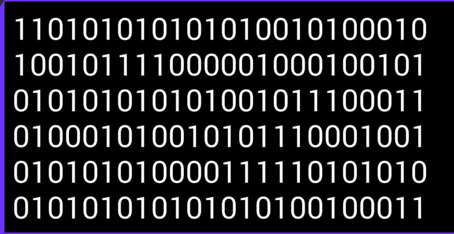
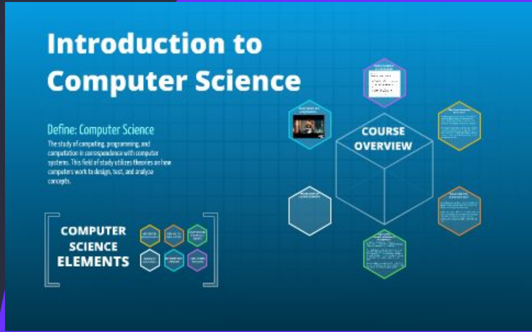
Unit 1 Computer Science Basics

Task 1 Data Manipulation

❖ Binary system

We use languages to communicate. Similarly, computers also use a language to “communicate” with one another and fundamentally each of its own hardware. However, computers only understand one base language; **binary language**. The binary system is the fundamental computer language that computers can only understand in two states; it’s either in **low-voltage state** or **high-voltage state**. All modern devices that are made out chips(circuit) or called digital devices use this system. This system gives out two digits based on voltage state; 0 (low-voltage) and 1 (high-voltage). Every characters, letters and symbols are presented by series 1s and 0s!





How to calculate binary

In our everyday lives, we use the Base-10 numerals which is called Decimal. But computers only understand 1s and 0s which is Binary system, so could the computers understand all the numbers? OK, let's of it this way!

In Base-10 numerals, it consists of 0 to 9. When the ones digit reaches to tens digit, the ones digit reset to 0 and it Increment to the tens digit. So this way, we can easily have as many numbers as we want.

Binary calculations are fundamental because computers store and process all data in binary format. Understanding binary us to interpret how computers handle information. Converting binary to decimal simplifies the representation of large binary numbers for human comprehension. Essentially, binary is the language of computers, making binary calculations essential for anyone working with digital systems. This impacts how we design hardware, write software, and understand data storage and transmission.

Decimal
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15

Hexadecimal

Hexadecimal, also known as hex, is the third commonly used number system. It has 16 units (0-9) and the letters A, B, C, D, E and F. Hex is useful because large numbers can be represented using fewer digits. For example, colour values and MAC addresses are often represented in hex. Additionally, hex is easier to understand than binary. Programmers often use hex to represent binary values as they are simpler to write and check than when using binary.

ASCII

ASCII, which stands for American Standard Code for Information Interchange, is a character encoding standard that assigns a unique numerical value to letters, numbers, and symbols to represent them in computers and other devices. It was the first major computer code for English text and uses 128 characters, with numbers 32 to 126 representing printable characters like letters, numbers, and punctuation. While it has been largely replaced by Unicode for wider language support, ASCII remains compatible with modern systems.

ASCII is a 7-bit characters code, with values from 0 to (7F)16. Unicode characters code is a superset of ASCII that contains the ASCII code with values from 0 to (10FFFF)16.

ASCII	Char	Hex	Bin	ASCII	Char	Hex	Bin
65	A	41	0100 0001	97	a	61	0110 0001
66	B	42	0100 0010	98	b	62	0110 0010
67	C	43	0100 0011	99	c	63	0110 0011
68	D	44	0100 0100	100	d	64	0110 0100
69	E	45	0100 0101	101	e	65	0110 0101
70	F	46	0100 0110	102	f	66	0110 0110
71	G	47	0100 0111	103	g	67	0110 0111
72	H	48	0100 1000	104	h	68	0110 1000
73	I	49	0100 1001	105	i	69	0110 1001
74	J	4A	0100 1010	106	j	6A	0110 1010
75	K	4B	0100 1011	107	k	6B	0110 1011
76	L	4C	0100 1100	108	l	6C	0110 1100
77	M	4D	0100 1101	109	m	6D	0110 1101
78	N	4E	0100 1110	110	n	6E	0110 1110
79	O	4F	0100 1111	111	o	6F	0110 1111
80	P	50	0101 0000	112	p	70	0111 0000
81	Q	51	0101 0001	113	q	71	0111 0001
82	R	52	0101 0010	114	r	72	0111 0010
83	S	53	0101 0011	115	s	73	0111 0011
84	T	54	0101 0100	116	t	74	0111 0100
85	U	55	0101 0101	117	u	75	0111 0101
86	V	56	0101 0110	118	v	76	0111 0110
87	W	57	0101 0111	119	w	77	0111 0111
88	X	58	0101 1000	120	x	78	0111 1000
89	Y	59	0101 1001	121	y	79	0111 1001
90	Z	5A	0101 1010	122	z	7A	0111 1010

<linuxhint/>

Character	Decimal	Hex	Binary	Character	Decimal	Hex	Binary	Character	Decimal	Hex	Binary	Character	Decimal	Hex	Binary
Null	0	0	000_0000	Space	32	20	010_0000	@	64	40	100_0000	`	96	60	110_0000
Start Heading	1	1	000_0001	!	33	21	010_0001	A	65	41	100_0001	a	97	61	110_0001
Start Text	2	2	000_0010	“	34	22	010_0010	B	66	42	100_0010	b	98	62	110_0010
End Text	3	3	000_0011	#	35	23	010_0011	C	67	43	100_0011	c	99	63	110_0011
End Transmission	4	4	000_0100	$	36	24	010_0100	D	68	44	100_0100	d	100	64	110_0100
Enquiry	5	5	000_0101	%	37	25	010_0101	E	69	45	100_0101	e	101	65	110_0101
Acknowledge	6	6	000_0110	&	38	26	010_0110	F	70	46	100_0110	f	102	66	110_0110
Bell	7	7	000_0111	'	39	27	010_0111	G	71	47	100_0111	g	103	67	110_0111
Backspace	8	8	000_1000	(	40	28	010_1000	H	72	48	100_1000	h	104	68	110_1000
Horizontal Tab	9	9	000_1001	)	41	29	010_1001	I	73	49	100_1001	i	105	69	110_1001
Line Feed	10	A	000_1010	*	42	2A	010_1010	J	74	4A	100_1010	j	106	6A	110_1010
Vertical Tab	11	B	000_1011	+	43	2B	010_1011	K	75	4B	100_1011	k	107	6B	110_1011
Form Feed	12	C	000_1100	,	44	2C	010_1100	L	76	4C	100_1100	l	108	6C	110_1100
Carriage Return	13	D	000_1101	-	45	2D	010_1101	M	77	4D	100_1101	m	109	6D	110_1101
Shift Out	14	E	000_1110	.	46	2E	010_1110	N	78	4E	100_1110	n	110	6E	110_1110
Shift In	15	F	000_1111	/	47	2F	010_1111	O	79	4F	100_1111	o	111	6F	110_1111
Data Link Esc	16	10	001_0000	0	48	30	011_0000	P	80	50	101_0000	p	112	70	111_0000
Dev Control 1	17	11	001_0001	1	49	31	011_0001	Q	81	51	101_0001	q	113	71	111_0001
Dev Control 2	18	12	001_0010	2	50	32	011_0010	R	82	52	101_0010	r	114	72	111_0010
Dev Control 3	19	13	001_0011	3	51	33	011_0011	S	83	53	101_0011	s	115	73	111_0011
Dev Control 4	20	14	001_0100	4	52	34	011_0100	T	84	54	101_0100	t	116	74	111_0100
Neg Acknowledge	21	15	001_0101	5	53	35	011_0101	U	85	55	101_0101	u	117	75	111_0101
Sync Idle	22	16	001_0110	6	54	36	011_0110	V	86	56	101_0110	v	118	76	111_0110
End Transmission	23	17	001_0111	7	55	37	011_0111	W	87	57	101_0111	w	119	77	111_0111
Cancel	24	18	001_1000	8	56	38	011_1000	X	88	58	101_1000	x	120	78	111_1000
End of Medium	25	19	001_1001	9	57	39	011_1001	Y	98	59	101_1001	y	121	79	111_1001
Substitute	26	1A	001_1010	:	58	3A	011_1010	Z	90	5A	101_1010	z	122	7A	111_1010
Escape	27	1B	001_1011	;	59	3B	011_1011	[	91	5B	101_1011	{	123	7B	111_1011
File Separator	28	1C	001_1100	<	60	3C	011_1100	\	92	5C	101_1100	|	124	7C	111_1100
Group Separator	29	1D	001_1101	=	61	3D	011_1101	]	93	5D	101_1101	}	125	7D	111_1101
Record Separator	30	1E	001_1110	>	62	3E	011_1110	^	94	5E	101_1110	~	126	7E	111_1110
Unit Separator	31	1F	001_1111	?	63	3F	011_1111	_	95	5F	101_1111	Delete	127	7F	111_1111

ASCII Table

CONTROL.COM

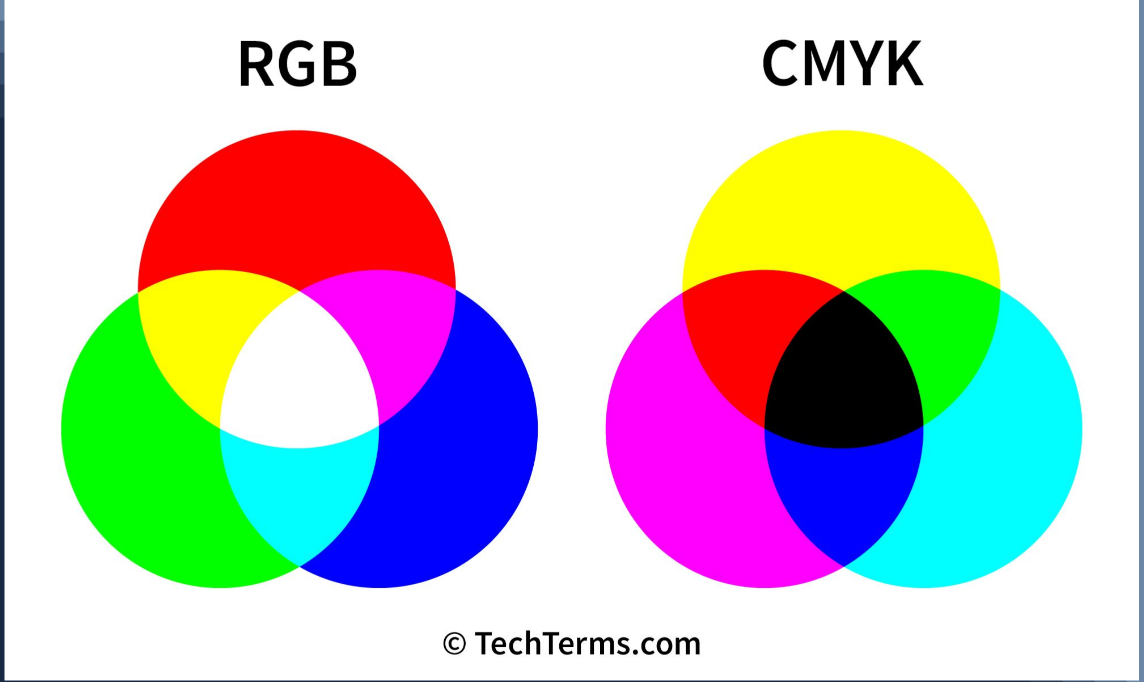
RGB is an acronym for three colours - red, green and blue - and describes a colour model most frequently used in digital displays.

This model works by combining red, green and blue light to create a vast array of colours from the spectrum.

While RGB has been used occasionally in conventional photography, its main purpose is in digital displays - smartphone screens, televisions and computer monitors.

RGB Hex

A Hex code is a six-digit, combining 3 two-digit pairs, hexadecimal representation of an RGB color. It's structure is (#RRGGBB) in which each pair represents a color channel from 0-255. EX: #00FF00 is the color green.



It's a more compact way to represent the byte values (0-255) used in RGB. Each pair of hexadecimal digits can represent a number from 0 (00) to 255 (FF).

How monitor display images?

Monitors display images by controlling how light and color appear on the screen.

1. A monitor screen is made up of thousands (or millions) of pixels. Each pixel is made of three smaller parts called subpixels (Red, Green, and Blue), which can mix to create a full spectrum of colors. By adjusting the intensity of these subpixels, the monitor produces images.
2. The graphics card in your computer sends data (signals) to the monitor, telling it how much red, green, and blue light each pixel should display. This forms the image, text, or video on the screen.
3. Once the signals are received, the monitor adjusts the pixels to show the image. This happens in real-time, constantly refreshing the screen to display what's happening on your computer.

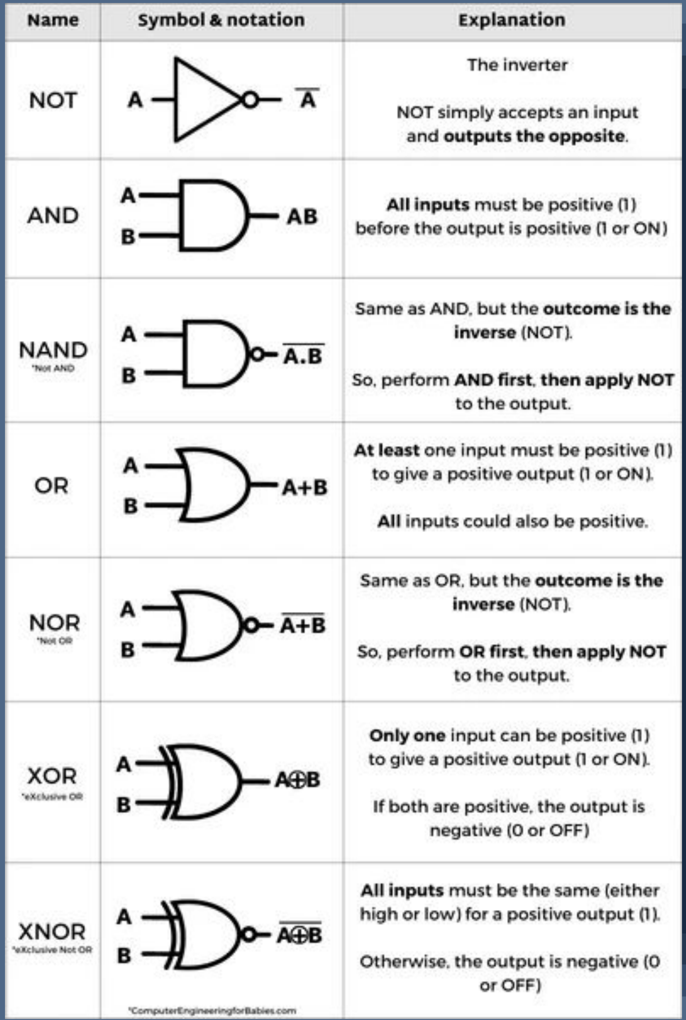
Logic

In math, logic is the study of valid reasoning, sensible conclusion using mathematical deduction. Simply put, we use values to represent the implication of reality around us.

For example, 1 + 2 = 3 and 4 is even are clearly true, while all prime numbers are even is false. In logic we are often not interested in these statements themself, but how true and false statements are related to each other. Therefore we represent the propositions simply by placeholders like P and Q. All these propositions have to be either true (T) or false (F).

The symbol ∧ is a conjunction and is used for "and": P and Q is notated $P \wedge Q$.

The symbol V is a disjunction and is used for "or" (here "or" is not exclusive): P or Q is notated $P \vee Q$.



Name	Symbol & notation	Explanation
NOT	$\overline{A}$	The inverter NOT simply accepts an input and **outputs the opposite**.
AND	AB	**All inputs** must be positive (1) before the output is positive (1 or ON)
NAND "Not AND	$\overline{A.B}$	Same as AND, but the **outcome is the inverse** (NOT). So, perform **AND first, then apply NOT** to the output.
OR	$A+B$	**At least** one input must be positive (1) to give a positive output (1 or ON). **All** inputs could also be positive.
NOR "Not OR	$\overline{A+B}$	Same as OR, but the **outcome is the inverse** (NOT). So, perform **OR first, then apply NOT** to the output.
XOR "eXclusive OR	$A \oplus B$	**Only one** input can be positive (1) to give a positive output (1 or ON). If both are positive, the output is negative (0 or OFF)
XNOR "eXclusive Not OR	$\overline{A \oplus B}$	**All inputs** must be the same (either high or low) for a positive output (1). Otherwise, the output is negative (0 or OFF)

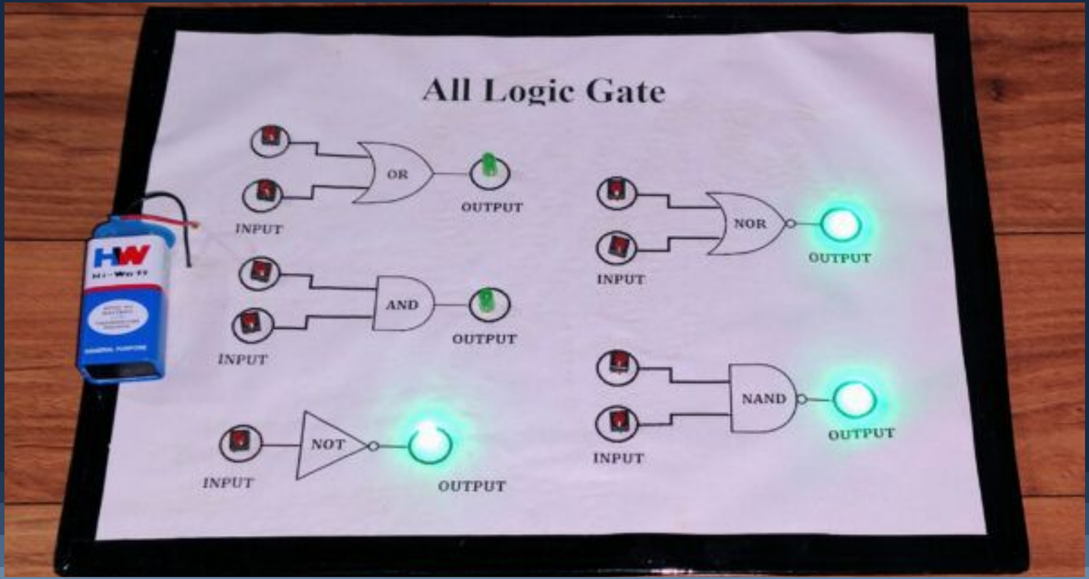
*ComputerEngineeringforBabies.com

Logic gates are the fundamental building blocks in digital electronics.

- Used to perform logical operations based on the inputs provided to it and gives a logical output that can be either high(1) or low(0).

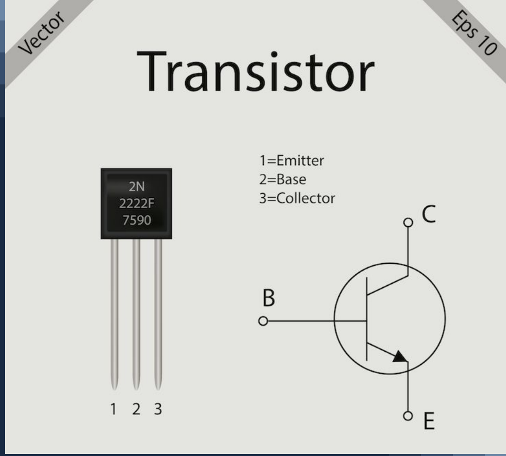
- There are basically seven main types of logic gates that are used to perform various logical operations in digital systems.
- By combining different logic gates, complex operations are performed, and circuits like flip-flops, counters, and processors are designed. In this article, we will see various types of logic gates in detail.

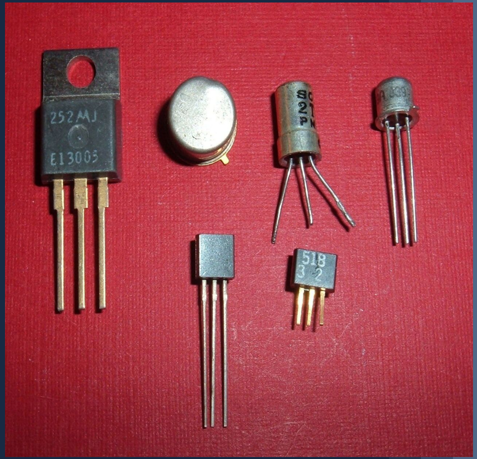
Logic gates find their uses in our day-to-day lives, such as in the architecture of our telephones, laptops, tablets and memory devices.



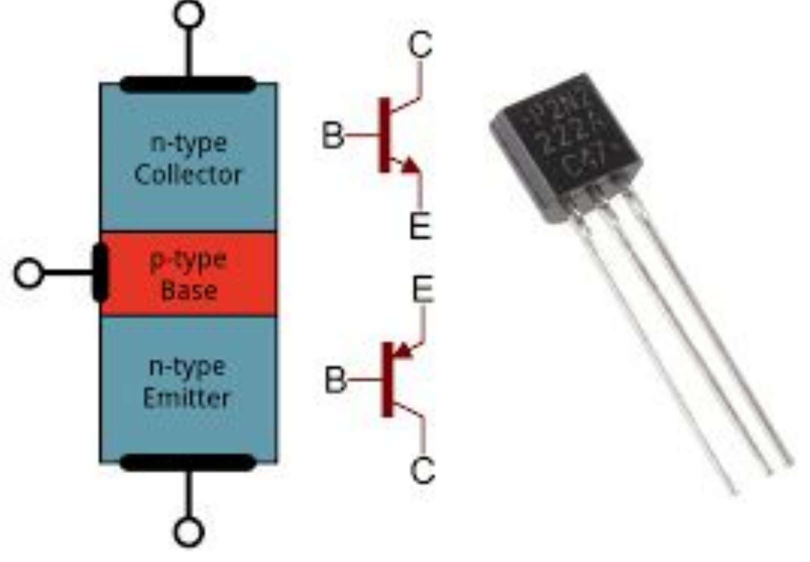
Transistor

Transistor is such a marvelous invention. A transistor is a semiconductor device that has at least three terminals and functions to amplify electrical signals and act as a switch. A transistor is usually made from silicon or another semiconductor material.





Transistors are the core components in integrated and modern microchips can contain billions of transistors, allowing devices to perform tasks at high speeds while using less power.



A transistor consists of three terminals: the base, collector, and emitter. Through these terminals, the transistor can control the flow of current in a circuit.