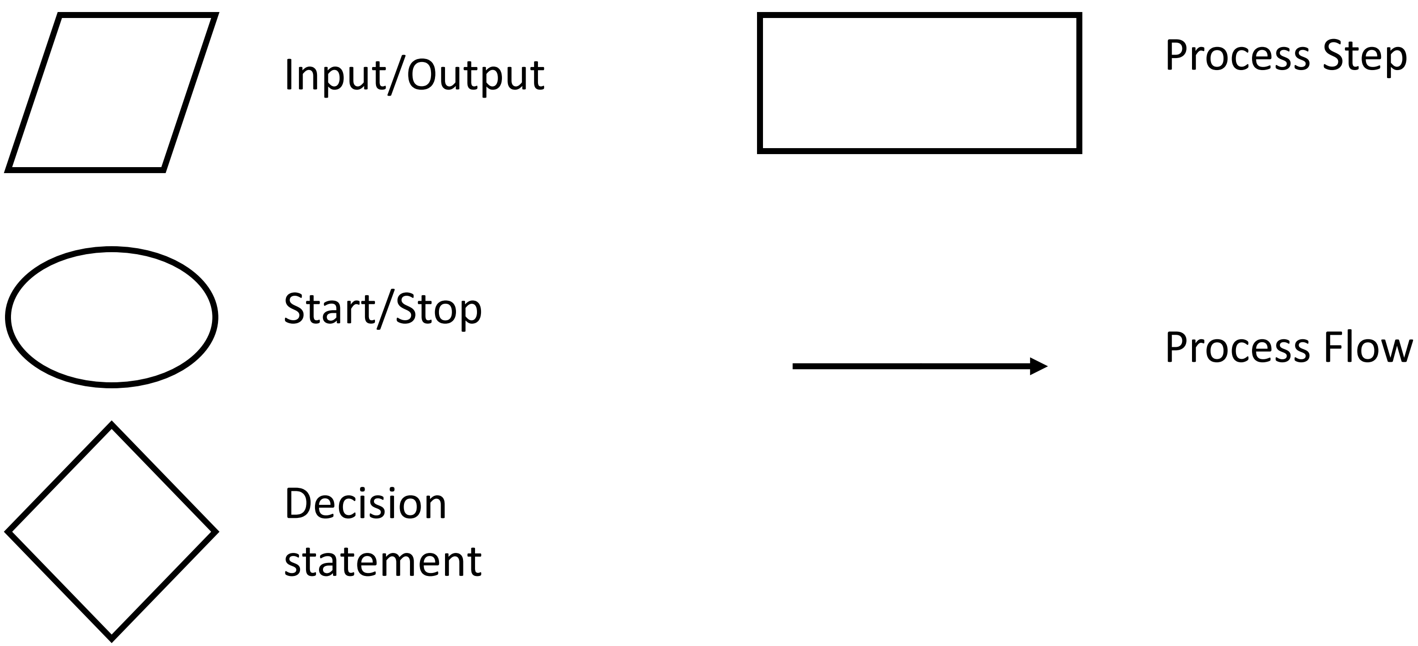
Sum = number1 + number2 + number3

Print sum

Stop

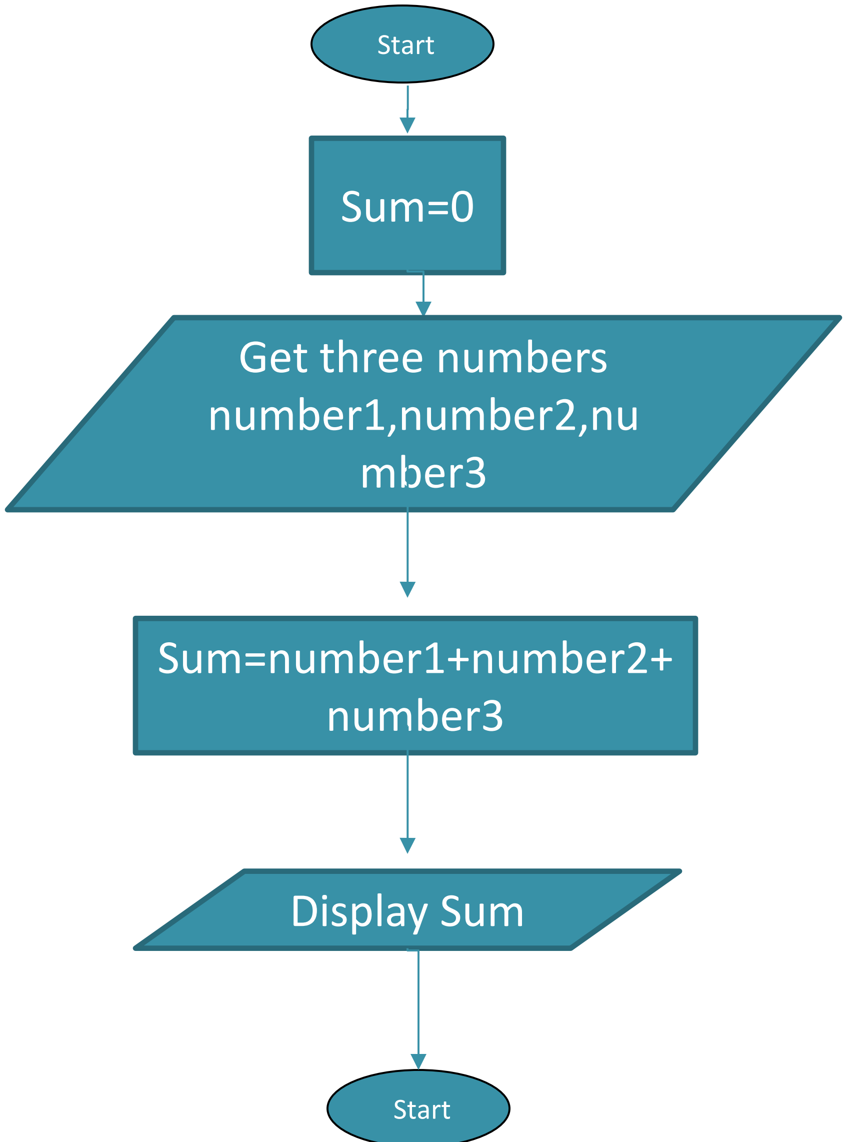
Flow chart

- Graphical description of algorithm
- Standard symbols used for specific operations



Example:

Flowchart for adding three numbers



Activity 1

- Design an algorithm and the corresponding flowchart for adding the test scores as given below:

  26, 49, 98, 87, 62, 75

- Design an algorithm and the corresponding flowchart for finding out average of six numbers.

- Design an algorithm and flowchart for finding out area of a rectangle

Number Systems & Binary Arithmetic

CHAPTER 2

Overview

- Explain the number system used in computers :
- Learn about the digit symbols, base, and representation forms of various number systems developed and used
- Explain the method of number system conversions
- Add and subtract unsigned binary numbers
- Differentiate signed magnitude, 1's complement, and 2's complement representation of binary numbers
- Subtract signed numbers in 1's complement and 2's complement representation
- Explain the technique of multiplication and division of binary numbers
- Explain binary codes and their classification

Introduction

- A number system defines a set of values used to represent quantity.
- For example, the number of mobile phones kept in a shop, the number of persons standing in a queue, and the number of students attending a class.
- There are many ways to represent the same numeric value.
- Long ago, humans used sticks to count; they then learned how to draw pictures of sticks on the ground and eventually on paper. So, the number 5 was first represented as: | | | | | (for five sticks).

Key Words

- **Symbol :** It is any graphic character.
- **Number system:** It is a set of symbols that represents a set of quantitative values.
- **Base:** It represents the number of digit symbols in a number system.
- **Radix:** It represents the number of digit symbols in a number system.
- **Decimal number system:** It is a number system that has ten symbols 0,1,……….,9 which represents values.
- **Binary number system:** It is a number system that uses two symbols 0 and 1 to represent zero and one respectively.
- **Octal number system :** It is a number system that has eight symbols 0,1,……….,7 which represents values zero to seven respectively.
- **Hexadecimal number system :** A number system that has eight symbols 0,1,……….,F which represents values zero to sixteen respectively.

Key Words

- **Carry :** In a number system, when the result of addition of two single digit numbers is greater than the largest representable number symbol, a carry is said to be generated. This carry is placed in the next left column.
- **Borrow :** In a number system, when a larger single digit number is subtracted from a smaller single digit, a borrow is generated.
- **Signed number :** A binary number in which the most significant bit represents the sign of the number and the rest the magnitude of the number.
- **1's complement :** A number system that was used in some computers to represent negative numbers. To form 1s complement of a number, each bit of the number is inverted which means zeros are replaced with ones and ones with zero.

Key Words

- **2's complement :** A number formed by adding 1 to the 1's compliment of a number.
  - The 2's complement representation has become the standard method of storing signed binary integers.
  - It allows the representation of an n-bit number in the range – 2n to 2n-1, and has the significant advantage of only having one encoding for 0.
- **Bit :** It is a binary digit.
- **BCD number :** Binary Coded Decimal (BCD) number is a number in which each of the digits of an unsigned decimal number is represented by the corresponding 4-bit binary equivalents.
- **ASCII code :** American Standard Code for Information Interchange is a 7-bit binary code formed to represent decimal numbers 0 to 9, alphabetic characters a to b(also A TO B), and special characters like ;, :, NUL, etc. for handling these characters in the digital computer and to also use this binary code for exchanging data between digital computers connected in a networked environment.

Base of a Number System

- The *base, or radix, of any number system is determined by* the number of digit symbols in the system.
- For example, binary is a base-2 number system since it uses two symbols and decimal is a base-10 system since it uses ten symbols.
- In general, in any number system, a number *N can be* represented by any one of the following forms:

Base of a Number System

(a) Positional notation form:

$$N = d_{n-1}\, d_{n-2} \ldots d_1\, d_0 \cdot d_{-1}\, d_{-2} \ldots d_{-m}$$

(b) Polynomial form:

$$N = d_{n-1} \times r^{n-1} + d_{n-2} \times r^{n-2} \ldots d_0 r^0$$
$$+ d_{-1} r^{-1} + d_{-2} r^{-2} \ldots d_{-m} r^{-m}$$

(c) Compact form:

$$N = \sum_{i=-m}^{n-1} d_i r^i$$

where

d = value of the digit symbol,

r = base or radix,

n = the number of integral digits to the left of the decimal point, and

m = the number of fractional digits or digits to the right of the decimal point.

Weighting Factor

- The numerical value of a number is the sum of the products obtained by multiplying each digit by the weight of its respective position.
- Decimal numbers are represented by arranging the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.
- These are known as decimal digits, in various sequences.
- The position of each digit in a sequence has a certain numerical weight, and each digit is a multiplier of the weight of its position.
- The decimal number system is hence an example of a weighted, positional number system.
- The weight of each position is a power of the base number 10.

Weighting Factor

- ***Important note :*** *Any number raised to the power of zero is* 1, even zero raised to the power of zero is 1, for example, 100 = 1, 00 = 1, *x0* = *1*

Let us consider the number 754 in the decimal number system.

$$7 \cdot \boxed{10}^{2} + 5 \cdot 10^{1} + 4 \cdot 10^{\boxed{0}} = 700 + 50 + 4 = 754$$

Base

Digit Position

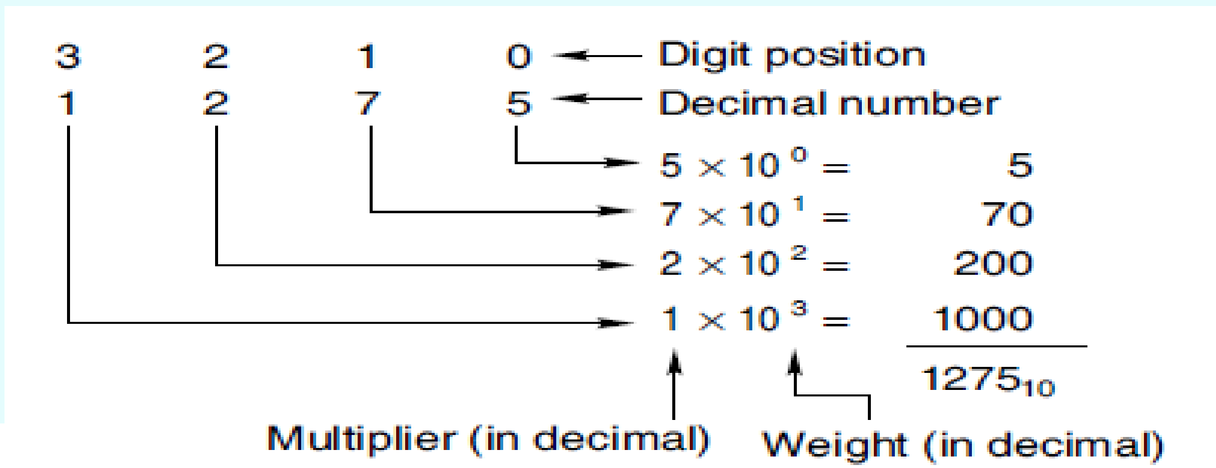
Types of Number Systems

Table 2.1 Number systems, bases, and symbols

Number system	Base	Digital symbols
Binary	2	0, 1
Ternary	3	0, 1, 2
Quaternary	4	0, 1, 2, 3
Quinary	5	0, 1, 2, 3, 4
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Duodecimal	12	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
Vigesimal	20	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, I, J

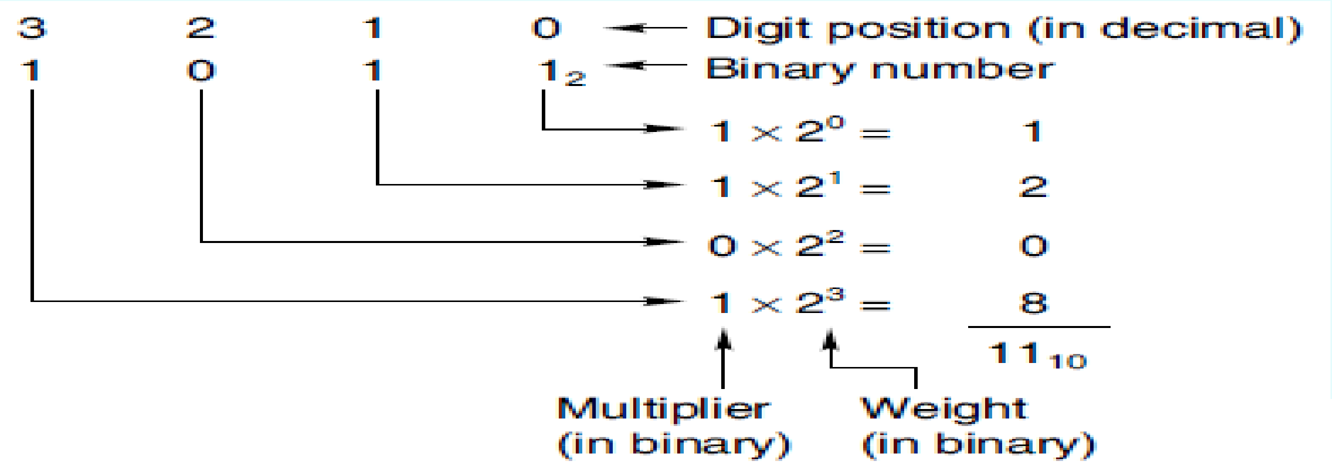
Decimal Number System [Base-10]

- Most people today use decimal representation to count.
  - This number system uses TEN different symbols to represent values.
  - In the decimal system there are 10 digit symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 with 0 having the least value and 9 having the greatest value.
  - For a number represented in decimal system, the digit on the extreme left has the greatest value, whereas the digit on the extreme right has the least value.
  - Each position to the left increases by a weight of 10.



Binary Number System [Base-2]

- The binary number system uses TWO symbols to represent numerical values.
  - These are 0 and 1 with 0 having the least value and 1 having the greatest value.
  - Number representation in the binary system is similar to that in the decimal system, in which the digit on the extreme left represents the greatest value and is called the most significant digit (MSD), whereas the digit on the extreme right is known as the least significant digit (LSD).
  - Each position to the left increases by a weight of 2.

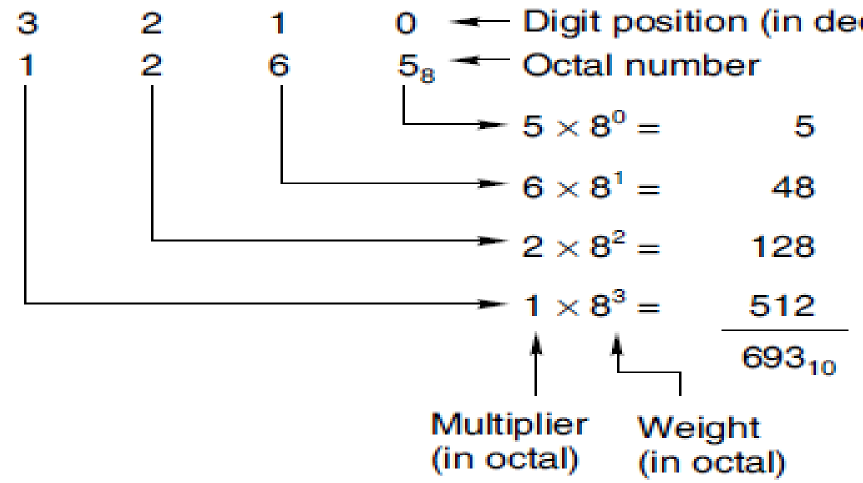


Octal Number System [Base-8]

- The octal number system uses EIGHT digit symbols to represent numbers. The symbols are 0, 1, 2, 3, 4, 5, 6, and 7 with 0 having the least value and 7 having the greatest value.
- The number representation in the octal system is done in the same way as in the decimal system, in which the digit on the extreme left represents the most significant digit.
- Each position to the left increases by a weight of 8.

(i) $7 + 1 = 10$ (seven plus one equals *zero, carry one* or 10)

(ii)



Hexadecimal Number System[Base-16]

- The hexadecimal number system uses SIXTEEN symbols to represent numbers.
  - The symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F with 0 having the least value and F having the greatest value.
  - Number representation in hexadecimal system is done in the same way as in the decimal system, in which the symbol on the extreme left represents the most significant digit.
  - Each position to the left increases by a weight of 16.

(i) F + 1 = 10 (F 'i.e. 15' plus one equals *zero, carry one* or 10)

(ii)

2 1 0 ← Digit position (in decimal)

2 A 6_{16} ← Hexadecimal number

$6 \times 16^0 = 6$

$10 \times 16^1 = 160$

$2 \times 16^2 = 512$

678_{10}

Table 2.2 Number systems equivalency table

Decimal (Base-10)	Binary (Base-2)	Octal (Base-8)	Hexadecimal (Base-16)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Common Rules of Number Systems

- *All number systems follow the following set of rules:*

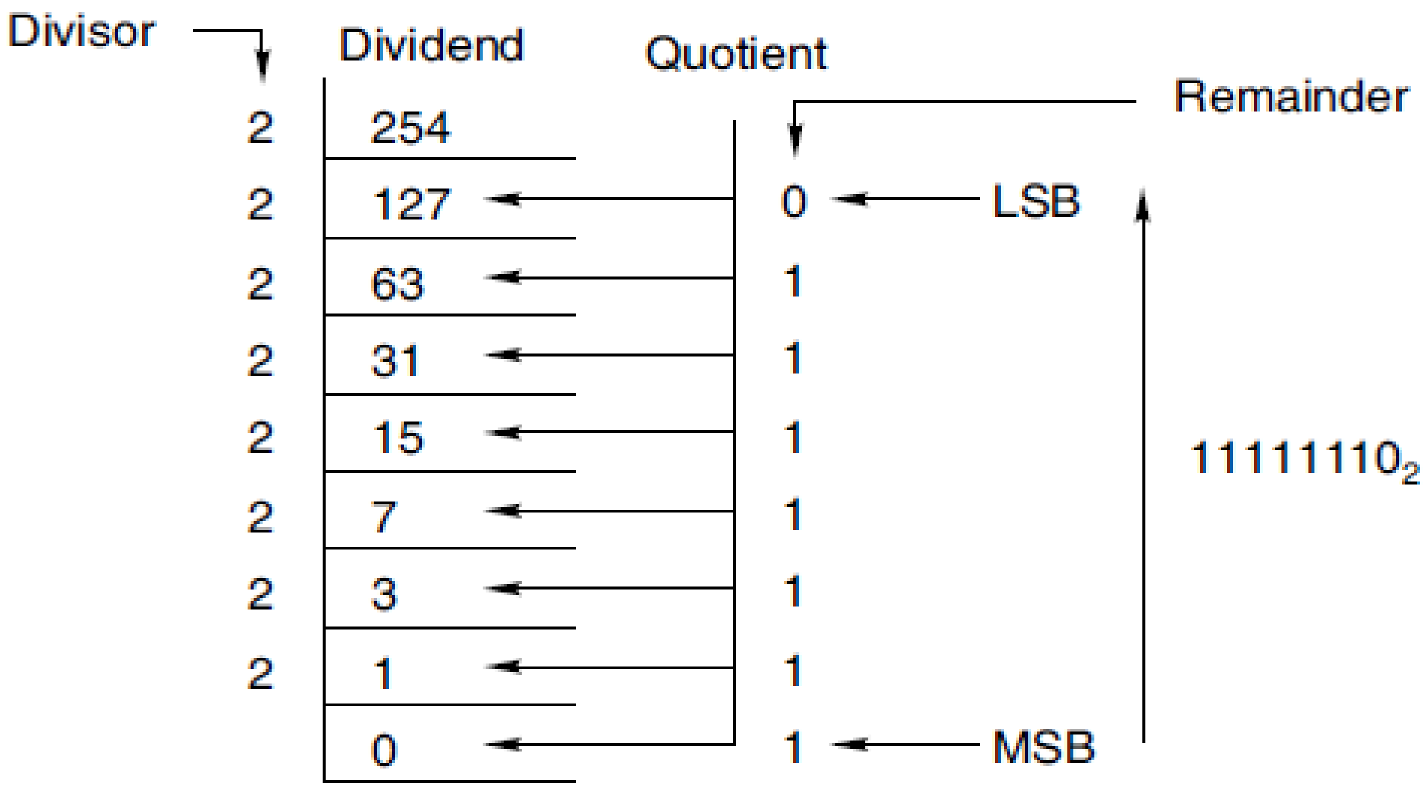
- *Rule 1 : The number of digit symbols available in a* number system is equal to the base.
- *Rule 2 : The value of the largest digit symbol is one less* than the base.
- *Rule 3 : Each position multiplies the value of the digit* symbol by the base raised to the power of the value equal to the digit symbol position.
- *Rule 4 : A carry from one position to the next increases* its weight base times.

Conversion of a Decimal number to its Binary equivalent

- **Method 1 Repeated-division-by-2 method**
  1. Divide the dividend, that is, the decimal number by two and obtain the quotient and remainder.
  2. Divide the quotient by two and obtain the new quotient and remainder.
  3. Repeat step 2 until the quotient is equal to zero (0).
  4. The first remainder produced is the least significant bit (LSB) in the binary number and the last remainder is the most significant bit (MSB). Accordingly, the binary number is then written (from left to right) with the MSB occurring first (list the remainder values in reverse order). This is the binary equivalent.

Example

Converting the decimal number 254 into its binary equivalent.



Thus, the binary equivalent is **11111110.**

Example

❑ Method 2 Power-of-2-subtraction method

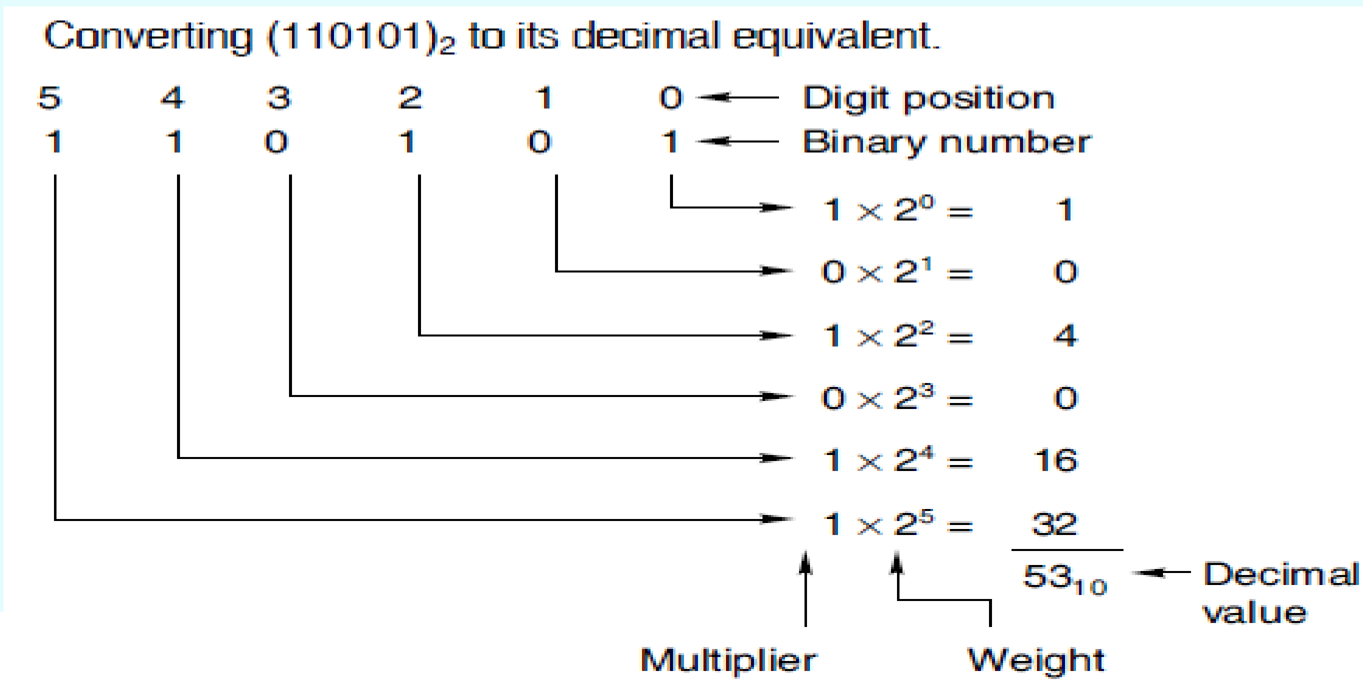
1. Let *D be the number that has to be converted from* decimal to binary.
2. (a) Find the largest power of two that is less than or equal to *D. Let this equal P.*

   (b) If |D| ≥ P, subtract *P from D, obtain a result which is* a decimal number. Put 1 in the digit position where the weighting factor is *P.*

   (c) Otherwise, if |D| < |P|, put 0 in the corresponding weighting factor column.
3. Repeat step 2 with *D = remainder decimal number until* D = 0, or |D| < |P|.

Conversion from Binary to Decimal

- To express the value of a given binary number as its decimal equivalent, sum the binary digits after each digit has been multiplied by its associated weight.
- Example :



Conversion of a Decimal number to its Octal Equivalent

- To convert from decimal whole numbers to octal, the systematic approach called the repeated-division-by-8 method is used.
- Example:

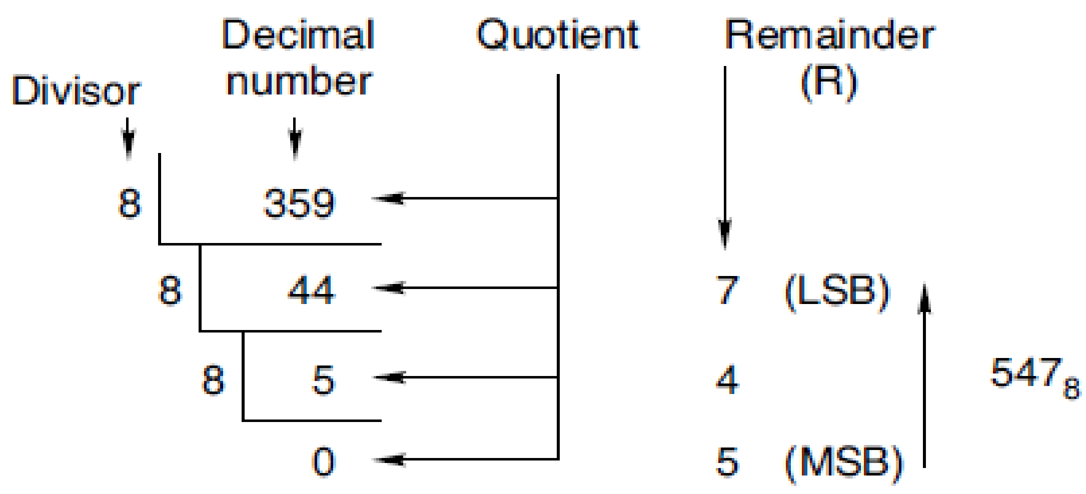
Converting (359)10 to octal.

a) Divide the decimal number by eight and obtain a quotient and a remainder.
b) Divide the quotient by eight and obtain a new quotient and a remainder.

Conversion of a Decimal number to its Octal Equivalent

(c) Repeat step (b) until the quotient is equal to zero (0).

(d) The first remainder produced is the LSB in the octal number and the last remainder (*R) is the MSB. Accordingly, the octal number* is then written (from left to right) with the MSB occurring first.

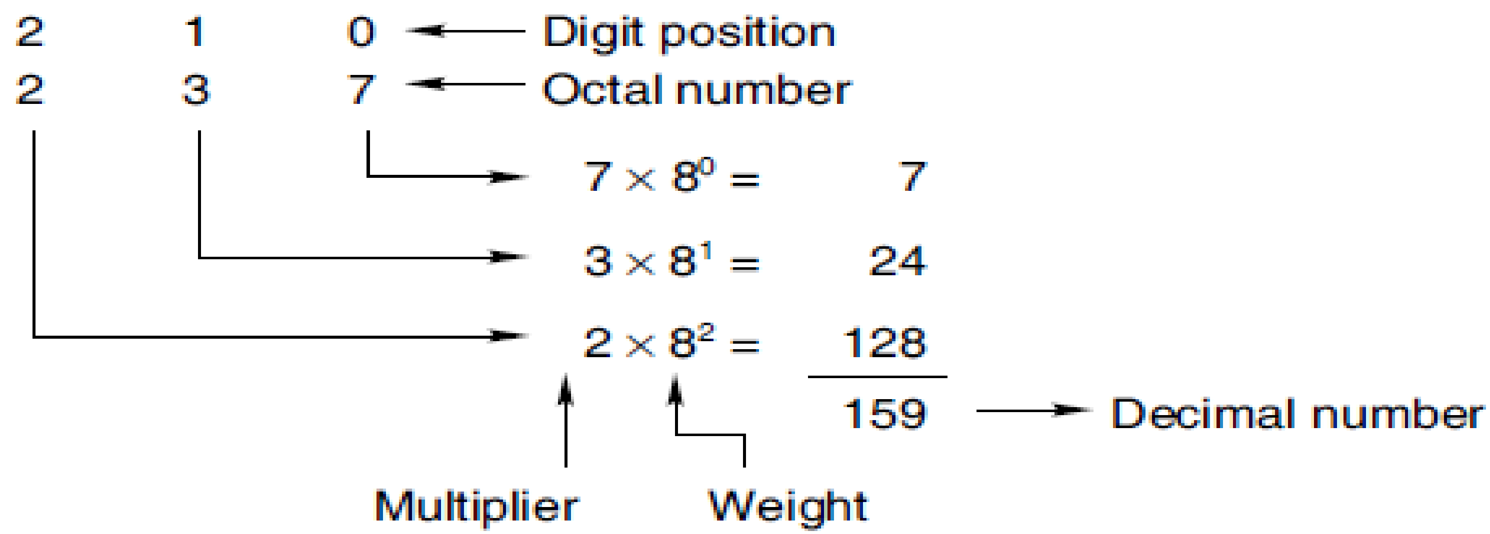


Therefore, $(359)_{10} = (547)_8$

Conversion of an Octal Number to its Decimal Equivalent

- To express the value of a given octal number as its decimal equivalent, add the octal digits after each digit has been multiplied by its associated weight.

Converting $(237)_8$ to decimal form.



Conversion of an Octal Number to its Binary Equivalent

- Since each octal digit can be represented by a three-bit binary number (see Table), it is very easy to convert from octal to binary. Simply replace each octal digit with the appropriate three-bit binary number as indicated in the following example.

Table Binary equivalents for octal digits

Octal digit	Equivalent binary number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Conversion of an Octal Number to its Binary Equivalent

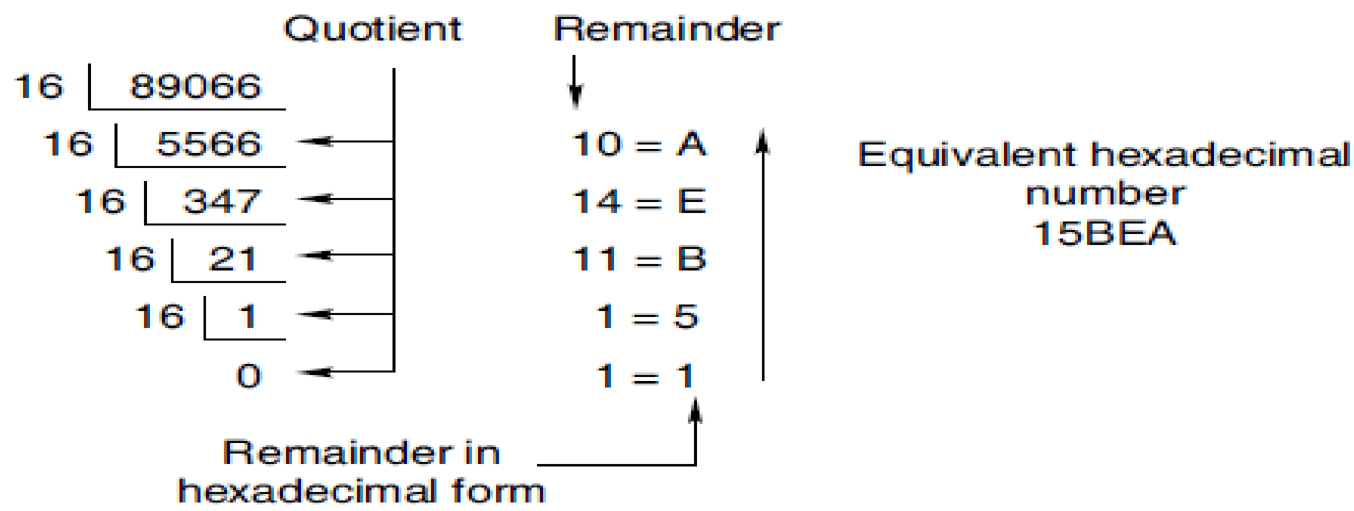
- Converting a binary number to an octal number is a simple process.
- Break the binary digits into groups of three starting from the binary point and convert each group into its appropriate octal digit.
- For whole numbers, it may be necessary to add zeros as the MSB, in order to complete a grouping of three bits. Note that this does not change the value of the binary number.

Converting $(010111)_2$ to its equivalent octal number.

Binary number	$\longrightarrow$	$\underbrace{010}$	$\underbrace{111}$
Equivalent octal number	$\longrightarrow$	2	7

Conversion of a Decimal Number to its Hexadecimal Equivalent

- The decimal number to be converted is divided by 16 until the quotient is 0. The hexadecimal number is formed from the remainders.



Starting with the last remainder, convert the remainders into hexadecimal numbers:

1 5 11 14 10 = 15BEA = the hexadecimal number.

Conversion of a Hexadecimal Number to its Decimal Equivalent

- To convert a hexadecimal to decimal, begin by multiplying each of the hexadecimal digits by their positional weight values as expressed in decimal.

- Then the resulting values are added to obtain the value of the decimal number.

Conversion of a Hexadecimal Number to its Decimal Equivalent

Converting the hexadecimal number **A4D31** to its equivalent decimal number.

The decimal value of each digit in relation to its positional weight value is evaluated first:

4	3	2	1	0	← Digit position
A	4	D	3	1_{16}	← Hexadecimal number

Multiplier in hexadecimal × Positional weight value (in decimal)	Multiplier in decimal × Positional weight value		Final decimal value
$1 \times 16^0 =$	1×1	$=$	1
$3 \times 16^1 =$	3×16	$=$	48
$D \times 16^2 =$	13×256	$=$	3328
$4 \times 16^3 =$	4×4096	$=$	16384
$A \times 16^4 =$	10×65536	$=$	655360
			675121_{10}

Multiplier in hexadecimal

Positional weight value (in decimal)

Multiplier in decimal

Final decimal value

Conversion of a Hexadecimal Number to its Binary Equivalent

- As each hexadecimal digit can be represented by a four-bit binary number (see Table 2.4), it is very easy to convert from hexadecimal to binary.
- Simply replace each hexadecimal digit with the appropriate four-bit binary number as indicated in the following examples.

Table Number systems equivalency

Decimal (Base-10)	Binary (Base-2)	Hexadecimal (Base-16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversion of a binary number to its hexadecimal equivalent

- Hexadecimal system works very much like the octal system, except that each digit needs exactly four bits to represent it.
- This means the binary number has to be divided into groups of four digits, again starting from the digit at the extreme right.
- The equivalent hexadecimals for each set of four digits are then written.
- For whole numbers, it may be necessary to add zeroes as the MSB in order to complete a grouping of four bits.

Conversion of a binary number to its hexadecimal equivalent

- Note that this addition does not change the value of the binary number.

Converting the binary number 1111011101101011011 to its equivalent hexadecimal number.

The conversion is done as follows:

A leading zero had to be added for the most significant group to have four bits.

Binary Numbar	⟶	0111	1011	1011	0101	1011
Equivalent hexadecimal numbar	⟶	7	B	B	5	B

Conversion from Hexadecimal to Octal, & Octal to Hexadecimal

- To convert from hexadecimal to octal, each digit of the hexadecimal number is written as its equivalent four-bit binary number.

- The resulting binary number is divided into groups of three binary digits. Then corresponding octal numbers for each of these groups are written.

Example

Converting the hexadecimal number AF35D02 to its equivalent octal number.

The given number is rewritten by replacing the hexadecimal digits by their equivalent four-bit binary numbers.

	In groups of three bits
A	1010
F	1111
3	0011
5	0101
d	1101
0	0000
2	0010

The binary number is regrouped as three-bit binary numbers that are replaced with octal symbols.

Example

The binary number is regrouped as three-bit binary numbers that are replaced with octal symbols.

In groups of three bits	Octal number
001	1
010	2
111	7
100	4
110	6
101	5
110	6
100	4
000	0
010	2

Therefore, we see that $AF35D02_{16} = 1274656402_8$

Working with Fractional Numbers

- One is familiar with the decimal (base-10) number system.
  - Each digit within any given decimal number is associated with a weight. Furthermore, the value of that number is the sum of the digits after each has been multiplied by its weight.
  - To illustrate, let us consider Table and assume that the number 654.52, written as (654.52)10 to specify base-10, is being represented.
  - Note that the digits range from 0 to 9.
  - Just as the decimal system with its ten digits is a base- 10 system, the binary number system with its two digits, 0 and 1, is a base-2 system.

Table Decimal number system

	Hundreds	Tens	Units	One-tenth	One-hundredth	
Weights	10^2	10^1	10^0	10^{-1}	10^{-2}	
Symbols	6	5	4	5	2	
Weighted value	600	50	4	0.5	0.02	Total 654.52

Working with Fractional Numbers

- Table shows the weighting for the binary number system up to two decimal places after and three places before the *binary point (.) Note the similarity* with the decimal system.
  - The least significant bit (LSB) is the rightmost binary digit, which has the lowest binary weight of a given number.
  - The most significant bit (MSB) is the leftmost binary digit, which has the highest binary weight of a given number.
  - Counting in binary is similar to the decimal number system.
  - The LSB begins with zero (0) and is incremented until the maximum digit value is reached. The adjacent bit positions are then filled appropriately as the iterative counting process continues.

Table Binary weights

Weights	2^2	2^1	2^0	2^{-1}	2^{-2}

Conversion from Decimal Fractions to Binary

- When converting a fractional decimal value to binary, a slightly different approach is needed. Instead of dividing by 2, the decimal fraction is multiplied by 2. If the result is greater than or equal to 1, then 1 is to be put as the quotient.
- If the result is less than 1, then 0 is put as the quotient.

Converting $(0.375)_{10}$ to binary.

$0.375 \times 2 = 0.750$	0
$0.750 \times 2 = 1.500$	1
$0.500 \times 2 = 1.000$	1

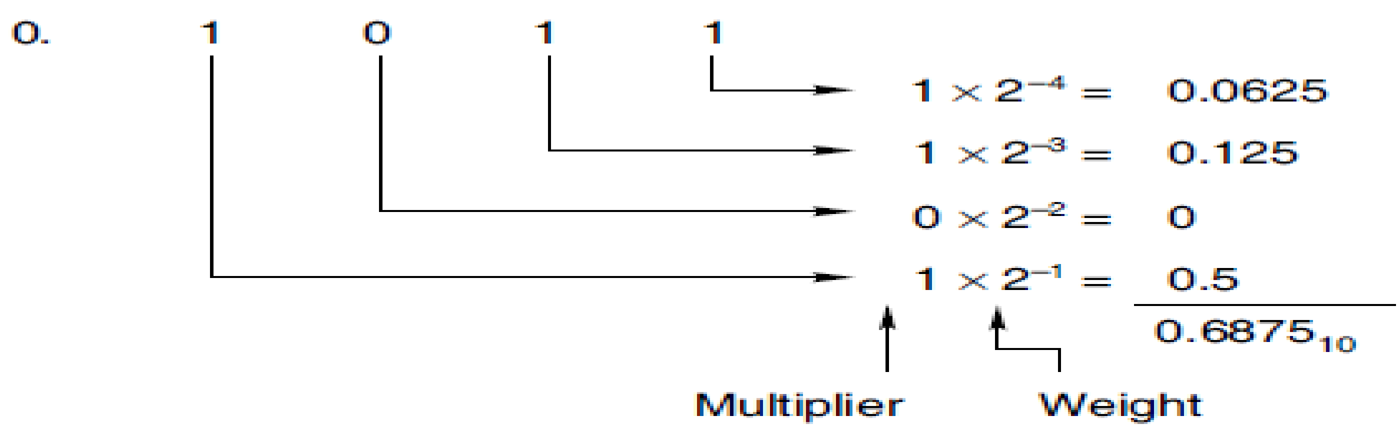
done.

Note that the last operation is complete when the fraction part equals zero. It is rarely possible to accurately represent a fractional value in binary. The answer to this problem is: .011

Conversion from Binary Fraction to Decimal

- To express the value of a given fractional binary number in equivalent decimal value, each bit is multiplied by its associated weight and the summation of these gives the desired decimal number.
- Example :

Converting $(0.1011)_2$ to a decimal number.



Conversion from Octal Fraction to Decimal

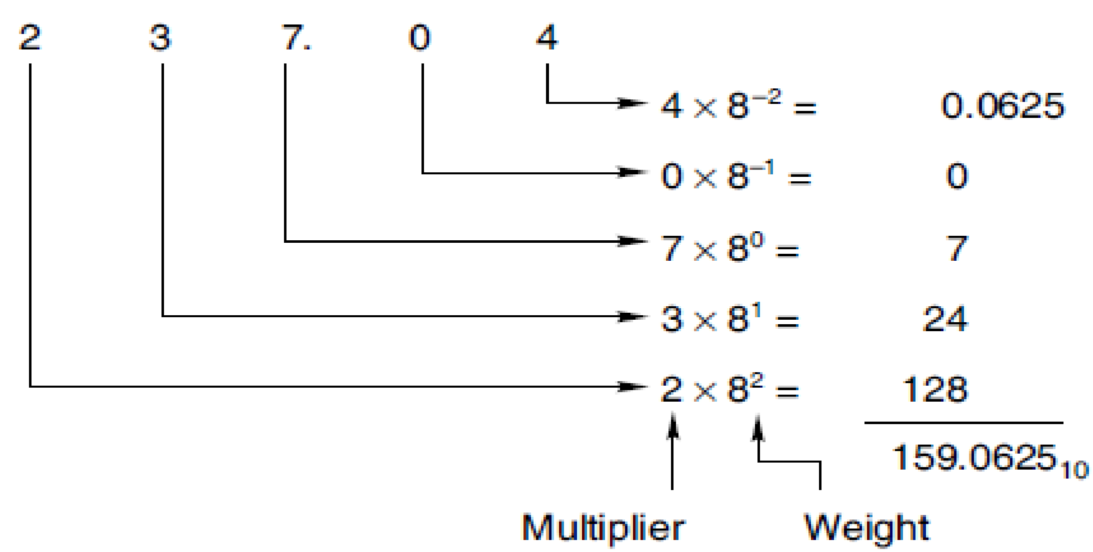
- Just as the decimal system with its ten digit symbols is a number system with base 10, the octal number system with its 8 digit symbols, '0', '1', '2', '3', '4', '5', '6' and '7', has eight as its base.
- Table shows the weights for an octal number that has three decimal places before and two digit places after the *octal point (.)*.
- Octal weights:

Weights	8^2	8^1	8^0	8^{-1}	8^{-2}

Conversion from Octal Fraction to Decimal

- To express a given octal number as its decimal equivalent, each bit is multiplied by its associated weight and the summation of these gives the decimal number.

Converting $(237.04)_8$ to a decimal number.



Conversion from Decimal Fractions to Octal

- The techniques used to convert decimal fractions to octal are similar to the methods demonstrated previously to convert decimal fractions to binary numbers.
  - The **repeated multiplication- by-8 method is used. In the multiplication by-** 8 method, the fraction is repeatedly multiplied by eight,
  - The integer number is recorded until the fraction part is zero. The first integer produced is the MSD, while the last integer is the LSD.

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

Conversion from Binary Fraction to Octal

- Converting binary to octal is also a simple process.
- Arrange the binary number into groups of three bits starting from the binary point and convert each of these groups into its appropriate octal digit symbol.
- For whole numbers, it may be necessary to add a zero with the MSD in order to form a grouping of three bits. Note that this does not change the value of the binary number.
- Similarly, when representing fractions, it may be necessary to add a trailing zero in the LSD in order to form a complete grouping of three bits.

Example

(a) Converting $(010.111)_2$ to octal

$$\text{Binary number} \longrightarrow \underbrace{0 \quad 1 \quad 0}_{2} \quad . \quad \underbrace{1 \quad 1 \quad 1}_{7}$$

$$\text{Octal number} \longrightarrow 2 \quad . \quad 7$$

Thus, $(010.111)_2 = (2.7)_8$

(b) Converting $(0.110101)_2$ to octal

$$\text{Binary number} \longrightarrow \underbrace{0 \quad 0 \quad 0}_{0} \quad . \quad \underbrace{1 \quad 1 \quad 0}_{6} \quad \underbrace{1 \quad 0 \quad 1}_{5}$$

$$\text{Octal number} \longrightarrow 0 \quad . \quad 6 \quad 5$$

Thus, $(0.110101)_2 = (0.65)_8$

Hexadecimal Number Conversion

- Just like the octal number system, the hexadecimal number system provides a convenient way to express binary numbers.
- Table shows the weight for the hexadecimal number system up to three digit places before and two places after the *hexadecimal point.*
- *Based on the trend in previous number* systems, the methods used to convert hexadecimal to decimal and vice versa should be similar

Weights	16^2	16^1	16^0	16^{-1}	16^{-2}

Hexadecimal Number Conversion

- Table lists the equivalent decimal, binary, and hexadecimal representations for the decimal numbers ranging from 0 to 15.
- Each hexadecimal number can be represented as a four-digit binary number.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B

Converting $(37.12)_{16}$ to binary number.

Hexadecimal number →	3	7	.	1	2
Binary number →	0011	0111	.	0001	0010

Conversion from Binary Fraction to Hexadecimal

- Converting from binary to hexadecimal is also a simple process.
  - Arrange the binary digits into groups of four starting from the binary point and convert each group into its appropriate hexadecimal digit symbol.
  - For whole numbers, it may be necessary to add a zero with the MSD in order to form a grouping of four bits. Note that this addition does not change the value of the binary number.
  - Similarly, while representing fractions, it may be necessary to add a trailing zero in the LSD in order to form a grouping of four bits.

Converting $(0.00111111)_2$ to hexadecimal

Binary number	⟶	0	.	0011	1111
Hexadecimal number	⟶	0	.	3	F

Thus, $(0.00111111)_2 = (0.3F)_{16}$

Binary Arithmetic

- In computers, numbers are represented in binary form. The arithmetic operations performed by a computer therefore involves binary numbers.
- The next few sections describe how binary arithmetic operations like binary addition, subtraction, multiplication, and division are performed.
- In this context, it may be mentioned that such arithmetic operations are primarily performed by the ALU within the computer system.
    - **Addition**
    - **Subtraction**
    - **Binary**

Addition

- Four basic rules are needed to perform binary addition. Table lists these rules. The first three rules are quite simple and there is no difference between these binary rules and the corresponding decimal rules. The fourth rule, however, is different from the decimal rule.

Table Rules for binary addition

Rule 1	Rule 2	Rule 3	Rule 4
0	0	1	1
+0	+1	+0	+1
0	1	1	10

1. Apply Rule 4 to find the sum of the first two 1's.

$$\begin{array}{r} 1 \\ +\,1 \\ \hline 10 \end{array}$$

2. Next, take the previous result of 10_2 and add the final 1 to it. Notice that Rule 2 $(0 + 1 = 1)$ is used to find the answer to the first column, and Rule 3 $(1 + 0 = 1)$ is used to find the answer to the second column.

$$\begin{array}{r} 10 \\ +\,1 \\ \hline 11 \end{array}$$

3. Hence another rule has been derived for binary arithmetic. The sum of three 1's in binary is 11_2.

$$\begin{array}{r} 1 \\ +\,1 \\ \hline 11 \end{array}$$

Subtraction

- For subtraction in the decimal system, normally the borrow method is used.
- The first three rules are similar to the decimal system rules.
- The fourth rule, however, needs a little more explanation .since it defines how borrowing is done from another column. Let us look at a simple example to see where this rule comes from. Consider the problem of subtracting 12 from 102.

Table Rules for binary subtraction

Rule 1	Rule 2	Rule 3	Rule 4
0	1	1	0
– 0	– 1	– 0	– 1
0	0	1	1

Binary

- Now, a way is needed to represent numbers such as –3210. When computations are done in decimal, a minus sign precedes a number to make it negative. Since computers can only work with 1’s and 0’s, it is necessary to modify this approach slightly.
  - Only positive numbers have been considered.
  - One solution is to add an extra binary digit to the left of the binary number to indicate whether the number is positive or negative.
  - In computer terminology, this digit is called a sign bit. Remember that a ‘bit’ is simply another name for a binary digit.

Binary

  - When a number is positive, the sign bit is zero, and when the number is negative, the sign bit is one.
  - This approach is called the **signed magnitude representation.**

- **Note that this** is very similar to adding a minus sign in decimal.

- In the signed magnitude representation of binary numbers, the integer is represented by a sign bit in the most significant bit position followed by the binary representation of the magnitude.

  - For a negative integer, the msb is 1 whereas for a positive integer the msb is 0.

- Subtraction of integers in signed 2's complement representation produces results in true signed magnitude form.

1's Complement

- A signed number with 1's complement is represented by changing all the bits that are 1 to 0 and all the bits that are 0 to 1.
- Reversing the digits in this way is also called complementing a number, as illustrated by the examples below.
- −0 and +0 are represented differently even though they are the same algebraically.
- This causes problems when carrying out tests on arithmetic results.
- Hence, 1's complement is an unpopular choice for integer representation.
- Most computers now use a variation of 1's complement (called 2's complement) that eliminates the problem.

2's Complement

- The 2's complement of a binary number is obtained by adding 1 to the 1's complement representation as illustrated by the examples below.

Obtain 2's complement of (i) 10001 and (ii) 101001.

Solution:

Number	1's Complement	2's Complement
10001	01110	01111
101001	010110	010111

Subtraction using signed 1's complement representation

- In this type of representation, subtraction is carried out by addition of 1's complement of the negative number. The sign bit is treated as a number bit in the subtraction method. For the sign bit being 0, i.e., positive, the magnitude part of the result is the true value.

  - For the sign bit being 1, i.e., negative, the magnitude part of the result is in 1's complement form.

- For subtracting a smaller number from a larger number, the 1's complement method is implemented as follows:

  1. Determine the 1's complement of the smaller number.
  2. Add the 1's complement to the larger number.
  3. Remove the final carry (overflow bit) and add it to the result, i.e., if the sum exceeds *n bits, add the extra bit to* the result. This bit is called the end-around carry.

Subtraction using signed 1's complement representation

- For subtracting a larger number from a smaller number, the 1's complement method is as follows:

1. Determine the 1's complement of the larger number.
2. Add the 1's complement to the smaller number.
3. There is no carry (overflow). The result has the proper sign bit but the answer is in 1's complement of the true magnitude.
4. Take the 1's complement of the result to get the final result. The sign of the result is obtained from the sign bit of the result.

Subtraction using signed 2's complement representation

- For subtracting a smaller number from a larger number, the 2's complement method is implemented as follows.

1. Determine the 2's complement of the smaller number.
2. Add the 2's complement to the larger number.
3. Discard the final carry (there is always one in this case).

Multiplication

- Binary multiplication uses the same techniques as decimal multiplication. In fact, binary multiplication is much easier because each multiplying digit is either zero or one.
- When performing binary multiplication, remember the following rules.

1. Copy the multiplicand when the multiplier digit is 1. Otherwise, write a row of zeros.
2. Shift the results one column to the left for a new multiplier digit.
3. Add the results using binary addition to find the product.

Division

- Division of binary numbers uses the same technique as division in the decimal system. It will be helpful to review some of the basic terms of division.

Division

- When doing binary division, some important rules need to be remembered.
- (a) When the remainder is greater than or equal to the divisor, write a 1 in the quotient and subtract.
- (b) When the remainder is less than the divisor, write a 0 in the quotient and add another digit from the dividend.
- (c) If all the digits of the dividend have been considered and there is still a remainder, mark a radix point in the dividend and append a zero. Remember that some fractions do not have an exact representation in binary, so not all division problems will terminate.

Binary Codes

- For practical reasons, it is very convenient to use the binary number system in digital systems or computers.
  - Data is represented by symbols in the form of decimal numbers, alphabets, and special characters. To facilitate extensive communication between humans and digital machines, binary digits 1 and 0 are arranged according to certain defined rules and designated to represent symbols.
  - The method of forming the binary representation is known as **encoding and the** complete group of binary representations corresponding to the symbols is known as **binary code.**

- Binary codes may be broadly classified into four categories.

(a) Numeric codes
(b) Alphanumeric codes
(c) Error-detecting codes
(d) Error-correcting codes

- Numeric codes are further classified as weighted, nonweighted, self-complementing, sequential, and cyclic codes as depicted in Fig.

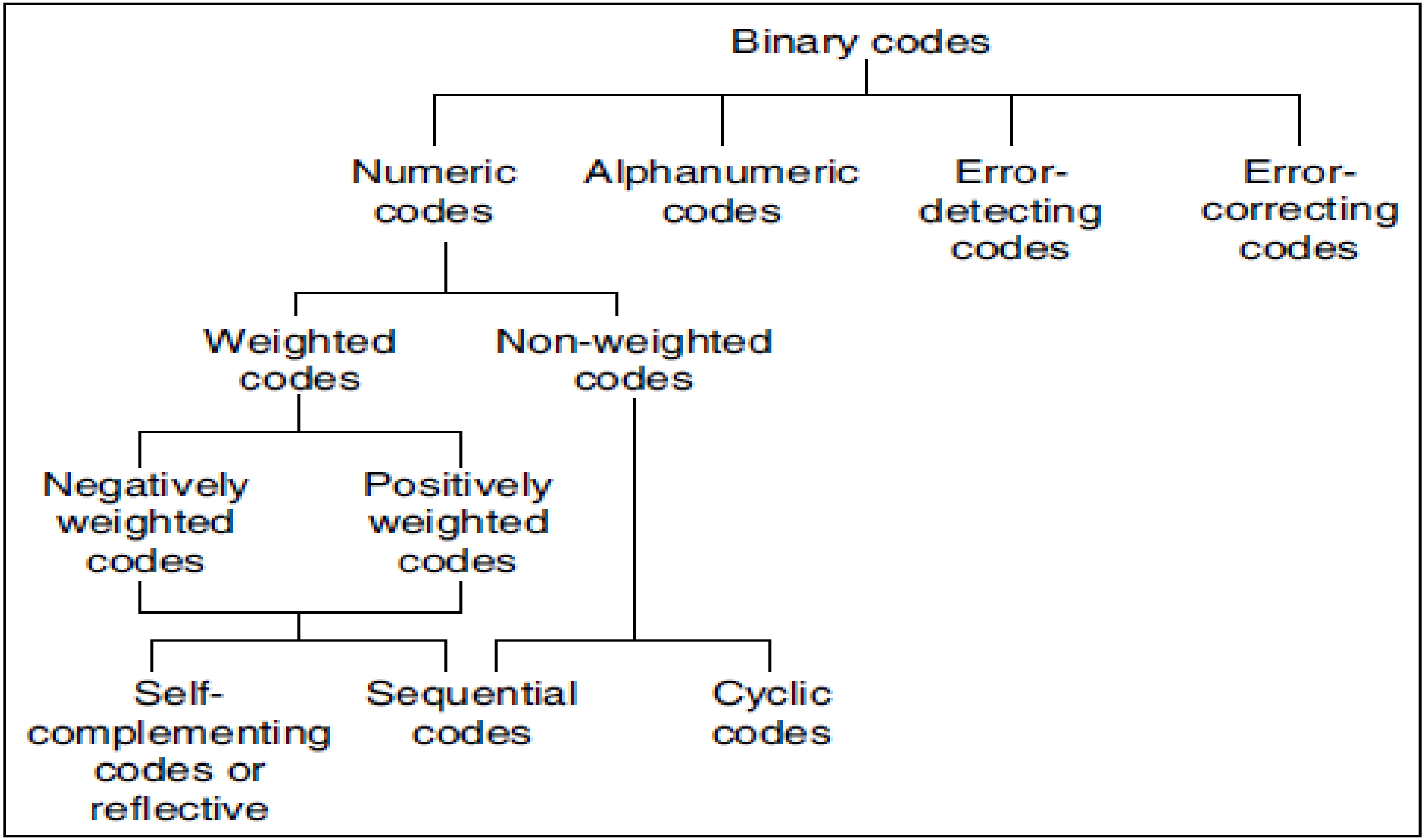


Fig. 2.1 Code classification

Weighted Binary Code

Table 2.13 Some weighted binary coded decimals

Decimal digits	Positive weights (w4, w3, w2, w1)				Negative weights (w3, w2, w1)	
	8421	5421	5211	2421	8 4 −2 −1	6 4 −2 −3
0	0000	0000	0000	0000	0000	0000
1	0001	0001	0001	0001	0111	0101
2	0010	0010	0011	0010	0110	0010
3	0011	0011	0101	0011	0101	1001
4	0100	0100	0111	0100	0100	0100
5	0101	1000	1000	1011	1011	1011
6	0110	1001	1001	1100	1010	0110
7	0111	1010	1011	1101	1001	1101
8	1000	1011	1101	1110	1000	1010
9	1001	1100	1111	1111	1111	1111