Lecture 1 – Overview - Computer Systems

What is a computer?

A **computer** is an electronic device that manipulates information, or data. It has the ability to **store**, **retrieve**, and **process** data. You may already know that you can use a computer to **type documents**, **send email**, **play games**, and **browse the Web**. You can also use it to edit or create **spreadsheets**, **presentations**, and even **videos**.

Computing systems are dynamic!

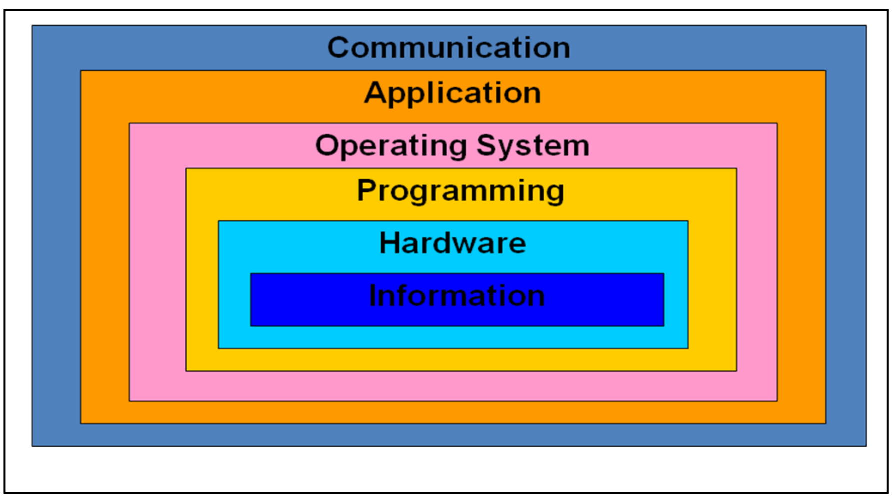
- Constant Modification, Revision, Correction of Hardware and Software
- Expansion of Hardware and Software Capabilities
- Maturation and Extension of Concepts
- Simplification of Procedures
- Improvements in Human Computer Interface

Hardware vs. software :

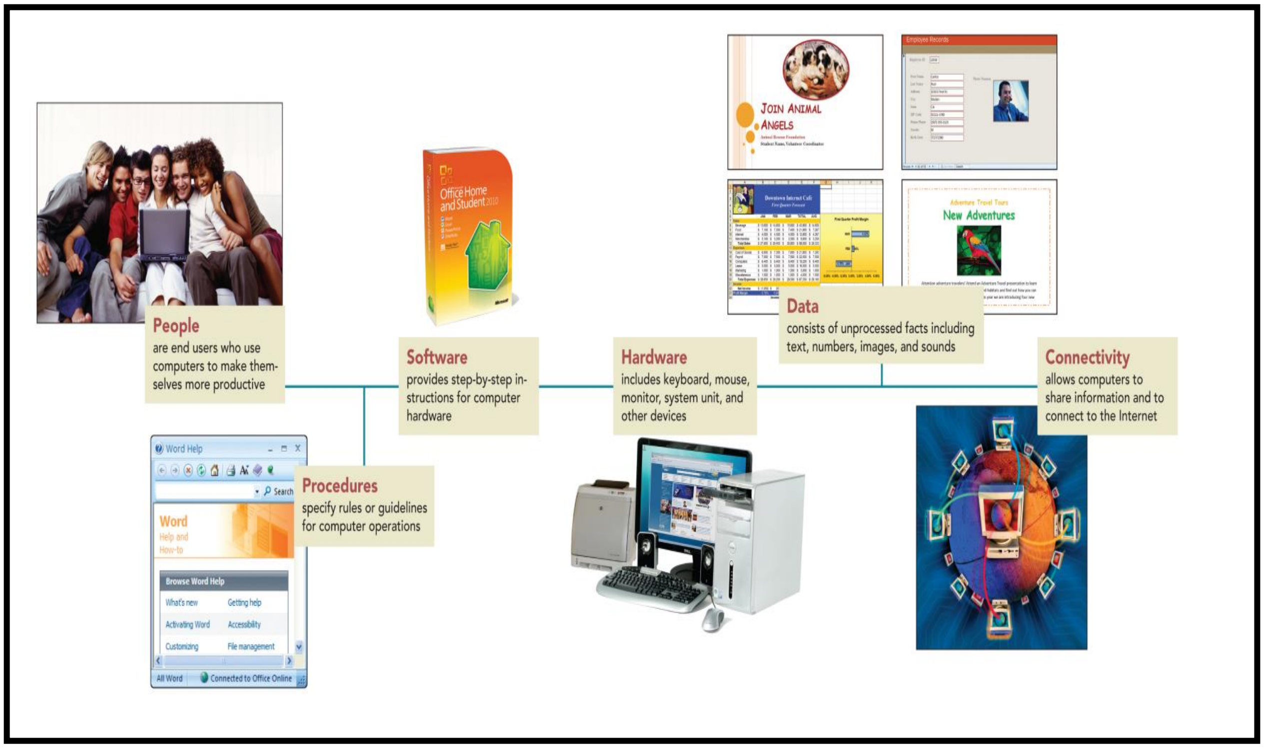
Hardware : The physical elements of a computing system (Mother board, disk drive, memory, controller cards, power supply, communication ports, bus, monitor, keyboard, etc.)

Software : The programs that provide the instructions for a computer to execute (boot program, assembler, compiler, operating system, applications, function library, web browsers, games, word processors , spreadsheets … etc.)

Layers of a Computing System



Parts of any Information System



Types of Computers

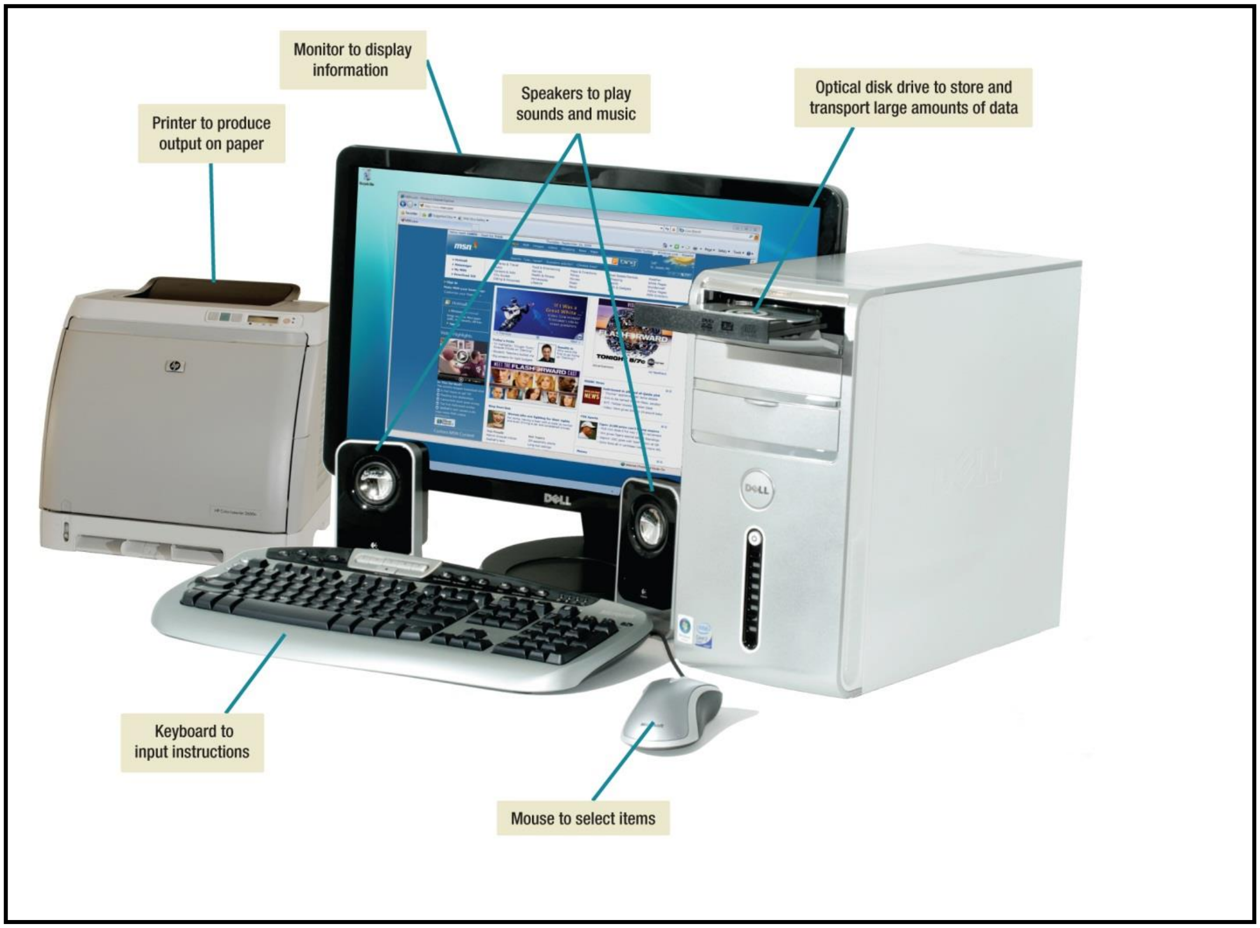
- Supercomputers
- Mainframe computers
- Minicomputers (or mid-range computers)
- Microcomputers

Microcomputer Types

- Desktop
- Media center system units
- Notebook or laptop
- Netbooks
- Tablet PC
- Handheld

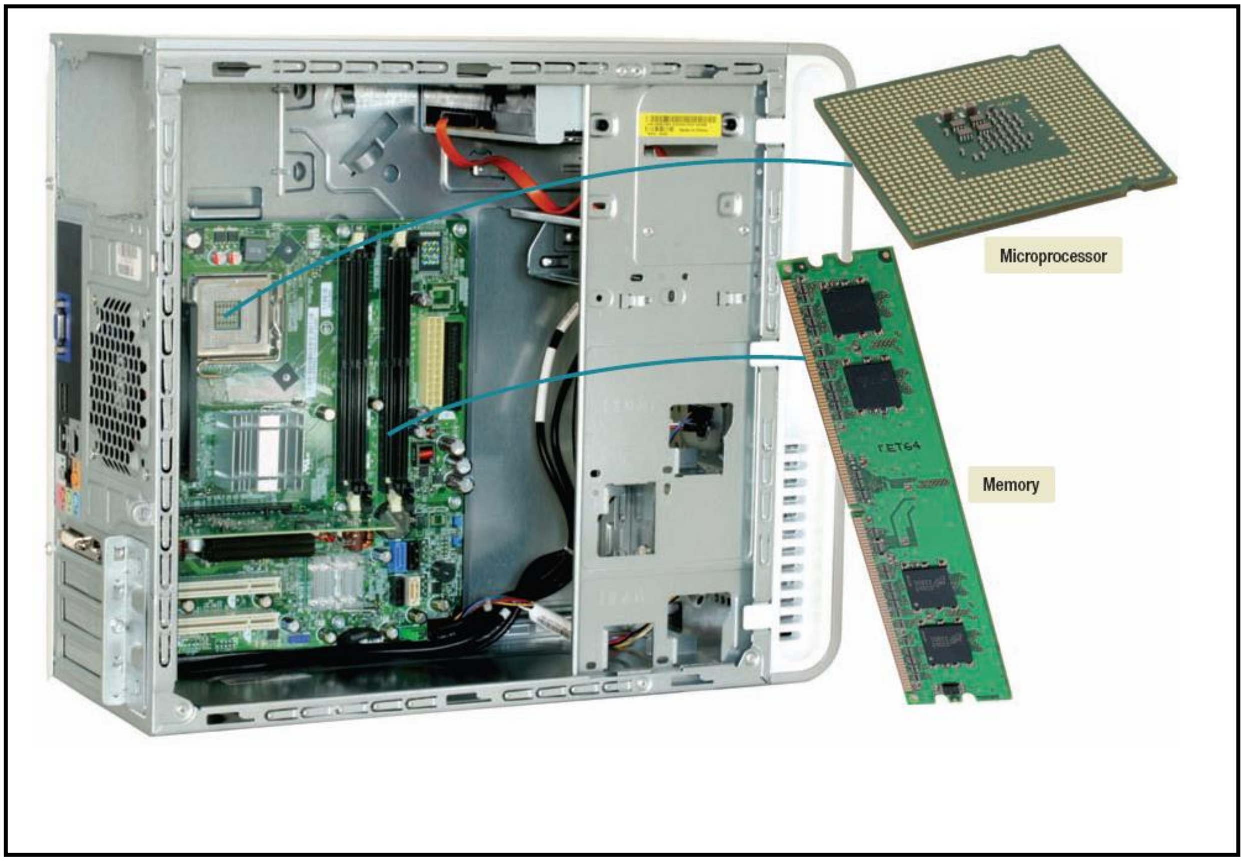
Microcomputer Hardware

- Four basic categories of equipment:
  - System Unit
  - Input/Output
  - Secondary Storage
  - Communications



System Unit

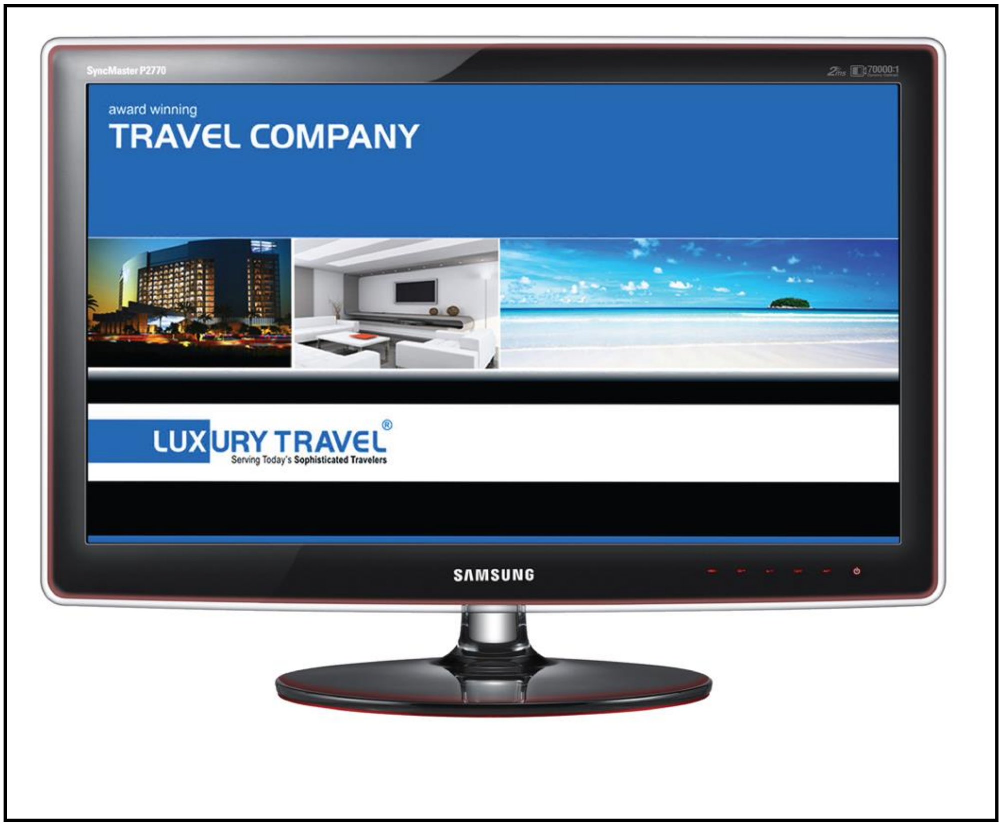
- Two important components
  - Microprocessor
  - Memory



Input/Output Devices

- Common input devices are the keyboard and the mouse

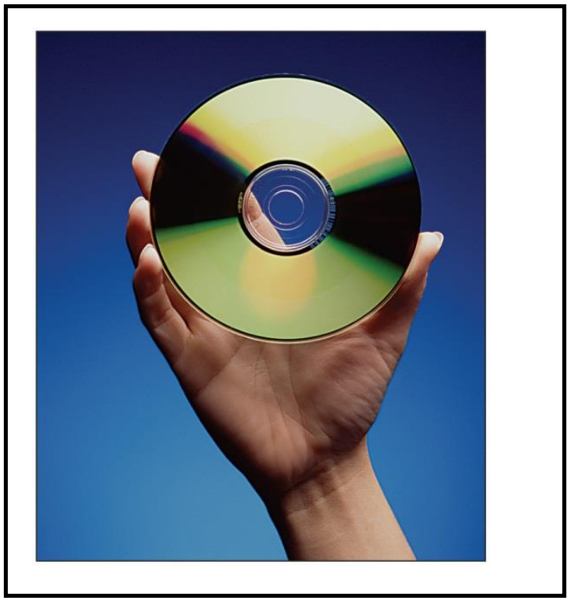
- Common output devices are printers and monitors



Secondary Storage

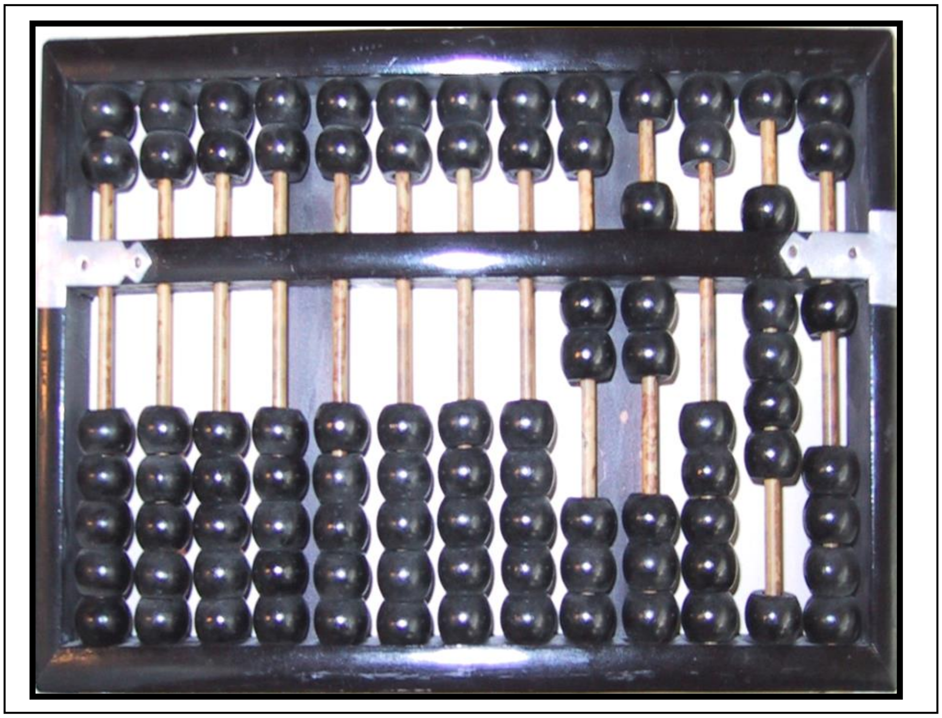
- Unlike memory, secondary storage holds data and programs even if electrical power is not available

- The most important types of secondary media are hard disks, solid-state storage, and optical disks, flash drives



Early History of Computing

Abacus: A device to record, **add and subtract** numeric values using a modified base 5 notation.



Blaise Pascal

Mechanical device to **add, subtract, divide & multiply**

Joseph Jacquard

Jacquard’s Loom, the **punched card**

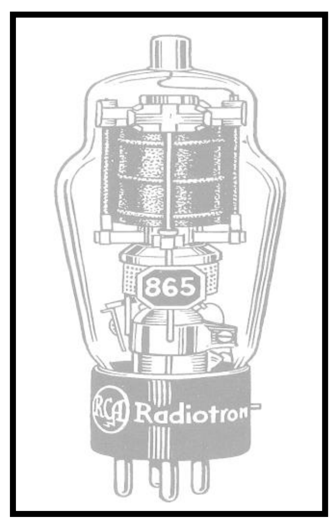
Ada Lovelace - First Programmer, the loop

Ada Lovelace described and published an algorithm for Charles Babbage's analytical engine to **compute** Bernoulli numbers. It is generally considered the first algorithm ever specifically tailored for implementation on a computer, and for this reason she is considered by many to be the first computer programmer.

Alan Turing - Turing Machine, Artificial Intelligence Testing

Alan Mathison Turing, 23 June 1912 – 7 June 1954), was an English mathematician, logician, cryptanalyst, and computer scientist. He was highly influential in the development of computer science, **giving a formalization of the concepts of "algorithm" and "computation"** with the Turing machine, which can be considered a model of a general purpose computer. Turing is widely considered to be the **father of computer science and artificial intelligence**.

First Generation Hardware (1951-1959)

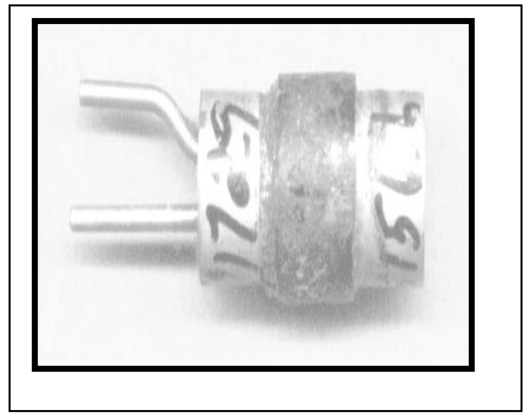


Vacuum Tubes
Large, not very reliable, generated a lot of heat

Magnetic Drum
Memory device that rotated under a read/write head

Card Readers → Magnetic Tape Drives
Sequential auxiliary storage devices

Second Generation Hardware (1959-1965)



Transistor

Replaced vacuum tube, fast, small, durable, cheap

Magnetic Cores
Replaced magnetic drums, information available instantly

Magnetic Disks
Replaced magnetic tape, data can be accessed directly

Third Generation Hardware (1965-1971)

Integrated Circuits - Replaced circuit boards, smaller, cheaper, faster, more reliable.
Transistors - Now used for memory construction
Terminal - An input/output device with a keyboard and screen

Fourth Generation Hardware (1971-?)

Large-scale Integration - Great advances in chip technology
PCs, the Commercial Market, Workstations - Personal Computers were developed as new companies like Apple and Atari came into being. Workstations emerged.

Parallel Computing and Networking

Parallel Computing
Computers rely on interconnected central processing units that increase processing speed.

Networking
With the Ethernet small computers could be connected and share resources. A file server connected PCs in the late 1980s.

ARPANET and LANs → Internet

First Generation Software (1951-1959)

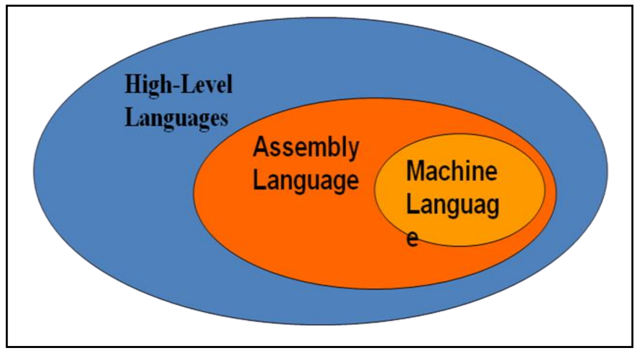
Machine Language
Computer programs were written in binary (1s and 0s)

Assembly Languages and translators
Programs were written in artificial programming languages and were then translated into machine language

Programmer Changes
Programmers divide into application programmers and systems programmers

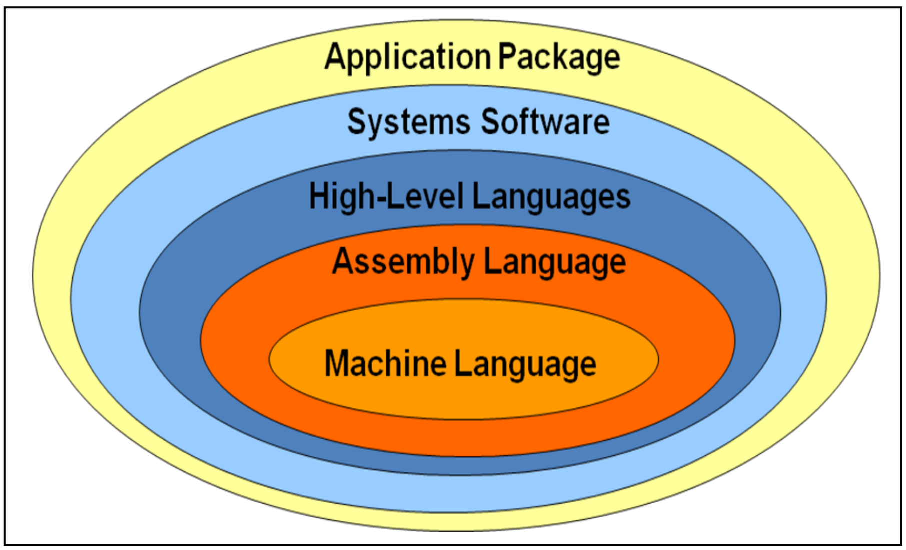
Second Generation Software (1959-1965)

High Level Languages
Use English-like statements and make programming easier. Fortran, COBOL, Lisp are examples.



Third Generation Software (1965-1971)

- **Systems Software**
  - utility programs,
  - language translators,
  - and the operating system, which decides which programs to run and when.
- **Separation between Users and Hardware**
  Computer programmers began to write programs to be used by people who did not know how to program



Fourth Generation Software (1971-1989)

Structured Programming **-** Pascal, C, C++, Rust.
New Application Software for Users **-** Spreadsheets, word processors, database management systems

Fifth Generation Software (1990- present)

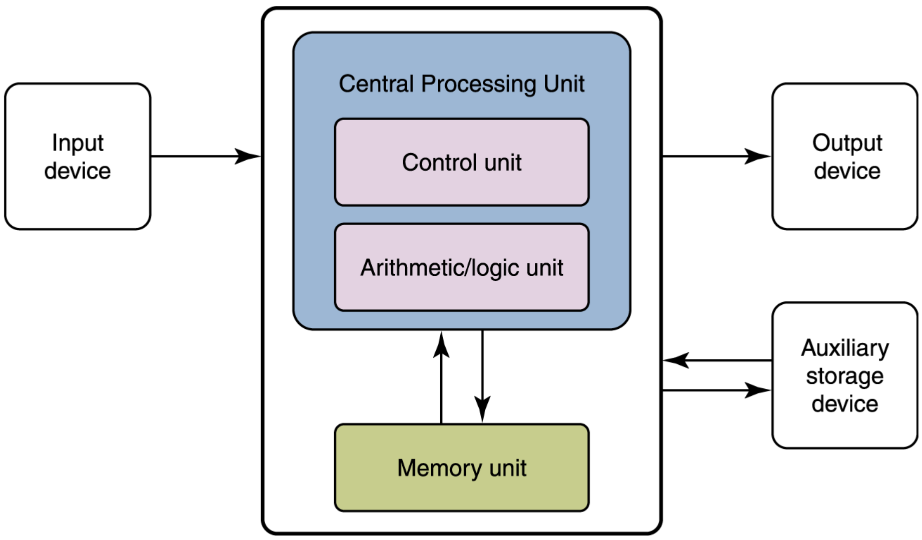
Microsoft **-** The Windows operating system, and other Microsoft application programs dominate the market

Object-Oriented Design **-** Based on a hierarchy of data objects (i.e. Java)

World Wide Web **-** Allows easy global communication through the Internet

New Users **-**Today's user needs no computer knowledge

Stored-Program Concept

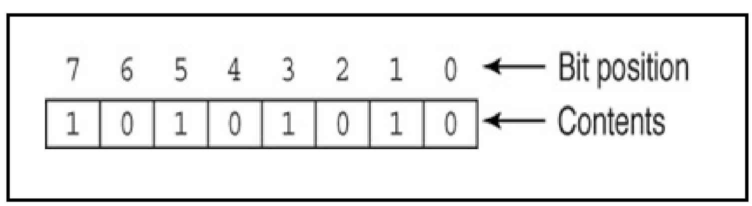


The von Neumann architecture

Memory

Memory is a collection of cells, each with a unique physical address ; Both addresses and contents are in binary

Address	Contents
00000000	11100011
00000001	10101001
:	:
11111100	00000000
11111101	11111111
11111110	10101010
11111111	00110011



Input / Output Devices

Have you ever wondered how information gets into your computer or comes out in a form you can use?

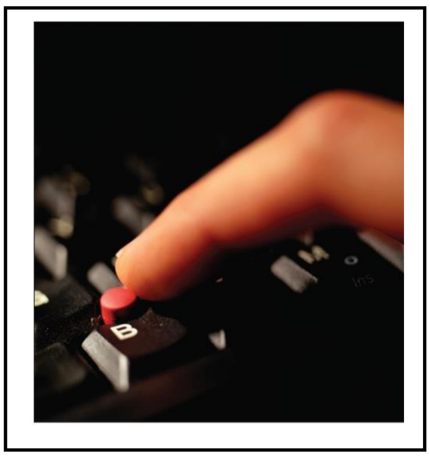
- Input devices convert what we understand into what the system unit can process

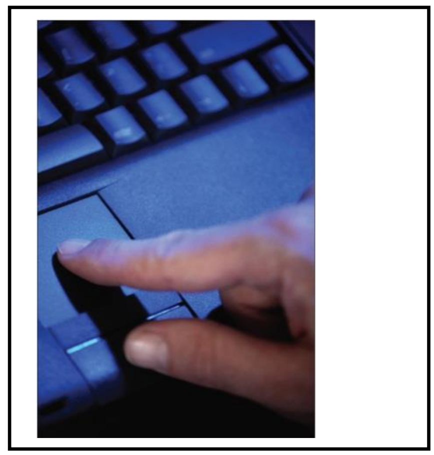
- Output devices convert what the system unit has processed into a form that we can understand.

Input Unit

A device through which data and programs from the outside world are entered into the computer

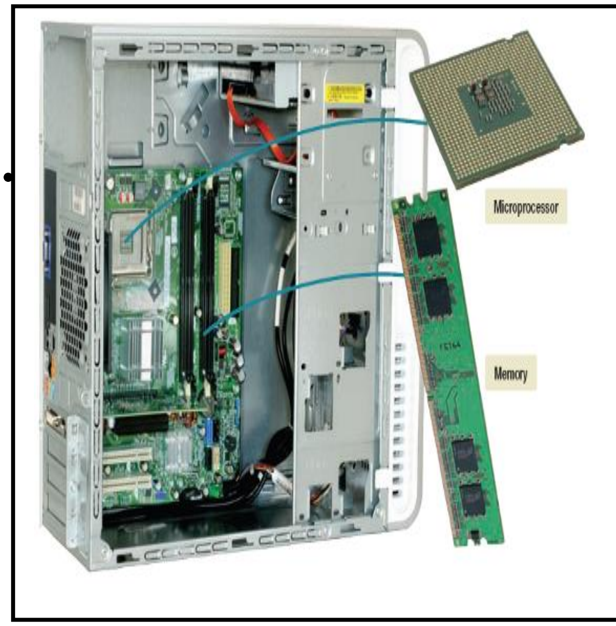
- Keyboard (traditional – wireless – PAD Keyboards)
- Mouse (Mechanical - Optical – Wireless) (Trackball – Touch Pads – Pointing or J joysticks – Stylus)
- Scanning devices (Magnetic Card Readers Bar Code Readers – Character Recognition Devices -Webcams – Voice Recognition Systems – Microphone)

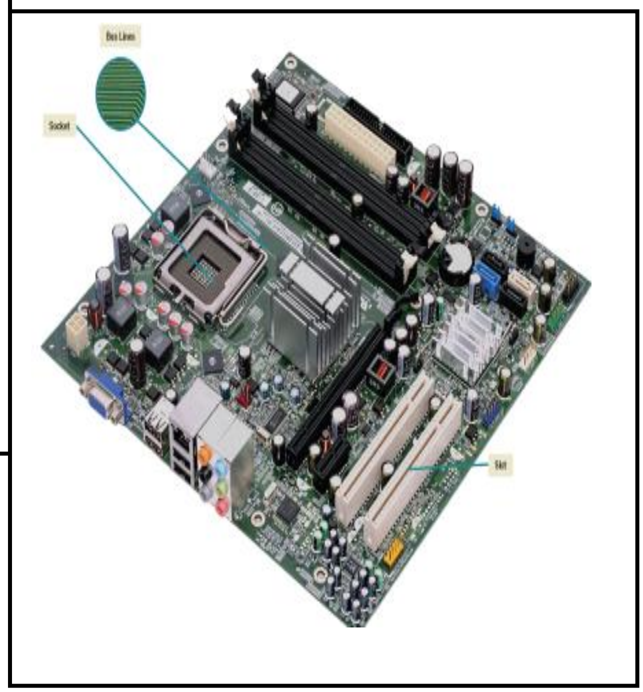




System Unit

System Chassis, System Board (Motherboard), Microprocessor, Memory, Socket, Bus Lines, and Expansion Slots



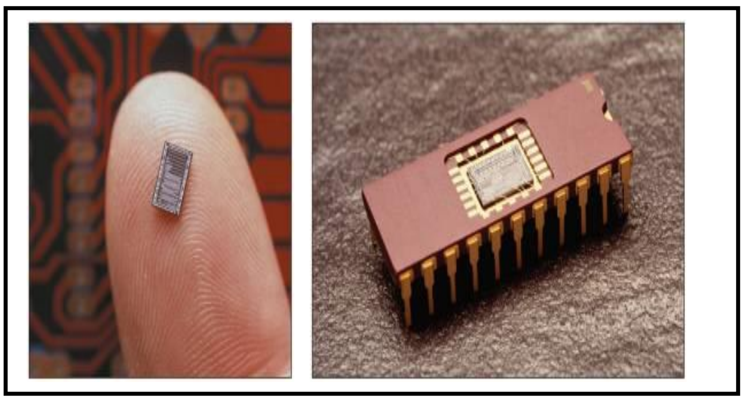


System Board (Main board or motherboard)

Central Processing Unit (CPU)

- **Measurement units for processing speed (shown here)**

- **Two Basic Components**
  - **Control unit**
  - **Arithmetic-logic unit (ALU)**

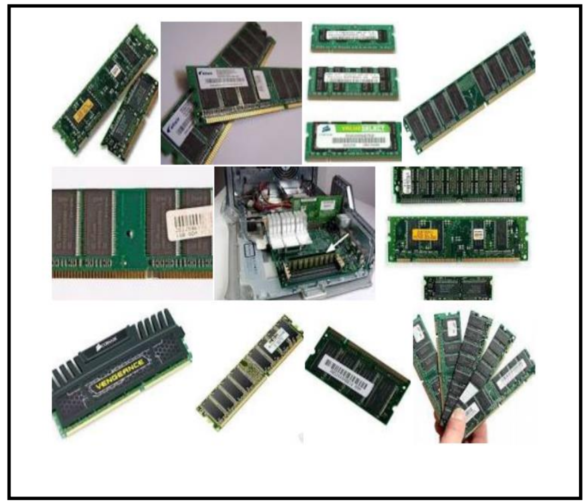


Unit	Speed
Microsecond	Millionth of a second
Nanosecond	Billionth of a second
Picosecond	Trillionth of a second

Memory :

- Holding area for data, instructions, and information
- Memory is contained on chips connected to the system board

RAM and ROM



RAM stands for **Random Access Memory**

- Inherent in the idea of being able to access each location is the ability to change the contents of each location **flash** memory or **Cache** memory

ROM stands for **Read Only Memory**

- The contents is in memory
- locations in ROM can be accessed but cannot be changed

- Contain special instructions
  - **Needed to start a computer**
  - **Give keyboard keys their special capabilities**

RAM is Volatile, ROM is not

- This means that RAM does not retain its bit configuration when the power is turned off, but ROM does

- **Expansion Cards**:
  - Graphics cards
  - Sound cards
  - Wireless network cards
  - TV tuner cards Allows you to view your favorite TV shows while running other applications such as word

Plug and Play

- Set of specific hardware and software standards developed by Intel, Microsoft, and others
- Creating devices that are able to configure themselves when installed

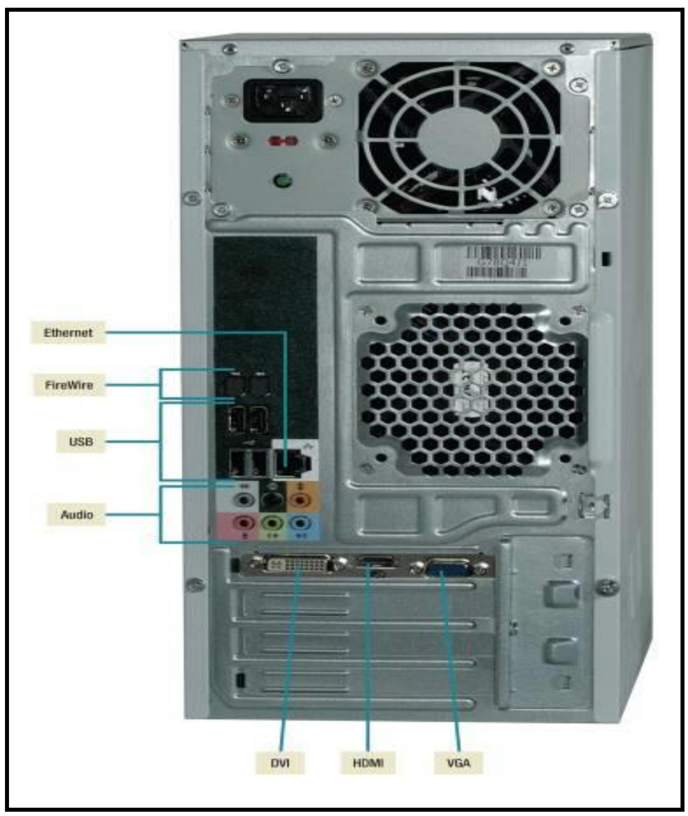
Ports

- Socket for connecting external devices

- **Three Types of ports:**
    - **Standard Ports** such as USB ports
    - **Legacy Ports** such as Keyboard and mouse ports
    - **Specialized Ports** such as High Definition Multimedia Interface (HDMI)

Standard Ports

- Four common ports
    - VGA
    - USB ports
    - FireWire ports
    - Ethernet ports



Legacy Ports

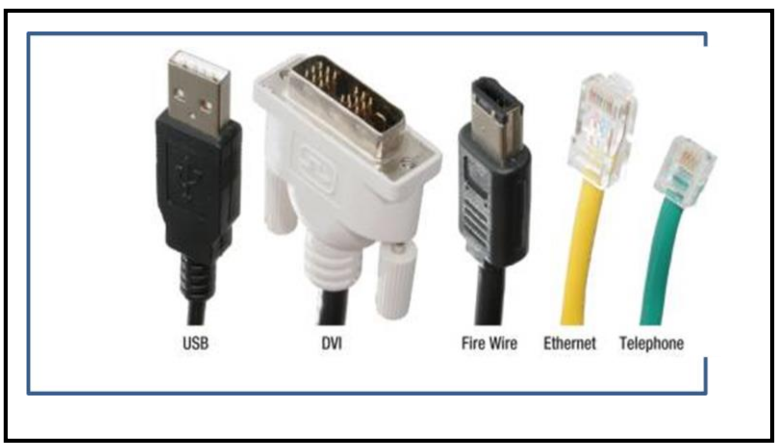
- Keyboard and mouse ports
- Game ports

Specialized Ports

- Musical Instrument digital interface (MIDI)
- High Definition Multimedia Interface (HDMI)

Cables

- Used to connect external devices to the system unit via the ports
- One end of the cable is attached to the device and the other end has a connector that is attached to a matching connector on the port



Power Supply

- Computers require direct current (DC)
- DC power provided by converting alternating current (AC) from wall outlets or batteries
- Desktop computers use power supply units
- Notebooks and handhelds use AC adapters

Secondary Storage Devices

- Because most of main memory is volatile and limited, it is essential that there be other types of storage devices where programs and data can be stored when they are no longer being processed

- Secondary storage devices can be installed within the computer box at the factory or added later as needed

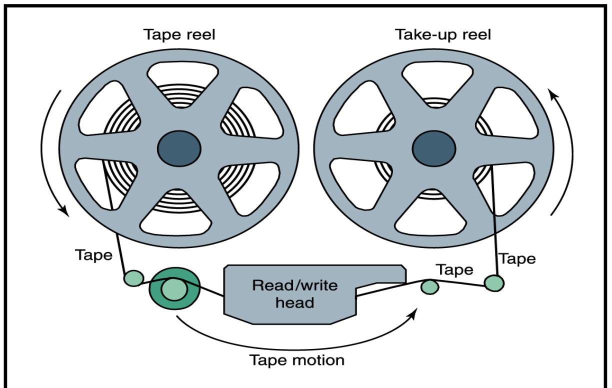
Can you name some of these devices ????

Why is it necessary to have secondary storage devices ???

Magnetic Tape

The first truly mass auxiliary storage device was the magnetic tape drive

Tape drives have a major problem;

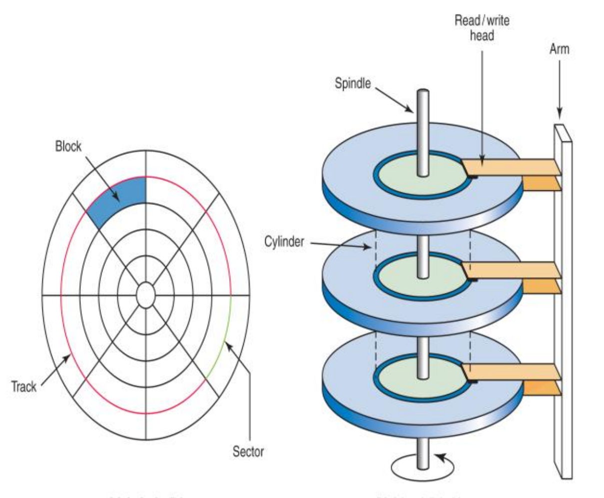


A magnetic tape storage mechanism

can you describe it ???

<u>Magnetic Disks</u>

A read/write head travels across a spinning magnetic disk, retrieving or recording data –
Floppy Desk (1970's) , and
Zip Drive (Late 1970's / early 198-'s).



(a) A single disk ⟷ (b) A hard disk drive

Compact Disks :

A CD drive uses a laser to read information stored optically on a plastic disk

- CD-ROM is A compact disk Read-Only Memory .
- CD-DA digital audio.
- CD-WORM write once, read many
- RW or RAM both read from and written to
- DVD stands for Digital Video Disk has higher storage capacity than CD-ROM. used for storing audio and video

Flash Memory

- Nonvolatile
- Can be erased and rewritten
- Flash memory offers a combination of the features of RAM and ROM.
- Flash memory is used for a wide of range of applications.
- If changes are made to the computer system, these changes are reflected in flash memory

Output Unit

A device through which results stored in the computer memory are made available to the outside world

- Printers and video display terminals

Output Devices

- Processed data or information

- Types of output
  - Text
  - Graphics/Photos
  - Audio & video

- Examples of Output devices
  - Monitors
  - Printers
  - Other Devices

Monitors

- Known as screens or display screens
- Output referred to as soft copy

Touch Screens

- **Touch screen** A computer monitor that can respond to the user touching the screen with a stylus or finger

Infrared touch

Printers

- Ink-jet printer
- Laser printer
  - Personal laser printers
  - Shared laser printers
- Thermal printer
- Other printers
  - Dot-matrix printers
  - Plotters
  - Photo printers
  - Portable printers

Audio-Output Devices

- Translates audio information from the computer into sounds that people can understand
- Common devices
  - Speakers
  - Headphones
- Digital Music Player
  - iPod

Binary Values and Number Systems, Data Representation.

Electronic Data and Instructions

- **Data and instructions are represented electronically**
- **Two-state system or Binary System**
  - **Off/On electrical states**
  - **Characters represented by 0's (off) and 1's (on)**
  - **Bits**
  - **Bytes**

Numbers

Natural Numbers

Zero and any number obtained by repeatedly adding one to it.
Examples: 100, 0, 45645, 32

Negative Numbers

A value less than 0, with a – sign
Examples: -24, -1, -45645, -32

Integers

A natural number, a negative number, zero
Examples: 249, 0, - 45645, - 32

Rational Numbers

Rational Number is an integer number that can be written as a simple fraction (i.e. as a **ratio** of two integers).
Examples: 3/7, -2/5

Irrational Numbers

Numbers that cannot be represented by the quotient of two integers
Examples: PI = 3.14159 . . ., e, √2, etc.

Natural Numbers

How many ones are there in 642?

Is it 600 + 40 + 2 ?
Or is it 384 + 32 + 2 ?
Or maybe… 1536 + 64 + 2 ?

We need a base for each number .

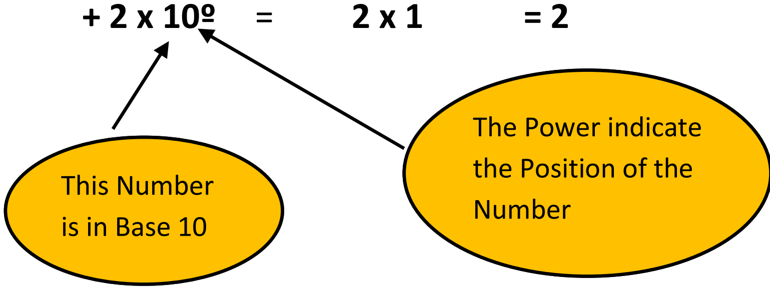
The base of a number determines the number of digits and the value of digit positions

Thus 642 is 600 + 40 + 2 in **BASE 10**

Positional Notation

Continuing with our example…
642 in base 10 positional notation is:

6×10^2 = 6 x 100 = 600
$+ 4 \times 10^1$ = 4 x 10 = 40
$+ 2 \times 10^0$ = **2 x 1 = 2** = **642 in base 10**



642 is $6_3 * 10^2 + 4_2 * 10^{+}\ 2_1$

What if 642 has the base of 13?

Then $6 \times 13^2 = 6 \times 169 = 1014$

$+ 4 \times 13^1 = 4 \times 13 = 52$

$+ 2 \times 13^0 = 2 \times 1 = 2$

= 1068 in base 10

642 in base 13 is equivalent to 1068 in base 10

Decimal Numeral System - Base-10

Decimal numbers uses digits from 0..9. These are the regular numbers that we use. Each digit has an associated value of an integer raised to the power of 10

Example: $2538_{10} = 2\times10^3 + 5\times10^2 + 3\times10^1 + 8\times10^0$

Example: $724.5_{10} = 7\times10^2 + 2\times10^1 + 4\times10^0 + 5\times10^{-1}$

Bases Higher than 10

How are digits in bases higher than 10 represented?

With distinct symbols for 10 and above.

Base 16 has 16 digits: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F

Hexadecimal Numeral System - Base-16

Hex numbers uses digits from 0..9 and A..F.

H denotes hex prefix.

Converting Hexadecimal to Decimal

What is the decimal equivalent of the hexadecimal number DEF?

$D \times 16^2$	=	13 x	256	= 3328
$+ E \times 16^1$	=	14 x	16	= 224
$+ F \times 16^0$	=	15 x	1	= 15
				= 3567 in base 10

Example : What is the decimal equivalent of the hexadecimal number

$(B65F)_{16}$?

$(B65F)_{16} = 11\times16^3 + 6\times16^2 + 5\times16^1 + 15\times16^0 = (46687)_{10}$

Remember, the digits in base 16 are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

So far , we have the following :

Decimal is base 10 and has 10 digits: 0,1,2,3,4,5,6,7,8,9
Hexadecimal is Base 16 has 16 digits: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F

Binary

Binary is base 2 and has 2 digits: 0,1

Each digit has an associated value of 0 or 1 raised to the power of 2 .

B denotes binary prefix

Converting Binary to Decimal

What is the decimal equivalent of the binary number 1101110?

1×2^6	=	1 x 64	= 64
$+ 1 \times 2^5$	=	1 x 32	= 32
$+ 0 \times 2^4$	=	0 x 16	= 0
$+ 1 \times 2^3$	=	1 x 8	= 8
$+ 1 \times 2^2$	=	1 x 4	= 4
$+ 1 \times 2^1$	=	1 x 2	= 2
$+ 0 \times 2^0$	=	0 x 1	= 0
			= **110 in base 10**

What is the decimal equivalent of the binary number $(11010)_2$

Sol : $(11010)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 = (26)_{10}$

Converting Decimal to Binary

The first way to convert a decimal number to a binary one is with a table like the one below (if needed you can add more columns - each new column to the left should be twice the value of the preceding one). Then, working from left to right, decide if you need that column's value to make your number. The value for any column you use should be subtracted from the value you are trying to make.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**

For example, I want to write 155 as a binary number:

```
Do I need 128? Yes, that is less than 155.
    155 - 128 = 27: that is what we have left to make
Do I need 64? No - I only have 27 left
Do I need 32? No - I only have 27 left
Do I need 16? Yes, that is less than 27.
    27 - 16 = 11: that is now what we have left
Do I need 8? Yes, that is less than 11.
    11 - 8 = 3
Do I need 4? No - I only have 3 left
Do I need 2? Yes.
    3 - 2 = 1
Do I need 1? Yes.
    1 - 1 = 0
```

If we think of each column we used as representing a 1 and each column we didn't as a 0, we get this:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	1	1	0	1	1

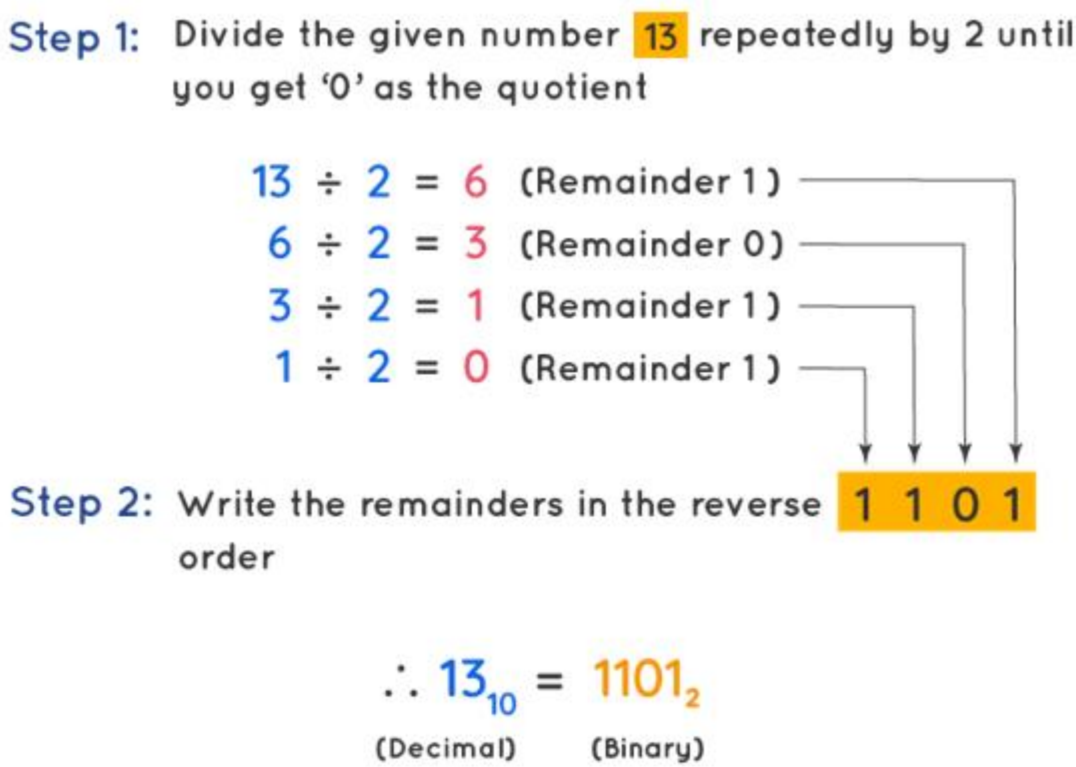
Or written without a table: $(10011011)_2$

Converting integer :

For example, 46 in base 2 is 101110_2

2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1
1	0	1	1	1	0

Another way to Convert from Decimal to Binary:



Or

2^3	2^2	2^1	2^0
8	**4**	**2**	**1**
1	**1**	**0**	**1**

Relationship between Bits and Byte → 8 bits = I Byte.

Name	Abbreviation	# of Bytes	Size
Byte	B	1 = 8 bits	Could hold 1 Character of data
Kilobyte	KB	1,024 Bytes = 1024 * 8 = 8192 bits	Could hold 1024 Characters of data (half of double spaced typewriter paper = 2^{10})
Megabyte	MB	KB * KB = 1,048,576 Bytes	A floppy disk holds 1.4 MB of data – around 768 pages of typed text ~~ 2^{20}
Gigabyte	GB	MB * KB = 1,073,741,824 Bytes	Around 786,432 pages of text - Stack of papers that is 262 feet high ~~ 2^{30}
Terabyte	TB	GB * KB = 1,099,511,627,776 Bytes	Stack of typewritten pages that is almost 51 miles high ~~ 2^{40}
Petabyte	PB	TB * KB = 1,125,899,906,842,624 Bytes	Stack of typewritten pages that is almost 52,000 miles high – about one-fourth distance from the earth to the moon.

The number of bits in a word determines the word length of the computer, but it is usually a multiple of 8

- 32-bit machines
- 64-bit machines etc.

Binary Coded Decimal (BCD)

Binary coded decimal (BCD) is a system of writing numerals that assigns a four-digit binary code to each digit 0 through 9 in a decimal (base-10) numeral. The four-bit BCD code for any single base-10 digit is its representation in binary notation, as follows:

0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111
8 = 1000
9 = 1001

Numbers larger than 9, having two or more digits in the decimal system, are expressed digit by digit. For example, the BCD rendition of the base-10 number 1895 is

0001 1000 1001 0101

The binary equivalents of 1, 8, 9, and 5, always in a four-digit format, go from left to right.

Example : $(791)_{10} = (0111 \quad 1001 \quad 0001)_{BCD}$

Note : BCD was used in some of the early decimal computers, as well as the IBM System/360 series systems.

Warning: Conversion or Coding?

Do NOT mix up:

- Conversion of a decimal number to a binary number
- **With** coding a decimal number with a binary code.

13_{10} = 1101_2 **(This is conversion)**

13 ⇔ 0001|0011 **(This is coding)**

ASCII Character Code

<u>**A**</u>merican <u>**S**</u>tandard <u>**C**</u>ode for <u>**I**</u>nformation <u>**I**</u>nterchange
This code is a popular code used to represent information sent as character-based data (like this presentation).

It uses 7-bits to represent:

- 94 Graphic printing characters.
- 34 Non-printing characters

Some non-printing characters are used for text format (e.g. BS = Backspace, CR = carriage return)

Other non-printing characters are used for record marking and flow control (e.g. STX and ETX start and end text areas).

ASCII Table :

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Del

asciicharstable.com

Gates and Circuits

Computers and Electricity

Gate A device that performs a basic operation on electrical signals

Circuits Gates combined to perform more complicated tasks

There are three different, but equally powerful, notational methods for describing the behavior of gates and circuits

1. Boolean expressions
2. Logic diagrams
3. Truth tables

Boolean Expressions: Expressions in Boolean algebra, a mathematical notation for expressing two-valued logic

This algebraic notation are an elegant and powerful way to demonstrate the activity of electrical circuits

Logic Diagram: A graphical representation of a circuit - Each type of gate is represented by a specific **graphical** symbol

Truth Table: A **table** showing all possible input value and the associated output values

Gates

Let's examine the processing of the following six types of gates

1. NOT
2. AND
3. OR
4. XOR
5. NAND
6. NOR

Typically, logic diagrams are black and white, and the gates are distinguished only by their shape

NOT Gate

A NOT gate accepts **one input** value and produces **one output** value

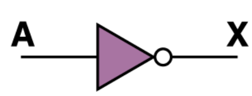
By definition, if the input value for a NOT gate is 0, the output value is 1, and if the input value is 1, the output is 0

A NOT gate is sometimes referred to as an ***inverter*** because it inverts the input value

Boolean Expression

X = A'

Logic Diagram Symbol



Truth Table

A	X
0	1
1	0

AND Gate

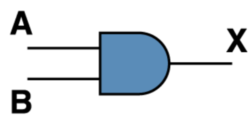
An AND gate accepts **two** input signals.

If the two input values for an AND gate are **both** 1, the output is 1; otherwise, the output is 0

Boolean Expression

$$X = A \cdot B$$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

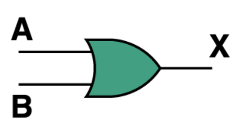
OR Gate

If the two input values are **both** 0, the output value is 0; otherwise, the output is 1

Boolean Expression

$$X = A + B$$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

<u>XOR Gate</u>

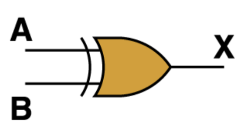
XOR, or *exclusive* OR, gate

- An XOR gate produces 0 if its **<u>two inputs are the same</u>**, and 1 otherwise
- **<u>Note</u>** the difference between the XOR gate and the OR gate; they differ only in one input situation
- When both input signals are 1, the OR gate produces a 1 and the XOR produces a 0

Boolean Expression

$$X = A \oplus B$$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

NAND and NOR Gates

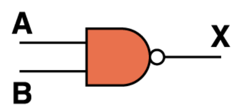
The NAND and NOR gates are essentially the **opposite** of the AND and OR gates, respectively

NAND Gate

Boolean Expression

X = (A · B)'

Logic Diagram Symbol



Truth Table

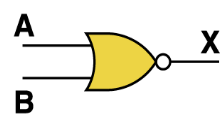
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate

Boolean Expression

X = (A + B)'

Logic Diagram Symbol



Truth Table

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Review of Gate Processing

- A **NOT** gate **inverts** its single input value
- An **AND** gate produces 1 if both input values are 1
- An **OR** gate produces 1 if one or the other or both input values are 1
- An **XOR** gate produces 1 if one or the other (but not both) input values are 1
- A **NAND** gate produces the opposite results of an **AND** gate
- A **NOR** gate produces the **opposite** results of an **OR** gate

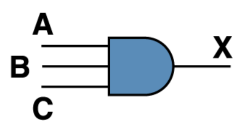
Gates with More Inputs

- Gates can be designed to accept three or more input values
- A three-input AND gate, for example, produces an output of 1 only if all input values are 1

Boolean Expression

$$X = A \cdot B \cdot C$$

Logic Diagram Symbol



Truth Table

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1