

Lesson 01

Boolean Algebra

- Algebra associated with binary number is called "Boolean Algebra".
- Variables used in Boolean Algebra are called "Boolean Variables".

Boolean Variables

- Variable that can have only one value from 1 and 0.

ex: If x is a boolean value,

$$x = 0 \text{ or } x = 1.$$

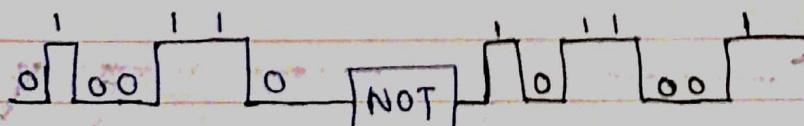
Boolean Algebra

- The two entities that 0 and 1 together with 3 operators AND, OR, NOT are called Boolean Algebra.

NOT operator, (complement)

x	$\text{NOT } x / (\bar{x})$
0	1
1	0

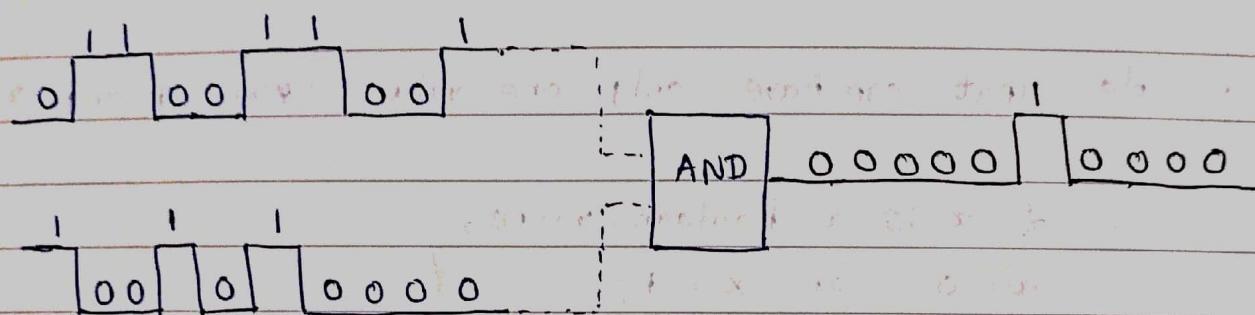
- Write the output of the following.



AND Operator, (Logical Multiplication)

X	Y	X AND Y / X.Y
0	0	0
0	1	0
1	0	0
1	1	1

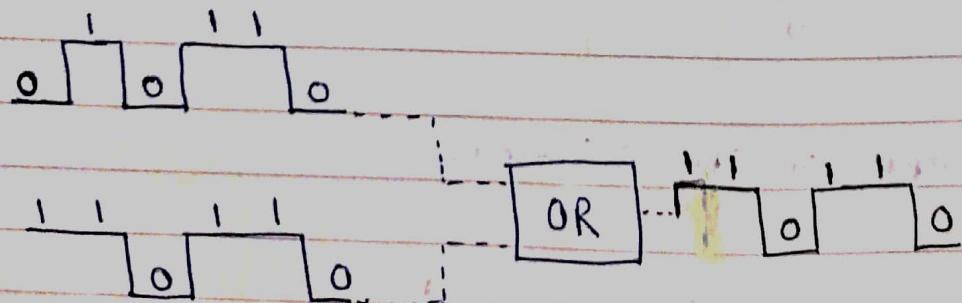
Q) write the output of following.



OR Operator, (Logical Addition)

X	Y	X OR Y / (X+Y)
0	0	0
0	1	1
1	0	1
1	1	1

Q) write the output of following.



Boolean Algebra Rules

$$01. X + 0 = X \quad \text{Additive Identity}$$

$$02. X + 1 = 1 \quad (\text{Any number added to 1 is 1})$$

$$03. X \cdot 0 = 0 \quad \text{Multiplicative Identity}$$

$$04. X \cdot 1 = X \quad \text{Multiplication by 1}$$

$$05. X + X = X \quad \text{Addition of like terms}$$

$$06. X \cdot X = X \quad \text{Multiplication of like terms}$$

$$07. X + \bar{X} = 1 \quad \text{Complement Law}$$

$$08. X \cdot \bar{X} = 0 \quad \text{Multiplication of complement}$$

$$09. \bar{\bar{X}} = X \quad \text{Double Complement}$$

10. Commutative Law

$$\text{i.) } A + B = B + A \quad \text{Addition}$$

$$\text{ii.) } A \cdot B = B \cdot A \quad \text{Multiplication}$$

11. Associative Law

$$\text{i.) } (A + B) + C = A + (B + C)$$

$$\text{ii.) } (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

⇒ These Rules
can use in
anywhere
without
proving.

12. Distributive Law

$$\text{i)} A \cdot (B+C) = A \cdot B + A \cdot C$$

$$\text{ii)} A + (B \cdot C) = (A + B) \cdot (A + C)$$

13. Redundance Law

$$\text{i)} A + A \cdot B = A$$

$$\text{ii)} A(A+B) = A$$

$$\text{14. } X + \overline{X}Y = X + Y$$

Q) Prove followings.

$$\text{i)} A + A \cdot B = A$$

L.H.S:

$$A \cdot 1 + A \cdot B$$

$$A(1+B)$$

$$A(1)$$

A // R.H.S.

$$\text{ii)} A(A+B) = A$$

L.H.S:

$$A(A+B)$$

Perfect Induction

⇒ Perfect Induction is proving a theorem by verifying every combination of values that the variable may assume.

Q) Prove the followings using perfect Induction.

$$1. x + xy = x$$

x	y	xy	x+xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

$$\text{L.H.S} = \text{R.H.S.} //$$

$$2. x(x+y) = x$$

x	y	x+y	x(x+y)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

$$\text{L.H.S.} = \text{R.H.S.} //$$

$$3. x + \bar{x}y = x+y$$

x	y	\bar{x}	$\bar{x}y$	$x+\bar{x}y$	$x+y$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

$$\text{L.H.S.} =$$

$$\text{R.H.S.} //$$

Q.) Prove the followings using boolean Rules.

1. $x + xy = x$

L.H.S:

$$x + xy$$

$$x(1+y)$$

$$x(1+y)$$

$$x \cdot 1$$

$$x \leftarrow R.H.S //$$

$$L.H.S. = R.H.S. //$$

2. $x(x+y) = x$

L.H.S:

$$x(x+y)$$

$$x \cdot x + x \cdot y$$

$$x + x \cdot y$$

$$x(1+y)$$

$$x \cdot 1$$

$$x \leftarrow R.H.S.$$

$$L.H.S. = R.H.S. //$$

3. $x + \bar{x}y = x+y$

L.H.S:

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

$$x + \bar{x} \cdot x+y$$

$$1 \cdot x+y$$

$$x+y \leftarrow R.H.S.$$

$$L.H.S. = R.H.S. //$$

Q) Using Boolean Algebra, simplify the following expressions.

$$1. A = \bar{x}y\bar{z} + \bar{x}yz + x\bar{y}z + xy\bar{z}$$

$$A = \bar{x}y(z + \bar{z}) + xz(\bar{y} + y)$$

$$A = \bar{x}\bar{y} + xz //$$

$$2. x = AB + A(B+C) + B(B+C) \quad B.B + BC$$

$$\bullet x = AB + A \cdot B + A \cdot C + B // B + BC$$

$$x = A(B+B) + A \cdot C + B$$

$$x = AB + AC + B$$

$$\overbrace{AB+AC}^{\leftarrow} + B$$

$$AB + B$$

$$x = B(A+1) + AC // = B(B+AC) // z = B(A+1) + AC$$

$$B+1 \cdot \overbrace{B+AC}^{\leftarrow}$$

$$3. A = x + \bar{y} + \bar{x}y + (x + \bar{y}) \cdot \bar{x}y$$

$$\overbrace{B+AC}^{\leftarrow}$$

$$A = x + \bar{y} + \bar{x}y + x \cdot \bar{x} \cdot y + \bar{y} \cdot \bar{x} \cdot y$$

$$A = x + \bar{y} + \bar{x}y + 0 \cdot y + \bar{x} \cdot 0$$

$$A = x + \bar{y} + \bar{x}y // \leftarrow \text{Distributive law apply func.}$$

$$4. z = ABC [AB + \bar{c}(BC + AC)]$$

$$z = ABC [AB + \bar{c} \cdot BC + \bar{c} \cdot AC]$$

$$z = ABC [AB + 0 + 0]$$

$$z = ABC [AB] \rightarrow AB(c) \rightarrow ABC //$$

$$5. y = (A + \bar{A})(AB + A\bar{B}\bar{C})$$

$$y = 1 (AB + A\bar{B}\bar{C})$$

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

$$y = AB(1 + \bar{C})$$

$$y = AB //$$

$$6. y = (A + \bar{B})(A + C)$$

$$y = A \cdot A + A \cdot C + A \cdot \bar{B} + \bar{B} \cdot C$$

$$y = A + A \cdot C + A \cdot \bar{B} + \bar{B} \cdot C$$

$$y = A \underbrace{(1 + C + \bar{B})}_{1} + \bar{B} \cdot C$$

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

7. $A = AB + ABC + ABCD + ABCDE$

$$Z = AB(1+C) + ABCD(1+E)$$

$$Z = AB + ABCD$$

$$Z = AB(1+C \cdot D)$$

$$Z = AB(1 \cdot D)$$

$$Z = ABD //$$

Q) Prove the followings.

i) $\bar{X}Y + \bar{Y}Z + YZ = \bar{X}Y + Z$

L.H.S:

$$\bar{X}Y + \bar{Y}Z + YZ$$

$$\bar{X}Y + Z(\bar{Y} + Y)$$

$$\bar{X}Y + Z //$$

R.H.S.

ii) $XZ + \bar{X}YZ = XZ + YZ$

L.H.S:

$$XZ + \bar{X}YZ$$

$$Z(X + \bar{X}Y)$$

$$Z(X + \bar{X})(\bar{X} + Y)$$

$$Z(\bar{X} + Y)$$

$$\bar{X}Z + YZ$$

R.H.S.

iii) $A\bar{C}\bar{D} + A\bar{B}D + AC\bar{D} = A(\bar{B} + \bar{D})$

L.H.S:

$$A\bar{C}\bar{D} + A\bar{B}D + AC\bar{D}$$

$$A\bar{D}(\bar{C} + C) + A\bar{B}D \quad A(\bar{C}\bar{D} + \bar{B}D + A\bar{D})$$

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

$$A(\bar{C} \cdot C + \bar{D} \cdot D + \bar{B} + \bar{D})$$

$$A(0 \cdot 0 + \bar{B} + \bar{D})$$

$$A(\bar{B} + \bar{D}) \text{ R.H.S.}$$

iv) $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{C} + A\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{C} = \bar{A}$

L.H.S: $\bar{A}(\bar{B}\bar{C} + \bar{C} + \bar{A}\cdot\bar{C} + \bar{C})$ $C, \bar{C} = 0,$

$$\bar{A}(\bar{B}\cdot 1 + \bar{B}\cdot 0 + \bar{C})$$

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

X $\bar{A}(1\cdot 1)$

De Morgan's Theorem

(i) $\overline{X+Y} = \bar{X}\cdot\bar{Y}$

(ii) $\overline{X\cdot Y} = \bar{X} + \bar{Y}$

Q1) Apply De Morgan's Theorems to the followings.

i) $Z = \overline{A+B}$

$$Z = \bar{A} \cdot \bar{B}$$

$$Z = \bar{A} \cdot B //$$

ii) $Z = \overline{AB + CD}$

$$Z = \overline{AB} \cdot \overline{CD}$$

$$Z = \bar{A} + \bar{B} + \bar{C} + \bar{D}$$

iii) $Z = \overline{\overline{AB} \cdot \overline{CD}}$

$$Z = \bar{A} + \bar{B} \cdot \bar{C} + \bar{D}$$

$$Z = \bar{A} \cdot \bar{B} + \bar{C} \cdot \bar{D}$$

Q2) Apply De Morgan's Theorems and simplify the followings.

i) $Z = \overline{(A+\bar{B}) \cdot (\bar{C}+D)}$

$$Z = \overline{(A+\bar{B})} + \overline{(\bar{C}+D)}$$

$$Z = \bar{A} \cdot \bar{B} + \bar{C} \cdot \bar{D}$$

$$Z = \bar{A} \cdot B + C \cdot \bar{D}$$

$$Z = \bar{A}B + C\bar{D} //$$

$$ii) z = \overline{x} \cdot \bar{y} \cdot (\bar{w} + \bar{y})$$

$$z = \overline{x} \cdot \bar{y} \cdot (\bar{\bar{w}} \cdot \bar{\bar{y}})$$

$$z = \overline{x} \cdot \bar{y} + (\bar{\bar{w}} \cdot \bar{\bar{y}})$$

$$z = \overline{x} + \bar{\bar{y}} + \bar{\bar{w}} + \bar{\bar{y}}$$

$$z = \bar{x} + \bar{w} + y + \bar{y}$$

$$z = \bar{x} + \bar{w} + 1$$

$$z = 1 //$$

$$iii) y = A + \overline{B \cdot \bar{C}} + C \cdot D + \overline{\bar{B} \cdot C}$$

$$y = \overline{A + \bar{B} + \bar{\bar{C}}} + C \cdot D + \overline{\bar{B} + \bar{C}}$$

$$y = \overline{A + \bar{B} + C + C \cdot D} + \overline{\bar{B} \cdot \bar{C}}$$

$$y = \bar{A} \cdot \bar{\bar{B}} + \bar{C} + \bar{D} + B + C$$

$$y = \bar{A} \cdot B + \bar{C} + \bar{D} + B \cdot C$$

$$y = \bar{A} \cdot B + B + \bar{C} \cdot C + \bar{D}$$

$$y = \bar{A} \cdot B + 0 + \bar{D}$$

$$y = \bar{A} \cdot B + \bar{D} //$$

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

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

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

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

$$Y = \overline{\bar{A} + A + 0} \cdot \overline{\bar{B} + A + B}$$

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

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

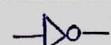
$$Y = 0 //$$

Lesson 02

Basic Logic Gates

- » It's a hardware component that consists of integrated circuit.
- » It is a digital circuit that performs common logical / Boolean functions.
- » There are 3 main basic logic gates.
 - 01.) NOT Gate
 - 02.) OR Gate
 - 03.) AND Gate.

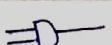
NOT Gate (Inverter)

- » Performs logical complementation.
- » Symbol : 

» Truth Table :

x	y
0	1
1	0

AND Gate

- » Performs logical multiplication.
- » Symbol : 

» Truth table :

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

→ Performs logical Addition.

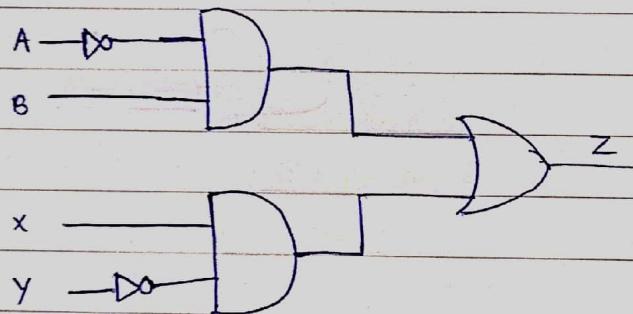
→ Symbol : 

→ Truth table :

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

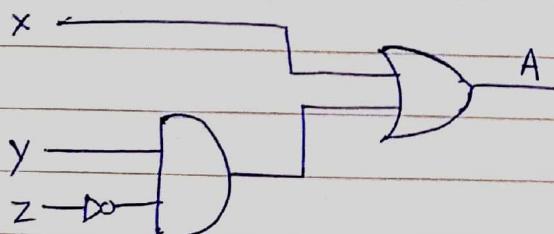
Q.) Implement the circuit for the following expression using AND, OR, NOT Gates.

$$z = \bar{A}B + \bar{x}\bar{y}$$

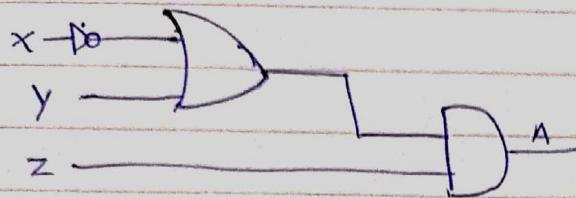


Q.) Implement the following boolean expressions using basic logic gates

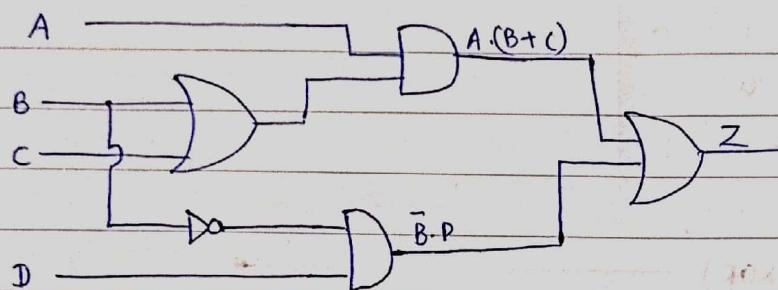
i) $A = x + y\bar{z}$



$$\text{ii) } A = (\bar{x} + y) \cdot \bar{z}$$



$$\text{iii) } Z = A(B+C) + \bar{B} \cdot D$$



other logic gates

- There are 4 other logic gates which has designed from basic logic gates.

NAND Gate

- Performs NOT(AND) operation.

Symbol: 

- Truth Table:

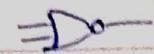
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$$Z = \overline{X \cdot Y}$$

$$Z = \text{NOT}(X \cdot Y)$$

NOR Gate

→ performs NOT(OR) operation.

→ symbol : 

→ Truth table :

x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

$$z = \overline{x+y}$$

$$z = (\text{NOT } x + y)$$

Exclusive OR Gate (XOR)

→ Perform OR operation when only one input is true.

→ symbol : 

→ Truth table :

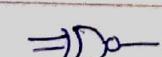
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

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

$$z = x \oplus y$$

Exclusive NOR Gate (XNOR)

→ When both inputs are same output will be true.

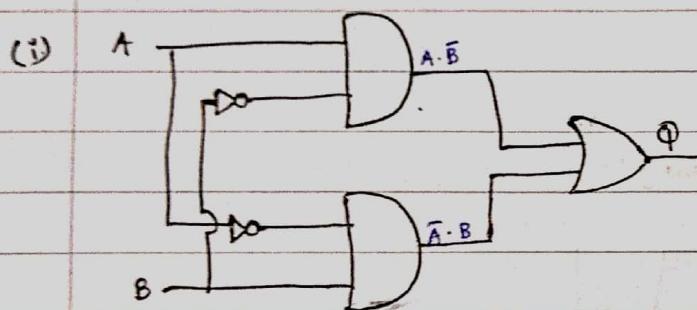
→ symbol : 

→ Truth table : $z = \bar{x}\bar{y} + xy$

$$\overline{x \oplus y}$$

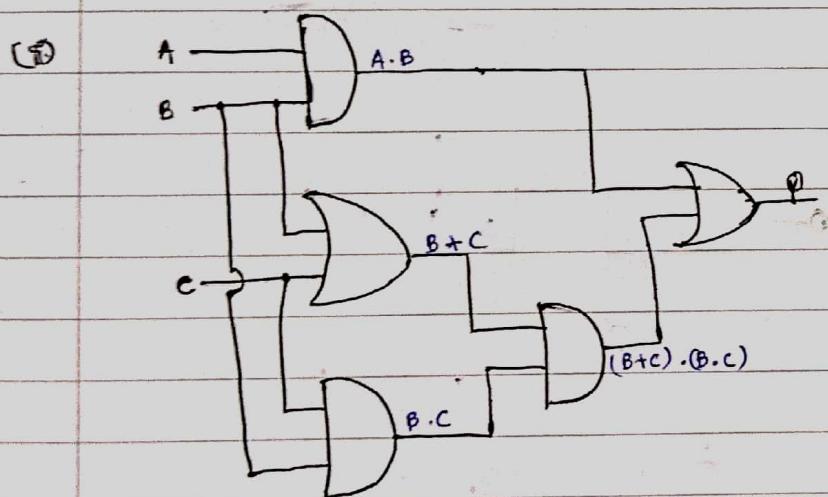
x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

Q) Write the logic expression for the output of each of the following circuits.

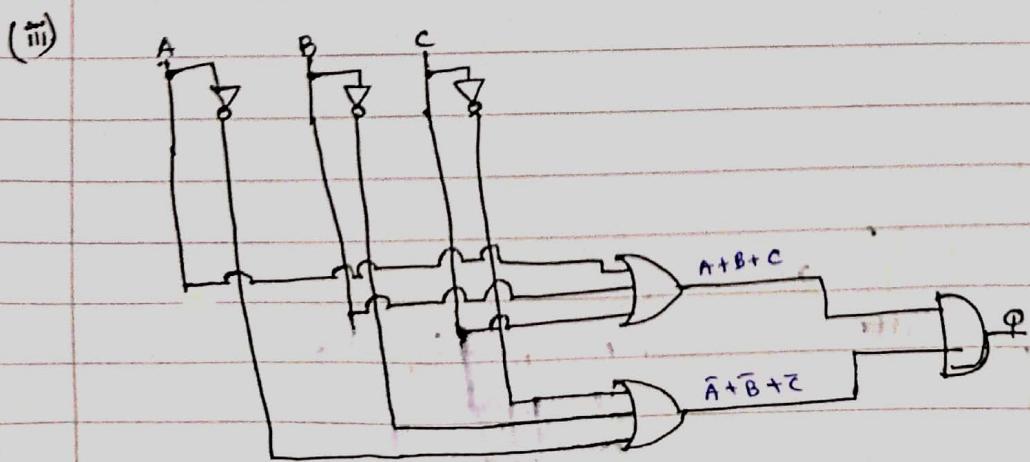


$$Q = A \cdot B + \bar{A} \cdot B$$

$$Q = A \oplus B$$

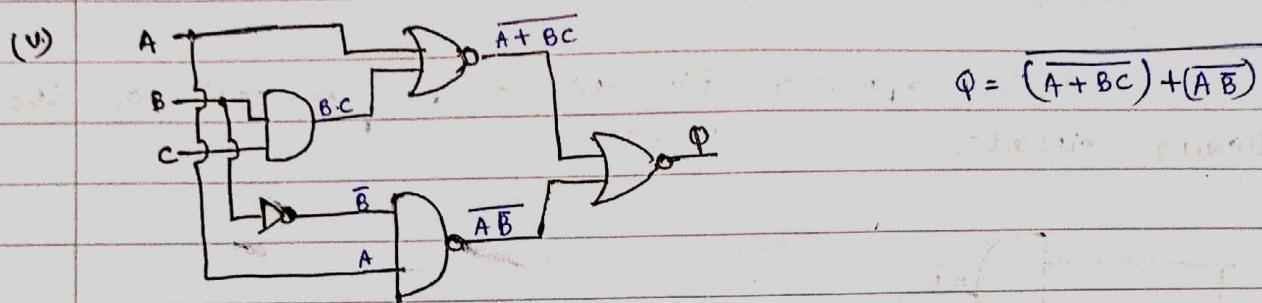
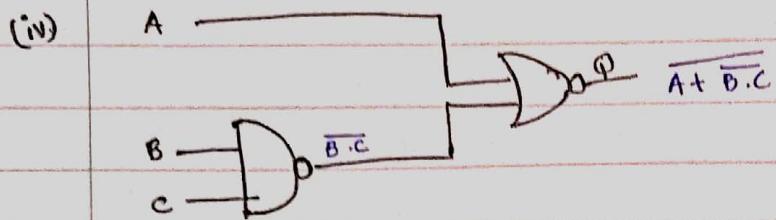


$$Q = (A \cdot B) + [(B+C) \cdot (B \cdot C)]$$

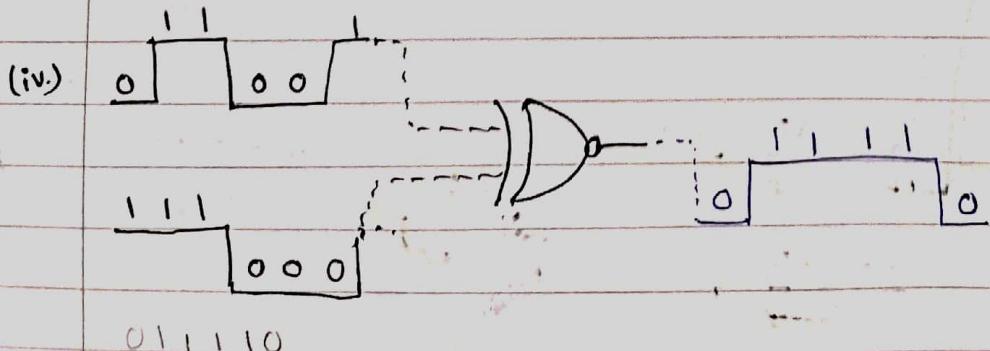
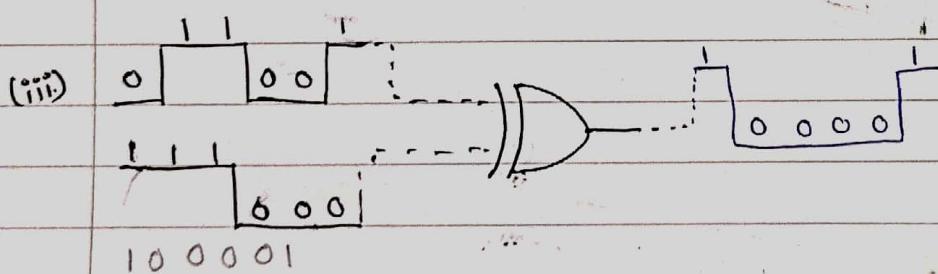
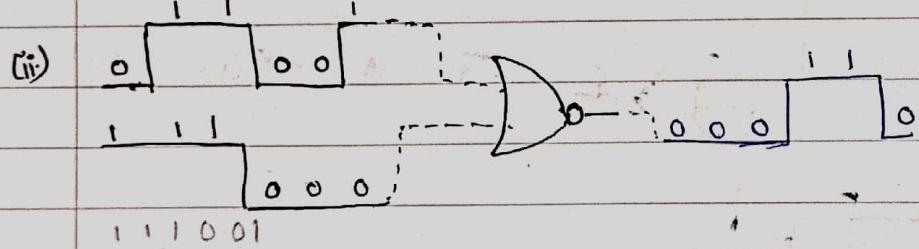
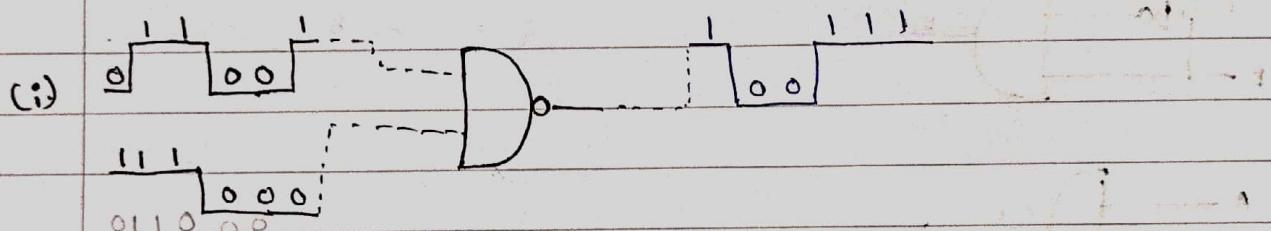


$$Q = (A+B+C) .$$

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



Q) Write the output of followings.



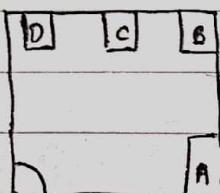
Lesson 03.

Logic circuit Design

Q) When designing a logic circuit, we work with:

- Various states of **input** combinations of a circuit.
- The desired **output** for each input combination.

Q) Construct the truth table of following logic.



⇒ If any two doors are open bulb should be ON.

⇒ If door A is open bulb should be open.

Door open - 1 Door close - 0.

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Steps in designing a logic circuit

01. Construct a truth table

02. Derive an algebraic expression for each output.

→ SOP Method

→ POS Method

03. Reduce the output expression as much as possible.

→ Algebraically

→ K-Maps

→ Tabular Method.

04. Implement the circuit using gates.

Binary Adders

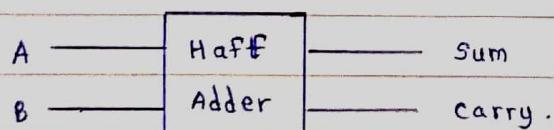
→ We use Adders to add binary numbers through circuits. There are two types of Adders,

01. Half Adder

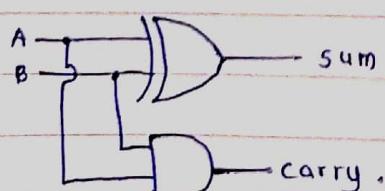
02. Full Adder.

Binary half Adder

→ It is a logic circuit for the addition of two, 1-bit binaries.



→ Logic circuit :



→ use to add single bit numbers

→ it does not take carry from previous sum.

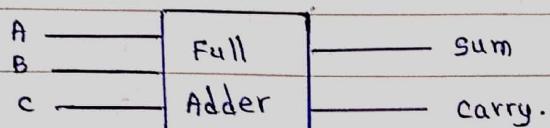
Binary Addition.

⇒ Truth table:

	A	B	Carry	Sum	sum = $\bar{A}B + A\bar{B}$
$0+0 = 0 - 0$	0	0	0	0	$\therefore = A \oplus B$
$0+1 = 1 - 0$	0	1	0	1	
$1+0 = 1 - 0$	1	0	0	1	
$1+1 = 0 - 1$	1	1	1	0	carry = AB .

Binary Full Adder

⇒ It is a logic circuit that can add three, 1-bit binary numbers.

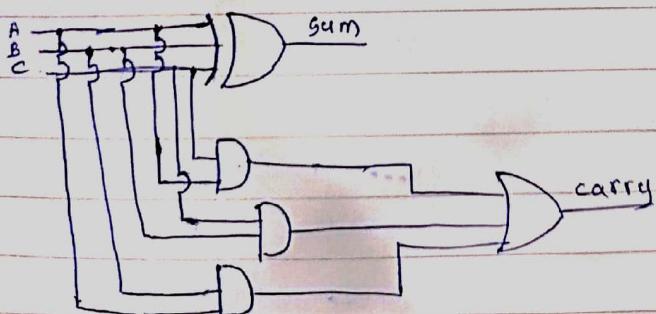


⇒ Truth Tables:

A	B	C	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$\text{sum} = A \oplus B \oplus C$$

$$\text{carry} = AB + BC + AC$$



Q1) Design a logic circuit to add two, 2-bit binary numbers.
 Implement the circuit using logic gates.

	2^3	2^2	2^1	2^0				
	A	B	C	D	carry ₁	carry ₀	sum	$No1 + No2$
No1 →	0	0	0	0	0	0	0	maximum number that can display is 6 (110)
	0	0	0	1	0	0	1	
	0	0	1	0	0	1	0	
	0	0	1	1	0	1	1	
	0	1	0	0	0	0	1	4 Inputs
	0	1	0	1	0	1	0	3 outputs
	0	1	1	0	0	1	1	
a	1	1	1	1	1	0	0	
	1	0	0	0	0	1	0	
	1	0	0	1	0	1	1	
	1	0	1	0	1	0	0	
	1	0	1	1	1	0	1	
	1	1	0	0	0	1	1	
	1	1	0	1	1	0	0	
	1	1	1	0	1	0	1	
↑	1	1	1	1	1	1	0	

Lesson 04

Karnomaps and Its Usage

- A boolean expression can be represented graphically using K-maps, by the use of binaries.

2-variable expression

	x			
	A	0	1	
0	$\bar{A} \bar{B}$	$A \bar{B}$		cell
1	$\bar{A} B$	$A B$		

$$\text{example: } x = A \bar{B} + \bar{A} B$$

	x			
	B	0	1	
0	0	0	1	
1	1	1	0	

3-variable expression

	x	AB	00	01	11	10	
	C	0	$\bar{A} \bar{B} \bar{C}$	$\bar{A} B \bar{C}$	$A \bar{B} \bar{C}$	$A \bar{B} C$	
		1	$\bar{A} \bar{B} C$	$\bar{A} B C$	$A \bar{B} C$	$A \bar{B} \bar{C}$	

$$\text{Example: } x = \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + A \bar{B} C.$$

	x	AB	00	01	11	10	
		0	0	0	0	1	
		1	1	0	1	0	

4-Variable expression

$\bar{C} \bar{D}$	00	01	11	10
00	$\bar{A} \bar{B} \bar{C} \bar{D}$	$\bar{A} \bar{B} \bar{C} D$	$A \bar{B} \bar{C} \bar{D}$	$A \bar{B} \bar{C} D$
01	$\bar{A} \bar{B} C \bar{D}$	$\bar{A} B \bar{C} D$	$A B \bar{C} \bar{D}$	$A B \bar{C} D$
11	$\bar{A} B C \bar{D}$	$\bar{A} B C D$	$A B C \bar{D}$	$A B C D$
10	$\bar{A} B C \bar{D}$	$\bar{A} B C D$	$A B C \bar{D}$	$A B C D$

example: $x = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} B C \bar{D} + A B \bar{C} \bar{D} + A \bar{B} C \bar{D}$

$\bar{C} \bar{D}$	00	01	11	10
00	0	0	1	0
01	1	0	0	0
11	0	1	0	0
10	0	0	0	1

Task: Draw K-Maps for following SOP expressions.

i) $z = \bar{A} B \bar{C} + A \bar{B} C + A B C$

$\bar{C} \bar{D}$	00	01	11	10
0	0	1	0	0
1	0	0	1	1

ii) $z = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} B \bar{C} D + \bar{A} B C \bar{D} + A \bar{B} C D$

$\bar{C} \bar{D}$	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	0	1
10	0	1	0	0

Task : Draw K-Maps for the following expressions.

i) $Z = \bar{A}C + A\bar{B}$

AB	00	01	11	10
C	0	0	0	1
	1	1	0	0

ii) $Z = B + \bar{A}\bar{B}C$

AB	00	01	11	10
C	0	1	1	0
	1	1	1	0

iii) $Z = \bar{A}\bar{C}\bar{D} + BC$

CD	00	01	10	11
00	1	1	0	0
01	0	0	1	0
11	0	1	1	0
10	0	1	1	0

iv) $Z = A + \bar{B}\bar{C}\bar{D}$

CD	00	01	11	10
00	1	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

Simplification of Boolean Expressions Using K-Map Methods.

- ⇒ Sub cube: A set of 2^n adjacent cells is referred to as a sub cube.
- ⇒ Maximal sub cube: Largest possible sub cube for a cell.
- ⇒ Minimal Map: Map with minimum number of maximal sub cubes that cover all the minterms of given expression.
- ⇒ Example: $X = \bar{A}BC + A\bar{B}C + A\bar{B}\bar{C} + ABC$

	00	01	11	10
0				1
1		1	1	1

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

Q) Simplify the following K-Maps. (Map)

\bar{Y}	00	01	11	10
0	0	1	0	0
1	0	1	1	0

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

\bar{Y}	00	01	11	10
0	0	1	0	0
1	0	0	1	1

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

\bar{Y}	00	01	11	10
0	1	0	0	1
1	0	0	0	1

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

y	$\bar{A}B$	00	01	11	10
0	1	1	1	1	1
1	0	0	0	1	1

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

y	$\bar{A}B$	00	01	11	10
0	1	1	0	0	0
1	1	1	1	1	0

$$y = \bar{A} + BC$$

y	$\bar{A}B$	00	01	11	10
00	0	0	0	0	0
01	0	0	1	1	0
11	1	0	0	1	1
10	1	0	0	1	1

$$y = \bar{B}C + AD$$

y	$\bar{A}B$	00	01	11	10
00	0	0	0	0	0
01	0	0	1	1	0
11	1	1	1	1	1
10	1	1	1	1	1

$$y = C + BD$$

y	$\bar{A}B$	00	01	11	10
00	1	0	0	1	0
01	0	0	0	0	0
11	1	1	1	0	0
10	1	1	0	1	1

$$y = \bar{B}\bar{D} + \bar{A}C + BCD$$

y	$\bar{A}B$	00	01	11	10
00	1	1	1	0	0
01	1	1	1	0	0
11	0	0	1	1	1
10	0	0	1	1	1

$$y = \bar{A}\bar{C} + AC + B\bar{C}$$

To get the simplified pos expression

- ⇒ steps:
 - i) Draw the K-map for X
 - ii) Get the minimal map for \bar{X} (cover all 0's)
 - iii) Get \bar{X} in SOP form.
 - iv) Compliment both sides. continue with de morgan's laws

⇒ Example : $y = A\bar{B}\bar{C} + \bar{A}BC + ABC$

c ↓	00	01	11	10
	0	0	0	1
1	0	1	1	0

$$\bar{y} = \bar{A}\bar{B} + B\bar{C} + \bar{B}C \text{ (SOP)}$$

(complement both sides)

$$\bar{y} = \overline{\bar{A}\bar{B} + B\bar{C} + \bar{B}C}$$

$$y = \overline{\bar{A}\bar{B}} \cdot \overline{B\bar{C}} \cdot \overline{\bar{B}C}$$

$$y = \overline{\bar{A} + \bar{B}} \cdot \overline{B + \bar{C}} \cdot \overline{\bar{B} + C}$$

$$y = (A+B)(\bar{B}+C)(B+\bar{C}) \leftarrow (\text{POS})$$

Q) Get the simplified pos expressions from the following k-maps.

i) Z

x_3	x_1x_2	00	01	11	10
		0	0	0	1
1	1	1	1	1	0
		1	1	1	0

$$\bar{Z} = \overline{x_1} \overline{x_3} + x_1 \overline{x_2}$$

$$\bar{Z} = \overline{x_1} \overline{x_3} + \overline{x_1} \overline{x_2}$$

$$Z = \overline{x_1} \overline{x_3} \cdot \overline{x_1} \overline{x_2}$$

$$Z = (\overline{x_1} + x_3)(\overline{x_1} + x_2)$$

$$\bar{Z} = x_1 \overline{x_3} + x_1 \overline{x_2}$$

$$\bar{Z} = \overline{x_1} \overline{x_3} + x_1 \overline{x_2}$$

$$Z = \overline{x_1} \overline{x_3} \cdot x_1 \overline{x_2}$$

$$Z = (\overline{x_1} + \overline{x_3})(\overline{x_1} + \overline{x_2})$$

$$Z = (\overline{x_1} + x_3)(\overline{x_1} + x_2)$$

ii) x_3x_2

x_3x_2	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	0	1
10	1	1	1	1

$$\bar{z} = \bar{x}_3\bar{x}_4 + \bar{x}_2\bar{x}_3 + x_1x_2x_3x_4$$

$$\bar{z} = \overline{\bar{x}_3\bar{x}_4 + \bar{x}_2\bar{x}_3 + x_1x_2x_3x_4}$$

$$z = \overline{\bar{x}_3\bar{x}_4} \cdot \overline{\bar{x}_2\bar{x}_3} \cdot \overline{x_1x_2x_3x_4}$$

$$z = (\bar{x}_3 + \bar{x}_4)(\bar{x}_2 + \bar{x}_3)(\bar{x}_1 + x_2x_3x_4)$$

$$z = (x_3 + x_4)(x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + x_4) //$$

22/11/2022.

⇒ Simplify x, y, z given by the following truth table using k-Maps

A B C	X Y Z
0 0 0	1 0 0
0 0 1	1 0 0
0 1 0	0 1 1
0 1 1	0 1 1
1 0 0	0 0 1
1 0 1	1 1 1
1 1 0	1 1 0
1 1 1	1 0 1

⇒ To get the equation, need to draw 3 k-Maps.

x

	00	01	11	10
0	0	0	1	0
1	1	0	0	1

y

	00	01	11	10
0	0	0	1	0
1	0	1	0	1

z

	00	01	11	10
0	0	0	0	1
1	0	1	1	1

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

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

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

Don't care conditions.

⇒ The value of the function at "Don't care" conditions, is either not significant or choose the A, B, C values that will never occur.

⇒ Don't cares can be either included or excluded in a maximal sub cube.

EX 01: $\begin{array}{c} Y \\ \diagdown AB \\ C \end{array}$

	00	01	11	10
0	1	1	0	X
1	0	0	1	X

Excluded from a maximal sub cube
 Included from a maximal sub cube.

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

Q) Find the simplified SOP expression.

i) $\begin{array}{c} Z \\ \diagdown AB \\ C \end{array}$

	00	01	11	10
0	X	0	1	0
1	1	1	X	0

$$Z = \bar{A}C + AB$$

ii) $\begin{array}{c} Z \\ \diagdown AB \\ CD \end{array}$

	00	01	11	10
00	0	0	0	1
01	0	X	1	1
11	0	1	X	0
10	X	0	0	0

$$Z = BD + A\bar{B}\bar{C}$$

iii) $\begin{array}{c} Z \\ \diagdown AB \\ CD \end{array}$

	00	01	11	10
00	1	0	0	X
01	1	X	0	0
11	X	0	1	X
10	1	0	0	1

$$Z = \bar{A}\bar{B} + \bar{B}\bar{C} + ACD$$

29/11/2022.

Q) Get the simplified POS expression.

$\begin{array}{c} Z \\ \diagdown AB \\ CD \end{array}$

	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	0	0	X	0
10	1	1	X	1

$$\begin{aligned} Z &= \bar{B}\bar{C} + CD \\ Z &= \overline{\bar{B}\bar{C} + \bar{C}\bar{D}} \\ Z &= (\bar{B}\bar{C}) (\bar{C}\bar{D}) \\ Z &= (B+C) (\bar{C}+\bar{D}) \end{aligned}$$

Minterm & Maxterms.

$$Y = \sum m(3, 4)$$

$$\text{Minterm } 3 \rightarrow \bar{A}BC$$

$$\text{Minterm } 4 \rightarrow A\bar{B}\bar{C}$$

$$\text{Maxterm} \rightarrow A + \bar{B} + \bar{C}$$

$$\text{Maxterm} \rightarrow \bar{A} + B + C$$

$$\text{EX: } z = \overline{\Pi M}(0, 2, 4, 5)$$

z is a function of A, B, C .

$$M_0 = A + B + C$$

$$M_1 = A + \bar{B} + C$$

$$M_2 = \bar{A} + B + C$$

$$M_3 = \bar{A} + B + \bar{C}$$

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

pos K-Map:		0	1
0	$A+B$	$\bar{A}+B$	
1	$A+\bar{B}$	$\bar{A}+\bar{B}$	

$$01) Y = (A+B+C)(A+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})$$

		00	01	11	10
0	0	1	1	1	
1	1	0	1	0	

$$02) Y = (A+B+C)(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$$

		00	01	11	10
0	0	1	0	1	
1	1	1	0	1	

Q) $Z = f(A, B, C, D)$

$Z = 1$ for the minterms (5, 2, 5, 7, 8, 13)

$Z = \text{don't care}$ for the minterms (10, 15)

$Z = 0$ for the remaining minterms.

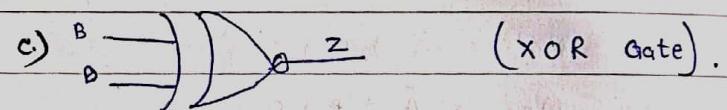
a. Draw the truth table.

b. Simplify Z using K-map method

c. Draw the circuit diagram.

		00	01	11	10	
		00	1	0	0	1
		01	0	1	1	0
		11	0	1	X	0
		10	1	0	0	X

b.) $Z = \bar{B}\bar{D} + BD$



K-Map locations.

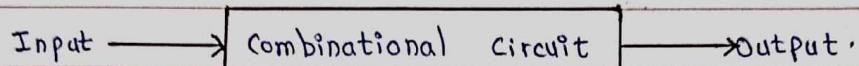
		00	01	11	10
0	0	2	6	4	
1	1	3	7	5	

		00	01	11	10
00	0	4	12	8	
01	1	5	13	9	
11	3	7	15	11	
10	2	6	14	10	

LESSON 05.

Sequential Logic Circuits.

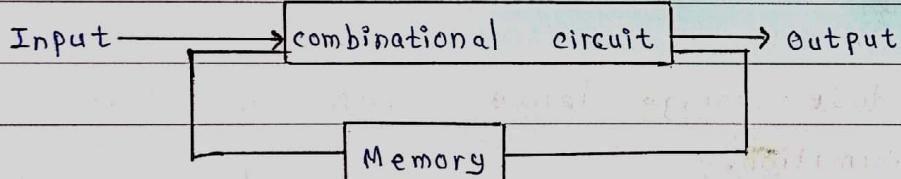
Combinational circuits.



- ⇒ A combinational circuit, is a circuit consists of several inputs and outputs with interconnection of gates.
- ⇒ For same inputs it gives same outputs always.
- ⇒ Every input has a single, unique output.

Sequential Logic circuits.

In here, the present output depends on the present input / past output



- ⇒ Sequential circuits are built out of memory elements and combinational logic devices.
- ⇒ Outputs are depending on both inputs and current contents of its memory .
- ⇒ Depending of the memory contents or data stored in the memory , these circuits produce different outputs for the same set of inputs.
- ⇒ Synchronous and Asynchronous are two type of Sequential Logic circuits.

Synchronous Sequential Logic Circuits.

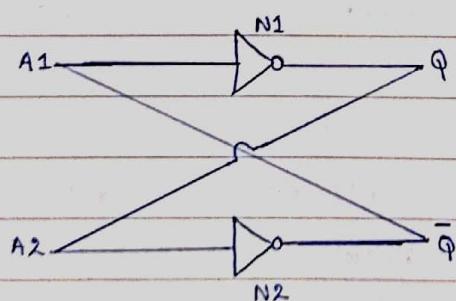
- ⇒ Various blocks of the logic circuit are triggered by a clock signal which distributed to all the memory elements in the circuit (Master clock).
- ⇒ Output of these blocks change at the same time.

Asynchronous Sequential Logic Circuits.

- ⇒ Various blocks of the logic circuit are triggered by various signals.
- ⇒ Outputs of these blocks can change at different instances.

Latches.

- ⇒ Latch is a data storage device which can store only one bit of information.
- ⇒ The basic circuit can implement by cross-coupling two NOT gates.
- ⇒ Symbol :



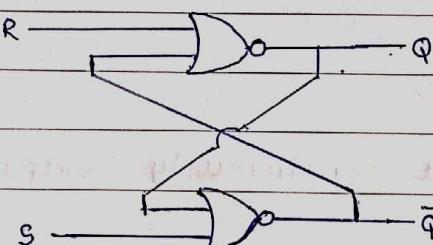
- ⇒ The output of each gate is connected to the input of the other.

SR Latch (Set-Reset Latch)

⇒ SR Latch is a circuit with:

- 2 cross-coupled NOR gate or 2 cross-coupled NAND Gate.
- 2 inputs (S for SET and R for RESET)
- 2 outputs (Q , \bar{Q})

⇒ Symbol:



if, $R=1$ Q always = 0

if, $S=1$ Q always = 1

case 1: if $S=0$ $R=1$, then $Q=0$ and $\bar{Q}=1$

if reset all inputs then

$S=0$ $R=0$ and Output gives as same $Q=0$ and $\bar{Q}=1$

because latches stores previous data in its memory.

case 2: $S=1$ $R=0$, then $Q=1$ and $\bar{Q}=0$

$S=0$ $R=0$, as memory.

case 3: $S=1$ $R=1$, $Q=0$, $\bar{Q}=0$ (Error/can't use)

Truth table :

S	R	Q	\bar{Q}
0	0	AS Memory.	
0	1	0	1
1	0	1	0
1	1	NOT Allowed	

Hold state. get the same out as previous out

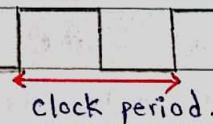
Reset state

Set state.

Flip-Flops.

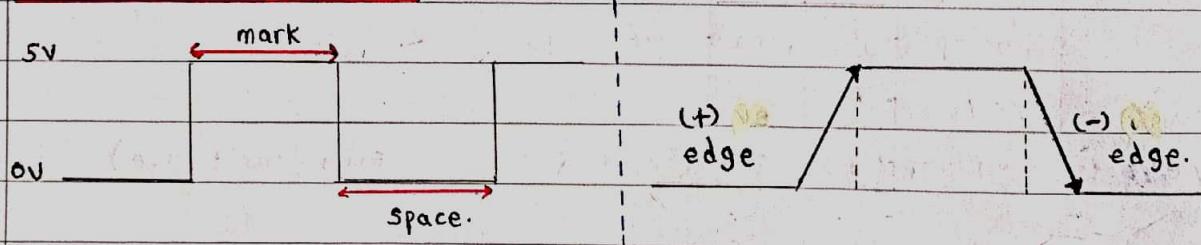
Clock

- A clock is a special device that continuously outputs 0's and 1's.
- The time it takes to change from one to zero is called a **clock period** or **clock cycle**.

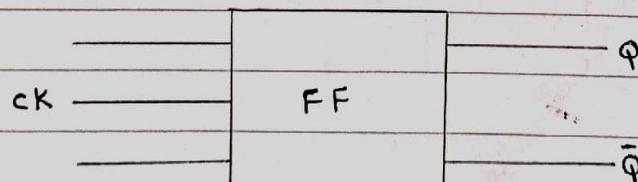


clock period.

Types of triggering

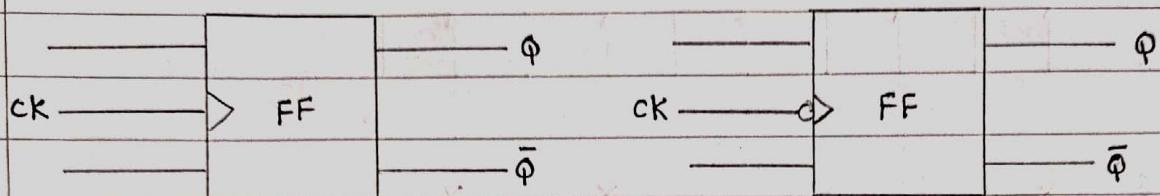


Level triggering



Activated by the mark.

Edge triggering.



Activated by the (+)ve edge

Activate by the (-)ve edge.

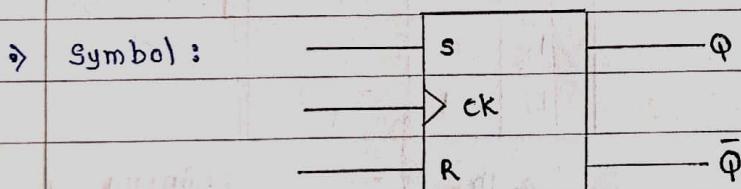
SR Flip-Flop

⇒ Truth table :

CK	S	R	Q _{n+1}
0	0	0	Q _n
0	0	1	Q _n
0	1	0	Q _n
0	1	1	Q _n
1	0	0	Q _n
1	0	1	0
1	1	0	Q ₁
1	0	1	NA

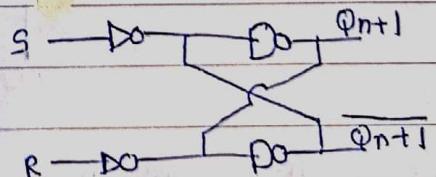
When CK = 0 ,

$$Q_{n+1} = Q_n.$$

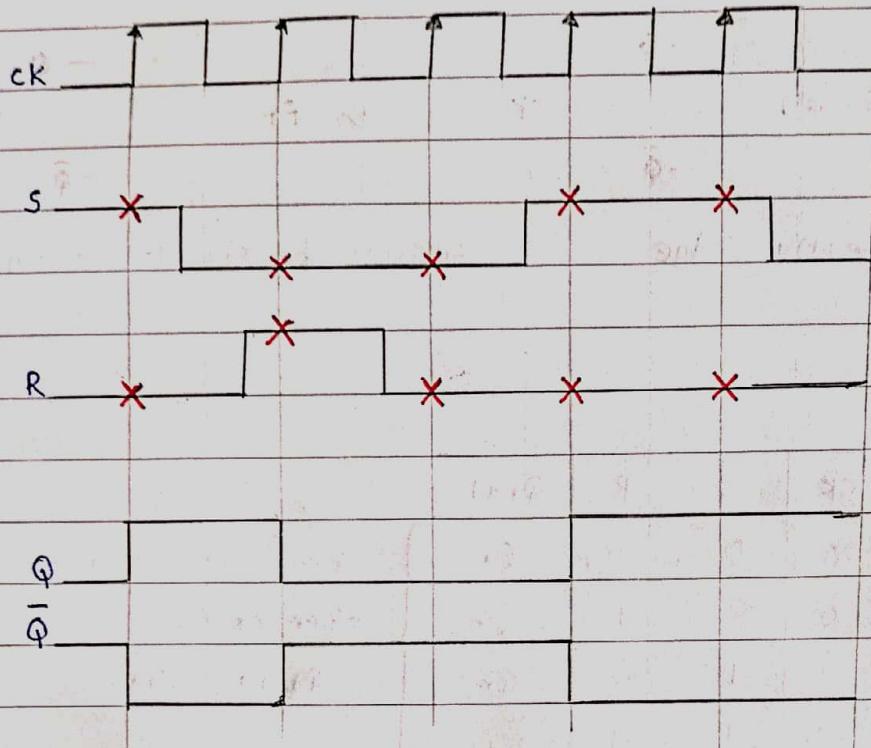


⇒ When CK = 1 :

S	R	Q _{n+1}	̄Q _{n+1}
0	0	As previous output / Memory.	
0	1	0	1
1	0	1	0
1	1	N/A	N/A



Q) Draw the output waveform for the (+) edge triggered SR FF



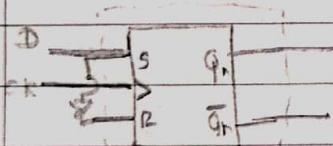
D Flip-Flop (Data FF)

⇒ Symbol :

D	Q
CK	\bar{Q}

CK	D	Q_{n+1}
0	0	Q_n
0	1	Q_n
1	0	0
1	1	1

} when ck is low
 $Q_{n+1} = Q_n$



⇒ The D F.F is containing SR F.F init

⇒ When $CK = 1$:

D	Q_n	Q_{n+1}
0	0	0
	1	0
1	0	1
	1	1

CK	D	Q_{n+1}
0	X	Q_n ← memory
1	0	0
1	1	1

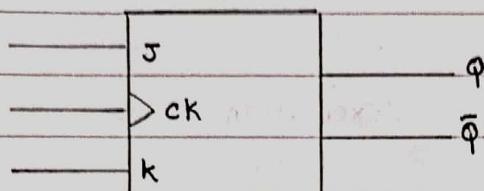
if we give D as input of S, R gets the compliment of S because D is the only input for this FF

$$D = 0 \rightarrow S = 0 \quad R = 1$$

$$D = 1 \rightarrow S = 1 \quad R = 0$$

JK Flip-Flop:

⇒ Symbol:



J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

when
 $CK=1$

when $CK=0$, $Q_{n+1} = Q_n$.

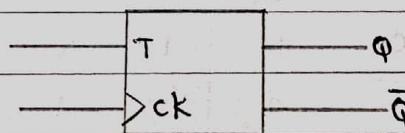
⇒ When $CK=1$:

J	K	Q_n	Q_{n+1}
0	0	0	0 (Q_n)
		1	1
0	1	0	0 (0)
		1	0
1	0	0	1 (1)
		1	1
1	1	0	1 (\bar{Q}_n)
		1	0 } compliment of the previous output: (\bar{Q}_n)

} output: (\bar{Q}_n)

Toggle Flip-Flop (T-FF)

⇒ Symbol:



T	Q_{n+1}
0	Q_n
1	\bar{Q}_n

when $CK=0$

$Q_{n+1} = Q_n$

⇒ When $CK=1$:

T	Q_n	Q_{n+1}
0	0	0
	1	1
1	0	1
	1	0

if $CK=0$ T is don't care
 memory and Q_{n+1} also is
 memory.

mem Q_{n+1}

Excitation Tables.

SR Flip Flop.

S	R	Q_n	Q_{n+1}
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	NA
		1	

Excitation Table :

Q_n	Q_{n+1}	S	R
0	0	0	X
	1	1	0
1	0	0	1
	1	1	X

JK Flip Flop.

J	K	Q_n	Q_{n+1}
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	1
		1	0

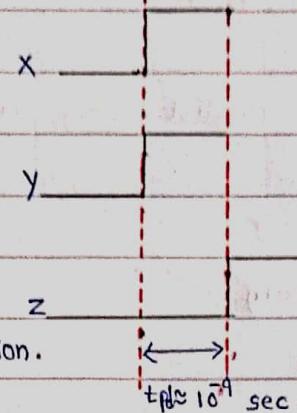
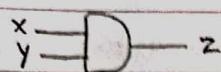
Excitation Table :

Q_n	Q_{n+1}	J	K
0	0	0	X
	1	1	X
1	0	1	X
	1	1	X

Excitation Tables.

Q_n	Q_{n+1}	SR	D	JK	T
0	0	0X	0	0X	0
0	1	10	1	1X	1
1	0	01	0	X1	1
1	1	X0	1	X0	0

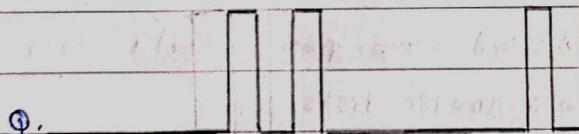
Propagation Delay.



- Propagation delay is the time duration taken for a signal to reach its destination.

Race around condition.

- Within one clock pulse the output will keep on toggling again and again, and it may become indeterminate; race around condition.
- If $J=K=1$, and $ck=1$ for a long time, the output will toggle as long as ck is high, it makes the output signal (Q) unstable.
- Race around condition occurs only in level-triggered flipflop.



Q.

What is race around condition? Explain with timing diagram.

Ans: In race around condition, the output of the flip flop toggles uncontrollably between high and low.

This happens when the clock signal (CK) is high for a long time and the inputs J and K are both high.

During each high pulse of the clock, the output toggles between high and low, creating a race-around condition.

The timing diagram shows the clock signal (CK) with multiple high pulses. The output Q toggles between high and low during each high pulse, illustrating the race-around condition.

Causes of race around condition:

1. Long clock pulse: If the clock signal (CK) is high for a long time, the output of the flip flop will keep on toggling again and again, creating a race-around condition.

2. High J and K inputs: If the inputs J and K are both high, the output of the flip flop will keep on toggling again and again, creating a race-around condition.

3. Low clock frequency: If the clock frequency is very low, the output of the flip flop will keep on toggling again and again, creating a race-around condition.

4. Poor circuit design: If the circuit design is poor, the output of the flip flop will keep on toggling again and again, creating a race-around condition.

Lesson 06

Computer Systems Organization

- » A digital computer consist of the following hardware components.
 - CPU
 - Main Memory
 - Auxiliary Memory
 - I/O devices.

Central processing unit

- » The brain of the computer. This directs the flow of data and basically controls all operations that performed on computers.
- » we can store data in a processor using transistors.
- » Its main function is to execute instructions stored in the main memory by fetching the instructions.

Memory

Memory

Main Memory

Auxiliary Memory (secondary)

» used to temporarily store data for processing

» used for permanently store data.
Magnetic-disks.

Instructions to process data.

» programs and data that are not immediately needed store in here.

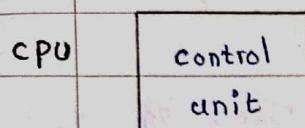
Information to be sent.

» RAM

» Volatile Memory.

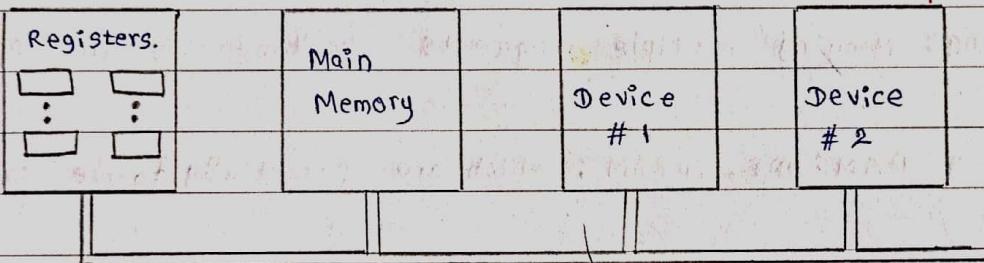
I/O Devices.

- » I/O Devices provides a intermediate between computer and outside world to communicating.



ALU

I/O Devices.



CPU Organization.

Von Neumann Architecture.

- » Basic of all computers, it assumes that every computation pulls data from memory, process it, and then sends it back to memory.

Von Neumann bottleneck.

- » This architecture is very important and is used in our PCs and as well as in super computers.
- » In this Architecture, programs and data are stored in memory and data moves between memory and the processor with latency.
- » "Bottleneck" is a limitation on throughput caused by the standard personal computer architecture.

Approaches to overcoming the von Neumann bottleneck:

- ⇒ Caching: The process of storing and accessing data from a cache; a high speed data storage layer which stores a subset of data.
- ⇒ Prefetching: A technique for speeding up fetch operations by beginning a fetch operation whose result is expected to be needed soon.
- ⇒ Multithreading: Managing multiple requests simultaneously in separate threads.
- ⇒ New types of RAM: DDR, SDRAM; which are potentially double output.
- ⇒ Rambus: A memory subsystem consisting of the RAM, the RAM controller and the bus(path) connecting RAM to the microprocessor and devices in the computer that use it.

Essentials of the von Neumann Architecture.

- ⇒ One instruction at a time.
- ⇒ Both data and input are in same memory.
- ⇒ The CPU is composed of several parts:
 1. Arithmetic Logical Unit.
 2. Control unit
 3. Registers.