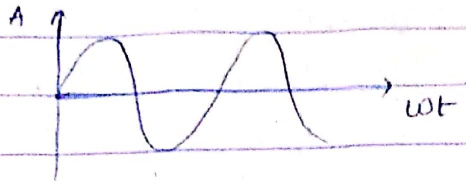
MODULE-4 : Logic Gates & Boolean Algebra:

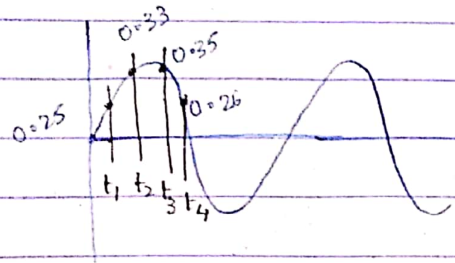
Analog signal: signal varies continuously with time.

Ex: sinusoidal signals:



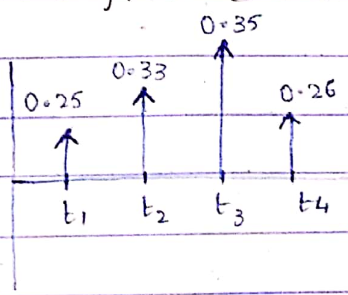
* for every moment, a value of amplitude is defined.

Discrete-time signal:



The process by which continuous signals (or analog signals) are converted into discrete-time signal is called '**SAMPLING**'

same signal can be represented as:



: Discrete-time signal

Quantisation:

The process by which continuous values are assigned values in form of '0' & '1' is called quantisation.

Digital signal: A quantised discrete-time signal is called digital signal. i.e. a discrete time signal in the form of binary bit pattern (0 & 1)

continuous signal → SAMPLER → QUANTISER → Digital signal.

* Positive ~~negat~~ logic system: ON → 1
OFF → 0

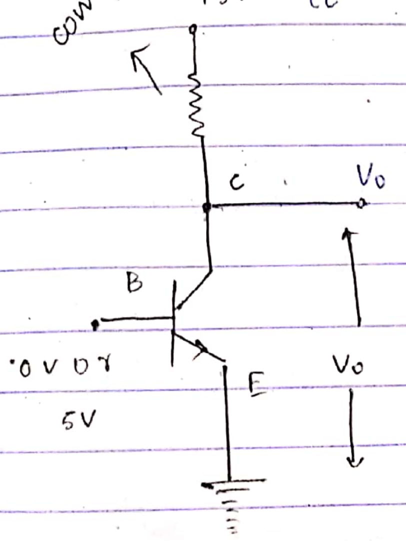
* Negative logic system: ON → 0
OFF → 1

(ON: True/High
OFF: False/Low)

7/3/19

Gate: Allows or restricts the flow of something.

complementor



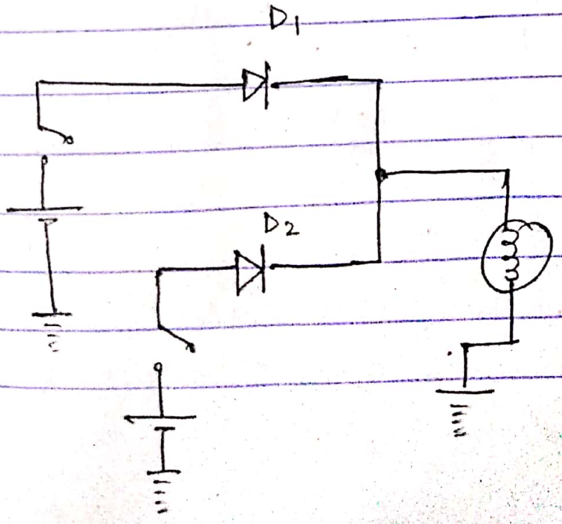
$V_0 \neq V_{BE} \neq 0$

$V_0 = V_{BE}$

When input voltage is 0V, output voltage is 5V

i.e.

Input voltage	output voltage
0 V	5 V
5 V	0 V.



Input		output
SW1	SW2	
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON.

→ allows one-way flow of input.

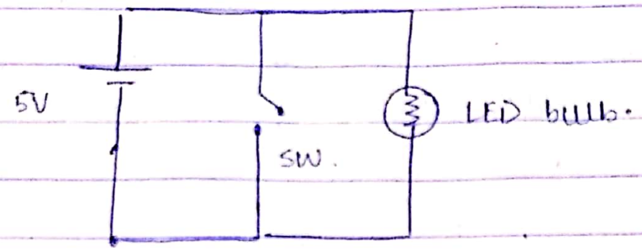
Gate:

Gate is ~~an electronic~~ a device which may have multiple inputs but has only one output.

Gate works on ~~the~~ switching property ~~of transistor~~ logical

Truth table: Represents the relation between input & output variables.

1. switching circuitory for ~~NOR~~ NOT gate:



SW	output
OPEN	LED glows
CLOSED	LED doesn't glow (due to short circuit).

Input (A)	~~(B)~~ output (Y)
0	1
1	0

} complementor action

Position of the switch is represented by input variables

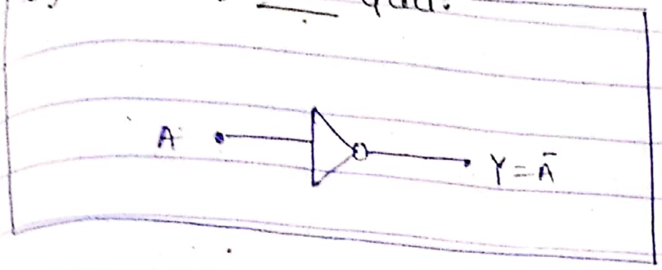
Mathematically,

A	Y
0	1
1	0

$\boxed{A = \bar{Y}}$

∴ It is not possible to remember large truth tables, ∴ some equations can be defined for each case. a truth table or any gate can be represented mathematically.

Symbol of NOT Gate:



2. AND Gate:

B	A	Y
0	0	1
0	1	0
1	0	0
1	1	1

Truth table { Input variables: A, B
output variable: Y

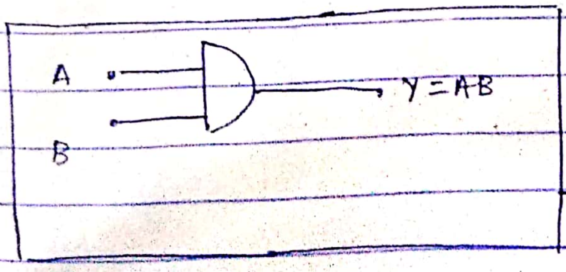
Mathematically,

$Y = (A) \text{ AND } (B)$

$\boxed{Y = A \cdot B}$

Boolean expression: $\boxed{Y = A \cdot B}$

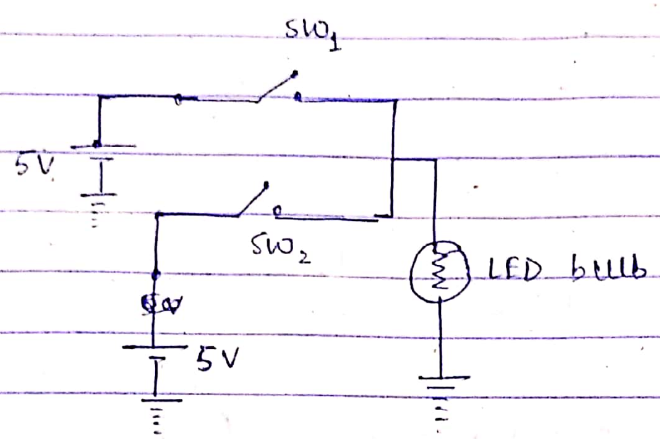
Symbol of AND Gate:



3. OR operation

B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1

: Truth table



circuit for OR Gate

Mathematically,

$Y = (A)$ OR (B)

$\boxed{Y = A + B}$: Boolean expression.

∴ The basic gates are:

AND	:	$Y = \overline{A}$
OR	:	$Y = A + B$
NOT	:	$Y = AB$

4. NAND GATE

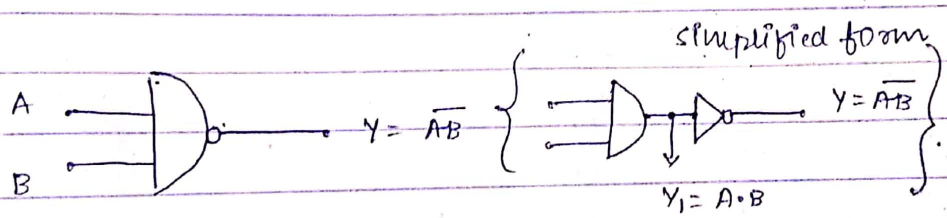
NOT + AND ⇒ complement of AND.

Boolean expression: $\boxed{Y = \overline{A \cdot B}}$

Truth table:

B	A	Y
0	0	$1 (= \bar{0})$
0	1	$1 (= \bar{0})$
1	0	$1 (= \bar{0})$
1	1	$0 (= \bar{1})$

Symbol of NAND:



1/4/19

Number System

$(x)_{(b)}$ → base

In decimal number system, $b = 10$ $[(x)_{10}]$

& In binary number system, $b = 2$ $[(x)_2]$

* Decimal number system uses: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. $(b=10)$
* Binary no. system uses: 0, 1 $(b=2)$
* Octal no. system: 0, 1, 2, 3, 4, 5, 6, 7. $(b=8)$

* Hexadecimal no. system uses: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (b=16)

Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	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
16	10000	~~18~~ 20	10
17		~~19~~ 21	11

$(10)_{16} = (?)_{10} \Rightarrow \boxed{16 = ?}$

Binary

$(1111) = (F)_{16} = (15)_{10}$

$1\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0$

Decimal

Binary	Hexadecimal
0000	0
0001	1
0011	2
0100	3
0110	4
0111	5
1000	6
1001	7
1000	8
1001	9
~~1100~~ 1010	A
~~1101~~ 1011	B
~~1110~~ 1100	C
~~1~~ 1101	D
1110	E
1111	F

Add : (4+5) in binary system:

$$(4)_{10} = (0100)_2$$

$$(5)_{10} = (0101)_2$$

$$+ \quad \overline{\qquad\qquad}$$

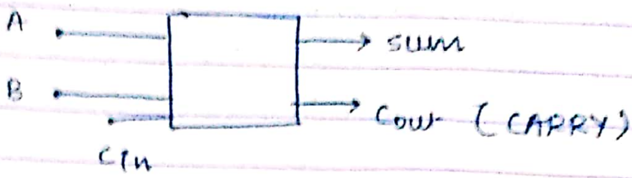
$$(9)_{10} = 201$$

↓

binary of 2 = 10

$$(9)_{10} = (1001)_2$$

Basic Gates — AND, OR, NOT
Universal Gates — NAND, NOR
Other two gates — XOR, XNOR.

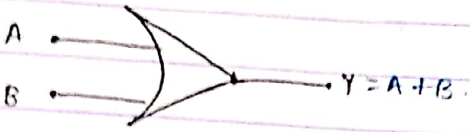


~~For a single bit adder,~~

$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

2/4/19

OR GATE:



A	B	Y	Binary OR $A+B=Y$	Binary addition
0	0	0	$0+0=0$	$0+0=0$
0	1	1	$0+1=1$	$0+1=0$
1	0	1	$1+0=1$	$1+0=1$
1	1	1	$1+1=1$	$1+1=10$

Carry = 1 Sum = 0

NAND Gate is called universal gate because it can provide the functionalities of all the 3 basic gates.

NOT GATE using NAND GATE:

To form NOT Gate from NAND Gate: one input is made 1.

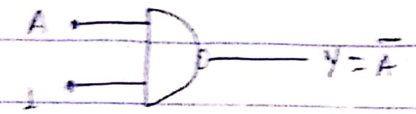
we want,

$Y = \overline{A \cdot B}$ & $Y = \overline{A}$ to be equal.

In $Y = \overline{A \cdot B}$

$\therefore$ If $B = 1$,

$Y = \bar{A}$

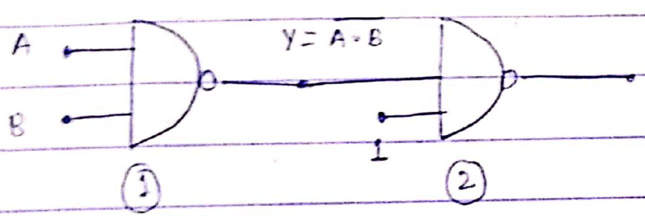


An important Result:

1. $\boxed{A \cdot 1 = A}$

AND GATE using NAND Gate:

$Y = A \cdot B$ & $Y = \overline{A \cdot B}$



NAND-1

A	B	$Y = \overline{A \cdot B}$
0	0	1
1	0	1
0	1	1
1	1	0

for NAND-2

$\underbrace{Y \quad 1}_{\text{Inputs}}$		Y'
1	1	0
1	1	0
1	1	0
0	1	1

same as $A \cdot B$

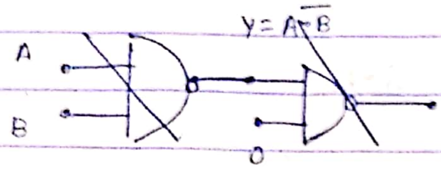
2. # Important result: $\boxed{A \cdot \bar{A} = 0}$

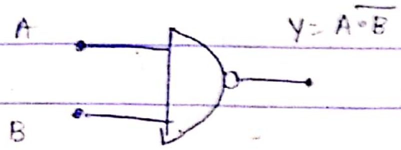
OR GATE using NAND GATE:

$Y = A + B$ & $Y = \overline{A \cdot B}$

$A + B = \overline{A \cdot B}$

A	B	$\overline{A \cdot B}$	A + B
0	0	1	0
0	1	1	1
1	0	1	1
1	1	0	1





$\bar{\bar{Y}} = \overline{\overline{A \cdot B}}$

De Morgan's theorem: $\overline{(A \cap B)} = \bar{A} \cup \bar{B}$ & $\overline{(A \cup B)} = \bar{A} \cap \bar{B}$

From De Morgan's theorem,

$\overline{(A \cdot B)} = \bar{A} + \bar{B}$

$\therefore \bar{\bar{Y}} = \overline{(\bar{A} + \bar{B})}$

$= (\bar{\bar{A}} \cdot \bar{\bar{B}})$

$= \bar{\bar{\bar{A}}} + \bar{\bar{\bar{B}}}$

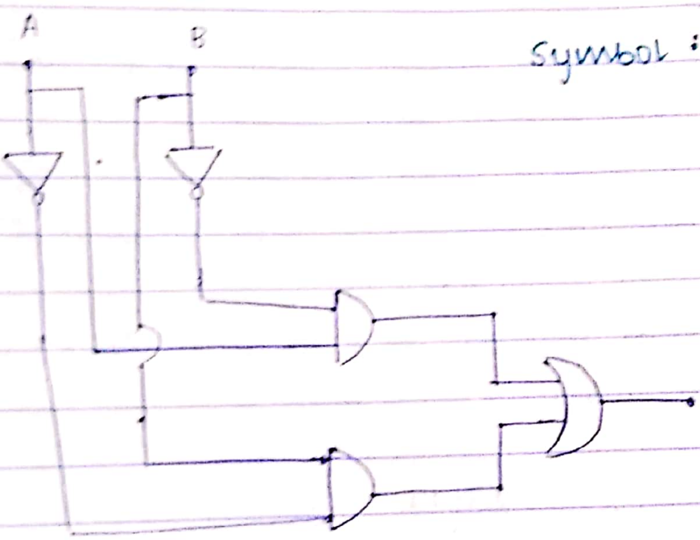
$\bar{\bar{\bar{Y}}} = \bar{A} + \bar{B}$

NOR Gate:

$\boxed{Y = \overline{A+B}}$: Boolean expression.

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XOR Gate: (Exclusive OR)



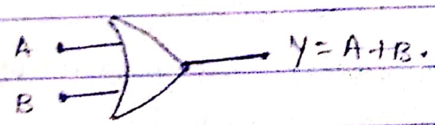
Symbol : $\oplus$

$X \oplus Y$

↓

XOR

circuit symbol:



Truth table

A	B	(A+B)	$(\bar{A}B + A\bar{B})$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0
		↓ OR	↓ XOR.

when both inputs are same, the output is '0'. It is "exclusive" because it "excludes" the case when inputs are same.

Digital circuits

combinational

* Digital circuits without memory elements. i.e. they are just a "combination" of gates.

Ex: adder / subtractor

sequential

* Digital circuits with memory elements.

(combinational circuit) + (Memory) → Sequential circuits.

Adder: A category of combinational digital circuits, which performs binary addition.

ADDER

Half-adder
Performs 2-bit addition

Full-adder
Performs 3-bit addition

Binary addition

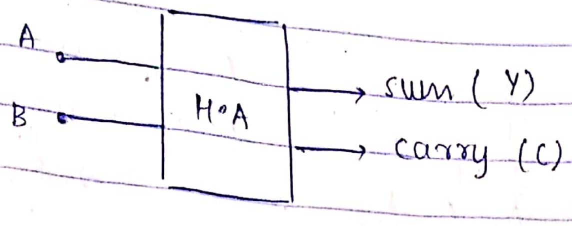
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

1 → carry, 0 → sum

Binary subtraction

0 − 0 = 0
0 − 1 = 1 (1 is borrow)
1 − 0 = 1
1 − 1 = 0

Half-adder:

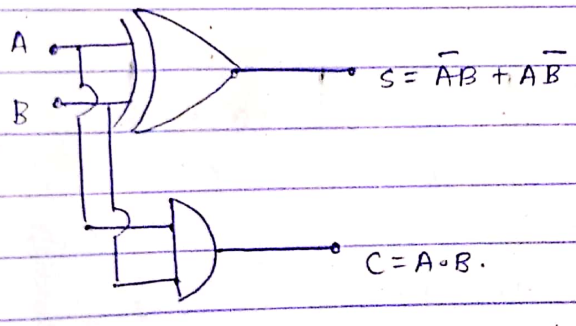


Truth table of half adder circuit:

A	B	Y	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

S or $Y = \overline{A}B + A\overline{B}$ (XOR Gate)

$C = A \cdot B.$ (AND Gate)



\# $\boxed{\begin{array}{l} A \cdot \overline{A} = 0 \\ A + \overline{A} = 1 \\ 0 + A = A. \end{array}}$

Q. $Y = A \cdot B \cdot C$

what are the duals of Y?

A —— B —— C

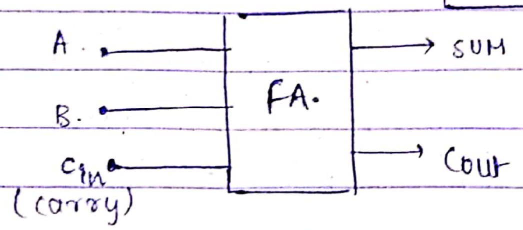
Dual of $A \cdot \bar{A} = 0$ → $\bar{A} + A = 1$

- A is changed with $\bar{A}$
- $\bar{A}$ is changed with A
- 0 is changed with 1.

A	B	C	Y	Y^D
0	0	0	0	1

∴ Dual of $A \cdot B \cdot C$ → $\bar{A} + \bar{B} + \bar{C}$ →

FULL ADDER



Truth table:

A	B	C	sum	Cout carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

Boolean expression for sum : $\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$\cancel{\bar{A}\bar{B}C} + \cancel{A\bar{B}\bar{C}} + \cancel{\bar{A}B\bar{C}} + \cancel{ABC}$

$\cancel{\overline{A\bar{B}C}}$

$\bar{A}(B \oplus C) + A$ $\Leftarrow \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$ *

★ $\underbrace{\bar{A}(\bar{B}C + B\bar{C})}_{P} + A(\bar{B}\bar{C} + BC)$

$\bar{P} = \overline{(\bar{B}C + B\bar{C})}$

$= \overline{(\bar{B}C)} \cdot \overline{(B\bar{C})}$

$= (\bar{\bar{B}} + \bar{C}) \cdot (\bar{B} + \bar{\bar{C}})$

$= (\bar{\bar{B}} + \bar{C}) \cdot (\bar{B} + \bar{\bar{C}}) = \bar{B}\bar{C} + BC$

$\cancel{(\bar{\bar{B}} \cdot \bar{C})} \cdot \cancel{X}$

$\therefore$ Expression becomes $\bar{A}P + A\bar{P} = A \oplus P$

$= A \oplus (B \oplus C)$

$\{\because P = \bar{B}C + B\bar{C}\}$

similarly, Boolean expression for carry,

$C_0 = AB\bar{C} + A\bar{B}C + \bar{A}BC + ABC$

$= A(B\bar{C} + \bar{B}C) + BC(A + \bar{A})$

$= A(B\bar{C} + \bar{B}C) + BC \quad (\because A + \bar{A} = 1)$

$A(\cancel{B} \oplus C) + \cancel{B}C$

$\cancel{= AB\bar{C} + A\bar{B}C +}$

$= AB(C+\bar{C}) + A\bar{B}C + \bar{A}BC$

$= AB + A\bar{B}C + \bar{A}BC$

$= A(B+\bar{B}C) + \bar{A}BC.$

5/4/19.

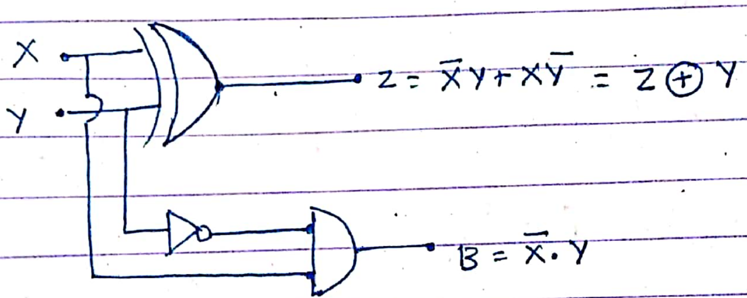
Half-subtractor:

Truth table

X	Y	Z	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Z ↓ XOR   B ↓ $\bar{X}.Y$

Circuit for Half subtractor:



If truth table is given, write the optimal boolean expression for output variables 'Z' & 'B' using the input variables 'X' and 'Y'.

So, follow the steps to design the circuit:

Step 1: Boolean expression for 'Z':

$$Z = \bar{x}y + x\bar{y}$$

$$\boxed{Z = X \oplus Y}$$

⇓

Expression cannot be minimized further

∴ It is the optimal expression.

Step 2: Boolean expression for 'B':

$$\boxed{B = \bar{X} \cdot Y}$$

Full- subtractor