DIGITAL CIRCUITS

- ❖ Basic building blocks of digital computer .
- ❖ Broadly classified as combinational & sequential.
  - ❖ In combinational circuits, output depends only on the present inputs .
  - ❖ In sequential circuits, output not only depends on present inputs but also on the previous outputs.
- ❖ Boolean algebra plays important role in the design and analysis of digital circuits .
- ❖ Boolean algebra is a branch of algebra in which values of the variable either 0 or 1.

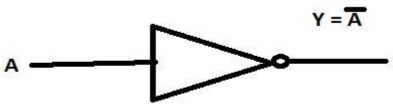
logic 1=HIGH (or) TRUE=5V

logic 0=LOW (or) FALSE=0V

- ❖ The logical functions involved in Boolean expressions are NOT,OR and AND

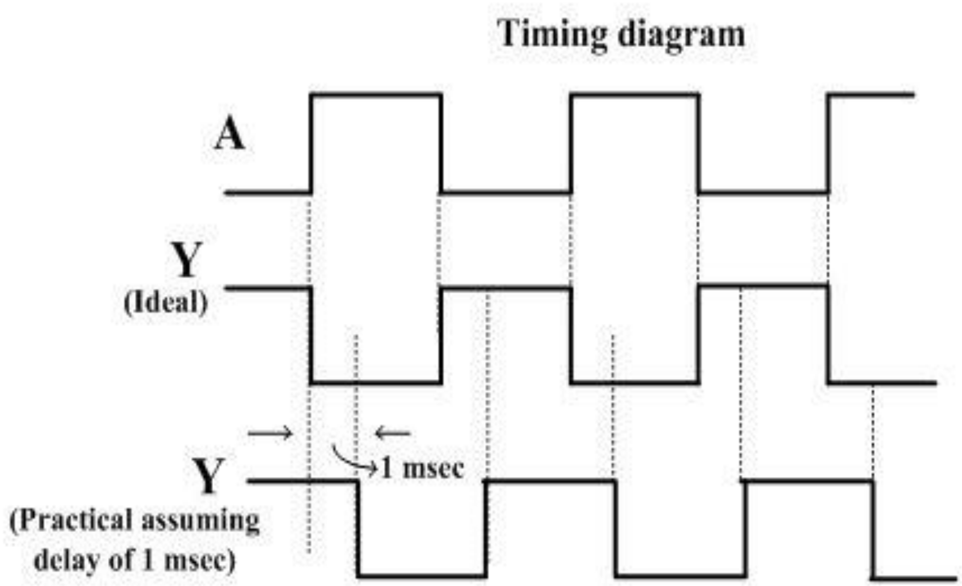
NOT Gate

- It has one input and one output signal. The output is complement of the input.



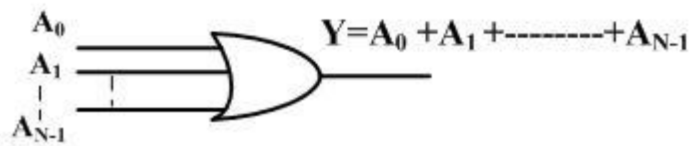
- Truth table is a table the describes output values for all the combination of input values.

A	Y
0	1
1	0



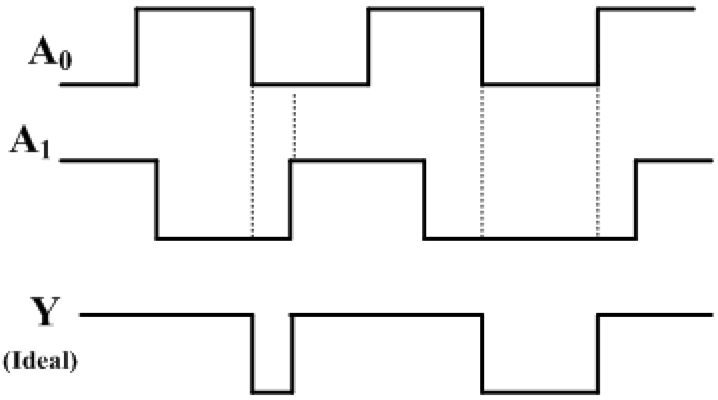
OR Gate

- It has two or more input signals but only one output signal.
- The output is HIGH, if any or all the inputs are HIGH.



Truth table of two input OR gate

A_1	A_0	Y
0	0	0
0	1	1
1	0	1
1	1	1



AND Gate

- AND gate has two or more inputs but only one output.
- Output is HIGH if all the inputs one HIGH.

Boolean laws and theorems

Laws of complementation	OR laws	AND laws
(1) $\overline{\overline{A}} = A$	(1) A+0= A	(1) A.0=0
(2) $\overline{1} = 0$	(2) A+1=1	(2) A.1=A
(3) $\overline{0} = 1$	(3) A+A=A	(3) A.A=A
	(4) $A+\overline{A}=1$	(4) $A.\overline{A}=0$

Commutative	Associative	Distributive
A+B=B+A	(A+B)+C=A+(B+C)	A(B+C)=AB+AC
AB=BA	=A+B+C	A+BC=(A+B)(A+C)
	(AB)C=A(BC)=ABC	

De Morgan´s Theorem

$$\overline{A+B+C+\cdots\cdots\cdots} = \overline{A}\ \overline{B}\ \overline{C}\cdots\cdots$$

$$\overline{ABC}\cdots\cdots\cdots = \overline{A}+\overline{B}+\overline{C}\cdots$$

Conversion of English statements to Boolean expression.

Ex:- Seat Belt warning system .

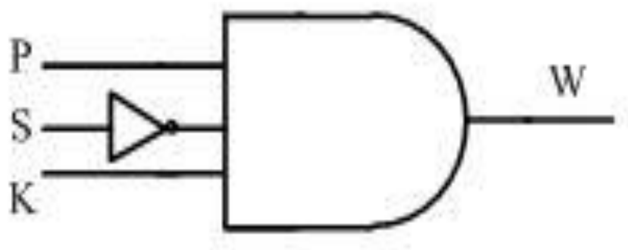
$S=1 \Rightarrow$ Seat beat fastened

$K=1 \Rightarrow$ key inserted

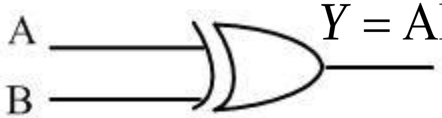
$P=1 \Rightarrow$ Person is seat

Let W= Output seat belt warning system.

$\therefore \quad W=P\overline{S}K$



Ex-OR Gate



$$Y = A\overline{B} + \overline{A}B = A \oplus B$$

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

if inputs are same output is zero
if inputs are different output is 1

Ex-NOR Gate

$$Y = \overline{A\overline{B} + \overline{A}B} = \overline{A\overline{B}}.\overline{\overline{A}B}$$

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

$$= AB + \overline{A}\,\overline{B}$$

$$= A \odot B$$

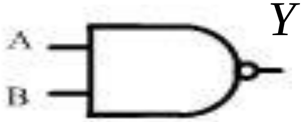
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

if inputs are same output is 1
otherwise output is zero.
Hence act as equality detector.

Universal Gates (NAND and NOR)

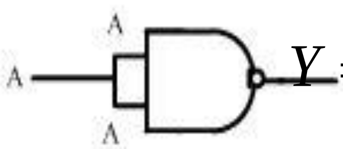
NAND Gate

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



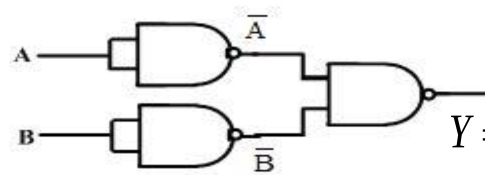
$Y = \overline{AB}$

NOT using NAND



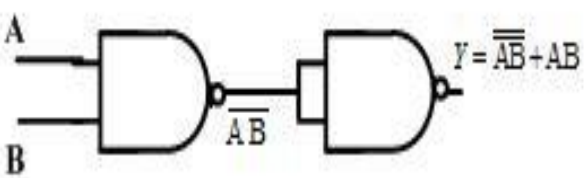
$Y = \overline{A.A} = A$

OR using NAND

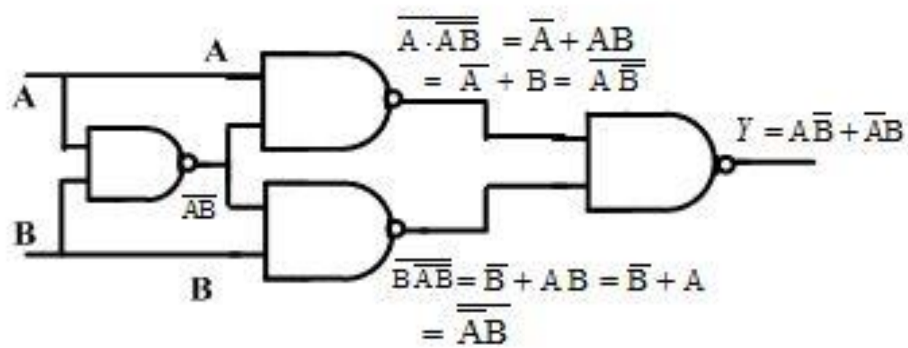


$Y = \overline{\overline{A}\,\overline{B}} = A + B$

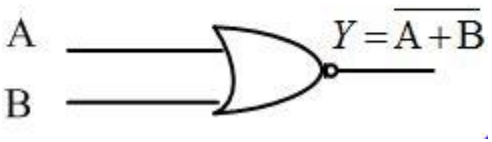
AND using NAND



Ex-OR using NAND

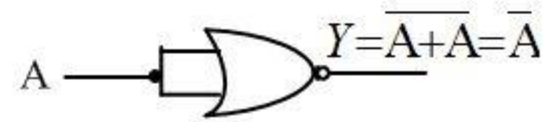


NOR Gate

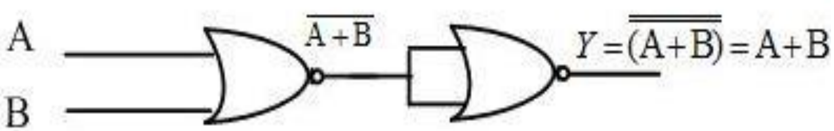


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

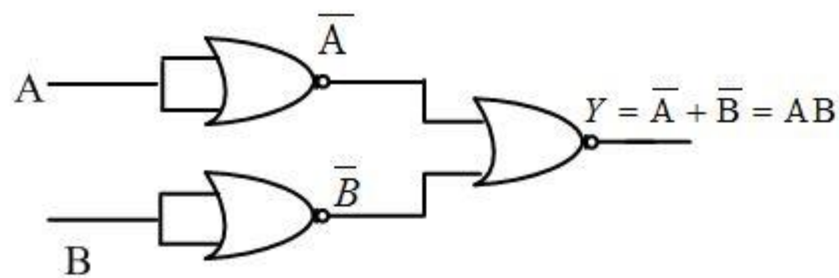
NOT using NOR



OR using NOR



AND using NOR



Ex-OR using NOR