Digital Electronics

Analog	Digital
Analog signals	Binary Number system
Sinosoidal signal	ASCII. value.

Continuous signal

Base (or) Radix of the number system.

0, 1, 2, ..., 9 → digits of the decimal system

0, 1 → digits of the binary system

$(345)_{10}$ ⇒ decimal system

$(10111)_2$ ⇒ Binary system

ALU = Arithmetic Logic Units

Logic Gates :-

A device which will perform certain logical operations such as (multiplication, Addition, Complement....).

(i) AND Gate
(ii) OR Gate
(iii) NOT Gate
(iv) XOR (or) Ex-OR Gate (Exclusive OR Gate)
(v) X NOR (or) Ex-NOR Gate (Exclusive NOR Gate)

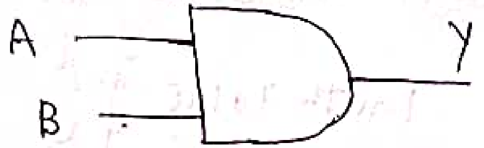
Basic Gates.

(vi) NAND Gate
(vii) NOR Gate

Universal Gates

→ Symbol
→ Boolean function (or) Output function
→ Truth Table

AND Gate:-



Two inputs AND Gate

The Boolean function of AND gate is Y = A.B.

$= 2^{\text{no of i/p}}$

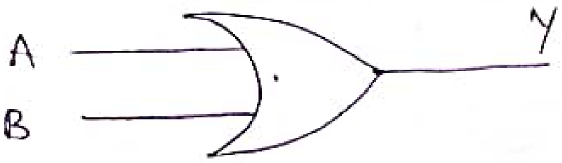
$= 2^2 = 4$ combination.

Truth Table

I/P		O/P
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

A	B	C	Dec
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

OR Gate:-



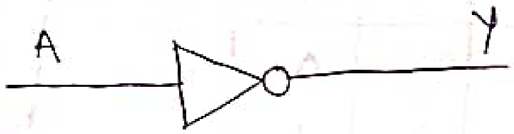
The Boolean function of OR Gate is $Y = A+B$

Truth Table

I/p		o/p
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate:-

(Inverter).

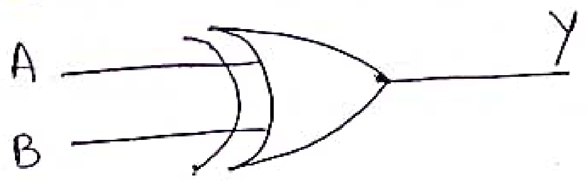


Boolean function of NOT Gate is $Y = \overline{A}$ (or) A'

Complement operation.

I/p	o/p
A	Y
0	1
1	0

XOR Gate:-



Boolean function is $Y = A \oplus B.$ (or) $A\overline{B} + \overline{A}B.$

Truth Table

I/p		O/p
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$0 \times 1 + 1 \times 0 = 0$

$0 \times 0 + 1 \times 1 = 1$

$1 \times 1 + 0 \times 0 = 1$

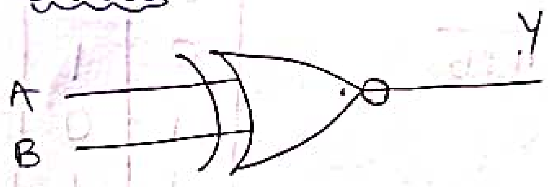
$1 \times 0 + 0 \times 1 = 0.$

same i/ps వుంటే = 0

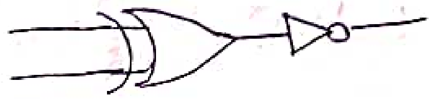
different i/ps వుంటే = 1.

$A \oplus B \oplus C = AB\overline{C} + A\overline{B}C + \overline{A}BC$

XNOR Gate:-



(or)



Boolean function is $Y = A \odot B$ (or) $\overline{A \oplus B}$ (or) $AB + \overline{A}\,\overline{B}$

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

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

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

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

$Y = 0 + \overline{A}\overline{B} + AB + 0$

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

Truth Table

I/p		O/p
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

$0 \cdot 0 + 1 \cdot 1 = 1$

$0 \cdot 1 + 1 \cdot 0 = 0$

$1 \cdot 0 + 0 \cdot 1 = 0$

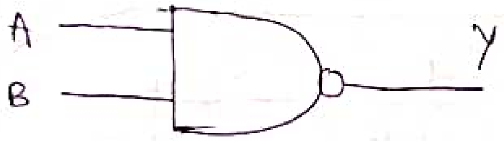
$1 \cdot 1 + 0 \cdot 0 = 1$

bar (‾) వుంటే (‾) add చేయాలి ↓ $A\overline{A} = 0$

* వుంటే +

+ వుంటే * చేయాలి.

NAND Gate:-

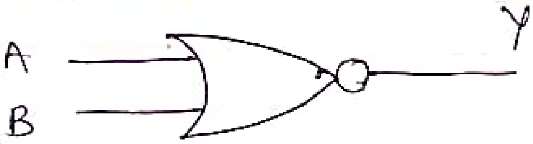


Truth Table

I/p		o/p
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Boolean Function $Y = \overline{A \cdot B}$

NOR Gate:-



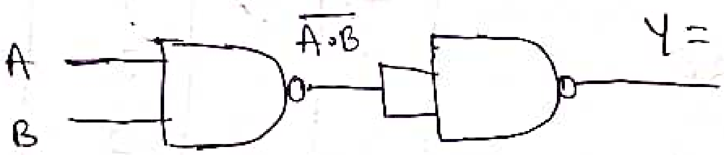
Truth Table

I/p		o/p
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Boolean function is $Y = \overline{A + B}$

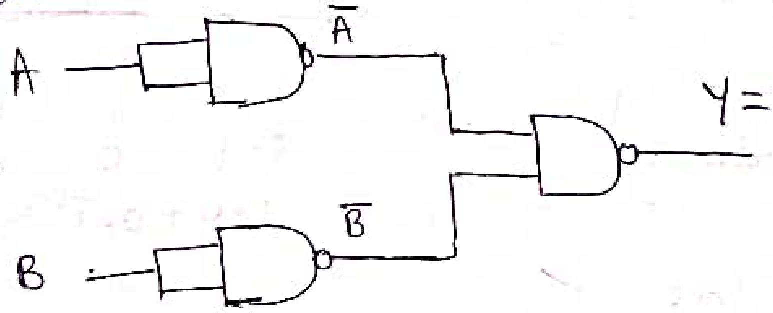
Implementation of Basic - Gates with NAND Gate:-

(i) AND Gate with NAND Gate:-



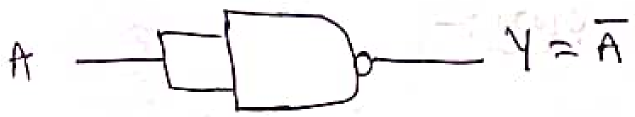
$Y = \overline{\overline{A \cdot B}} = A \cdot B$

(ii) OR Gate with NAND Gate:-

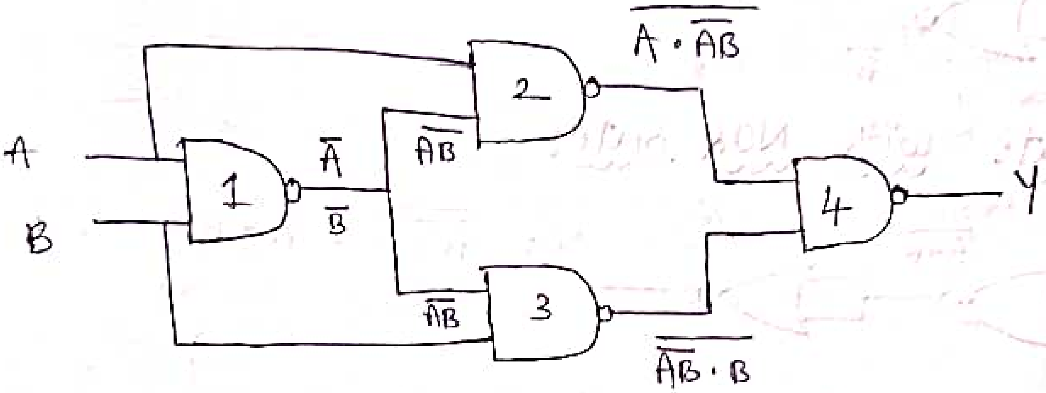


$Y = \overline{(\overline{A} \cdot \overline{B})} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$

(iii) NOT Gate with NAND Gate:-



(iv) XOR Gate with NAND Gate:-


$\overline{A\cdot\overline{AB}}$
2
A
B
1
$\overline{A}$
$\overline{B}$
$\overline{AB}$
$\overline{AB}$
3
4
Y
$\overline{\overline{AB}\cdot B}$

$Y = \overline{(\overline{A\cdot\overline{AB}})\cdot(\overline{\overline{AB}\cdot B})}$ Demorgans Theorem

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

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

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

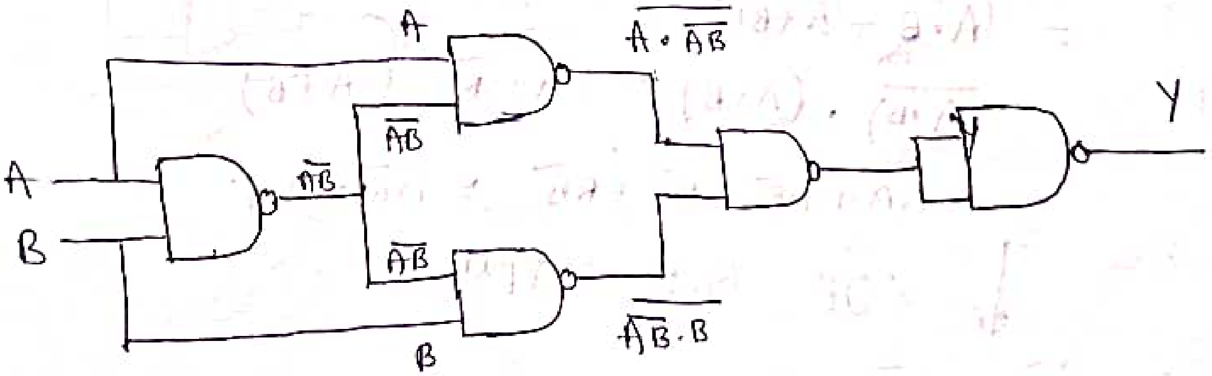
$Y = A\overline{A} + A\cdot\overline{B} + B\overline{A} + B\overline{B}$

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

$A\cdot\overline{A} = 0$

$B\overline{B} = 0.$

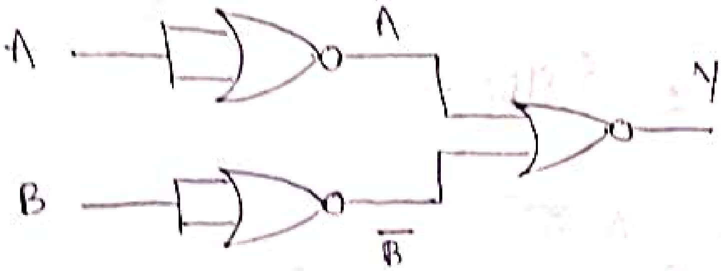
(v) XNOR Gate with NAND Gate:-


A
$\overline{A\cdot\overline{AB}}$
Y
$\overline{AB}$
A
B
$\overline{AB}$
$\overline{AB}$
$\overline{\overline{AB}\cdot B}$
B

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

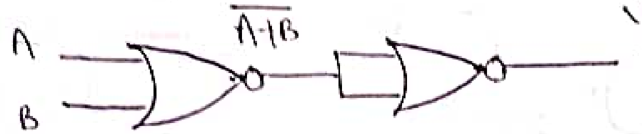
Implementation of Basic Gates with NOR Gate:-

(i) AND Gate with NOR Gate:-



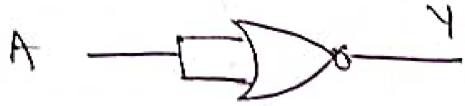
$Y = \overline{(\bar{A} + \bar{B})} = \bar{\bar{A}} \cdot \bar{\bar{B}} = A \cdot B$

(ii) OR Gate with NOR Gate:-



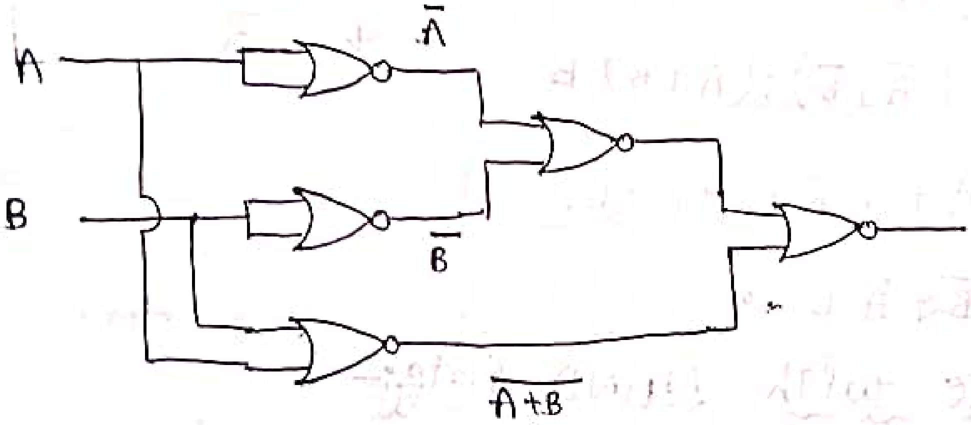
$Y = \overline{\overline{A+B}} = A+B$

(iii) NOT Gate with NOR Gate:-



$Y = \bar{A}$

(iv) XOR Gate with NOR Gate:-



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

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

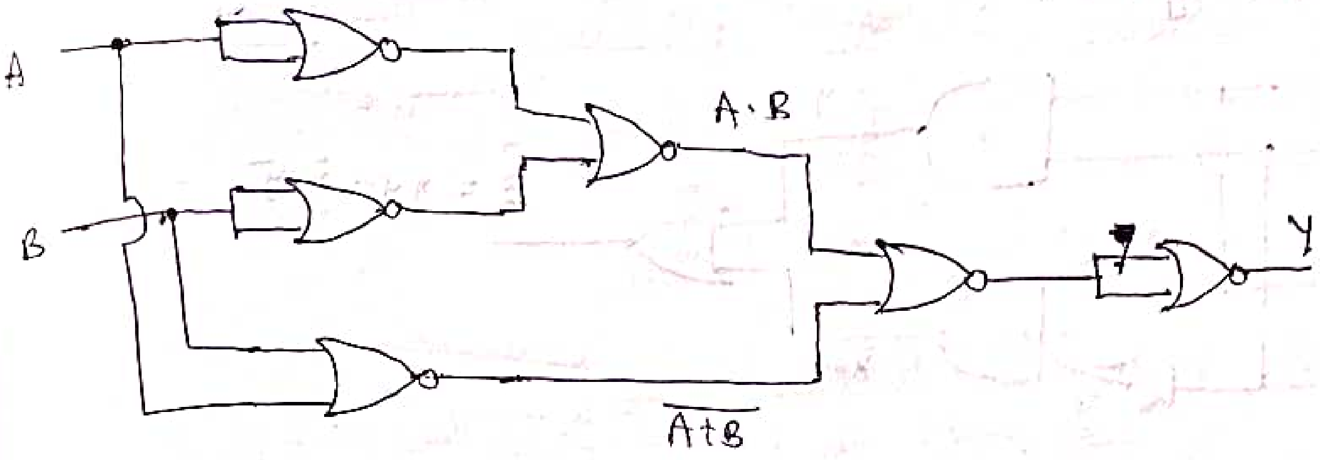
$= \overline{(A \cdot B)} \cdot (A+B) = (\bar{A} + \bar{B})(A+B)$

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

↓ XOR Gate output.

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

(v). XNOR Gate with NOR Gate:-



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

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

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

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

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

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

④. Implement, the boolean function $S = x\overline{y}+\overline{x}y$ with basic gates.

Sol:- $S = x\overline{y}+\overline{x}y$.

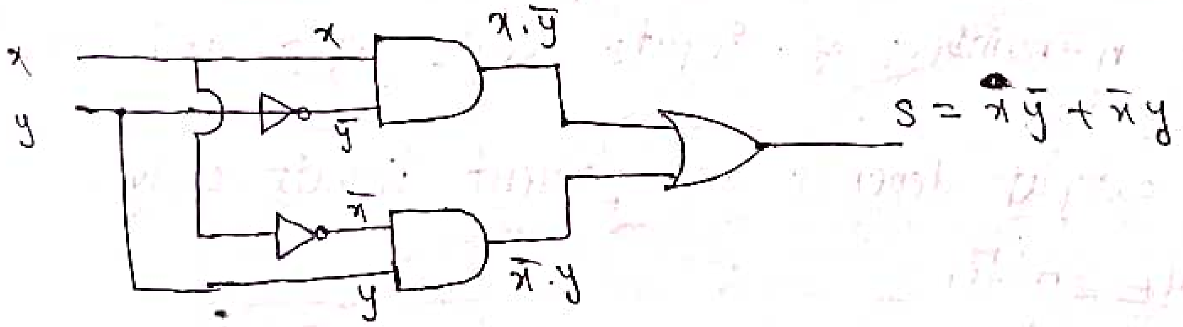


fig:- XOR Gate with Basic Gates

(Q). Implement the Boolean function $z = xy + \bar{x}\bar{y}$ with basic gates.

Sol:-

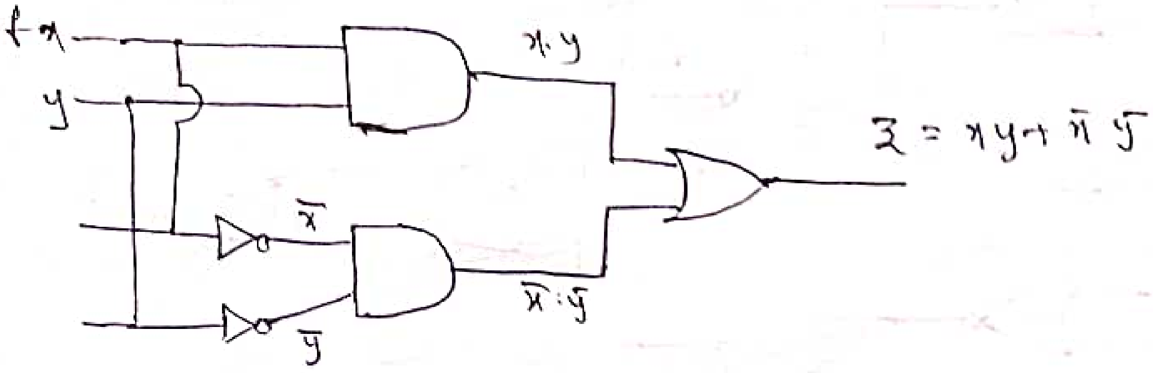
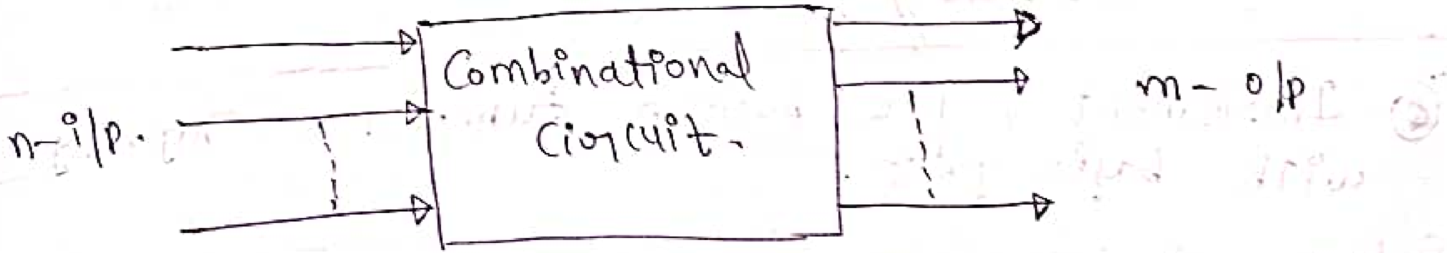


fig:- X NOR Gate with Basic Gates.

$SOP = A\bar{B} + \bar{A}B\bar{C} + A\bar{B}C + ABC$ = (Sum of product form)

$POS = (A+B+C)(A+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$ = (Product of Sum form).

Combinational Circuits:-



(i) Combinational circuit is a logic circuit which consist n-number of inputs logic gates, and m-numbers of outputs.

(ii) The output depends on present inputs only.

⇒ Half Adder

⇒ Full Adder

⇒ BCD Adder.

Binary Addition:- MSB ⇓ Most Significant Bit ⇙ LSB = Least Significant Bit

```
    4 2 5
    6 2 5
      1 → carry bit
Carry bit ← 1 0 5 0 → Sum
```

$$\begin{array}{r} 1 \\ 0 \\ \hline 01 \end{array}$$

carry ↙ 0, 1 → sum

$$\begin{array}{r} 1 \\ 1 \\ \hline 10 \end{array}$$

Carry ↙ 1, 0 ↘ sum

(1+1+1)

$$\begin{array}{r} 1 \\ 1 \\ \hline 10 \\ 1 \\ \hline 11 \end{array}$$

carry ← 1, 1 → sum

0 + 0 = 00

0 + 1 = 01

1 + 0 = 01

1 + 1 = 10

Carry ↙ 1, 0 ↘ Sum

$$\begin{array}{rrrrrl} & 1 & 1 & 1 & 0 & \to 14 \\ (+) & 1 & 0 & 0 & 1 & \to 9 \\ \hline 1 & 0 & 1 & 1 & 1 & \to 23 \\ \hline \end{array}$$

$$\begin{array}{rrrrrrrl} & 1 & 0 & 1 & 0 & 1 & 1 & \to 43 \\ (+) & 1 & 1 & 1 & 0 & 1 & 1 & \to 59 \\ \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & \to 102 \\ \hline \end{array}$$

→ Truth Table

→ Boolean function ~~co~~ o/p func^n.

→ Implementation of o/p function with logic gates.

Half Adder:-

Half adder is a combinational circuit, which will add two binary bits.

(i) Truth Table

Dec	I/p		o/p	
	x	y	Carry	Sum
0	0	0	0	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	0

x & y are called i/p ~~are~~ variables.

(ii) Boolean function (or) o/p equation

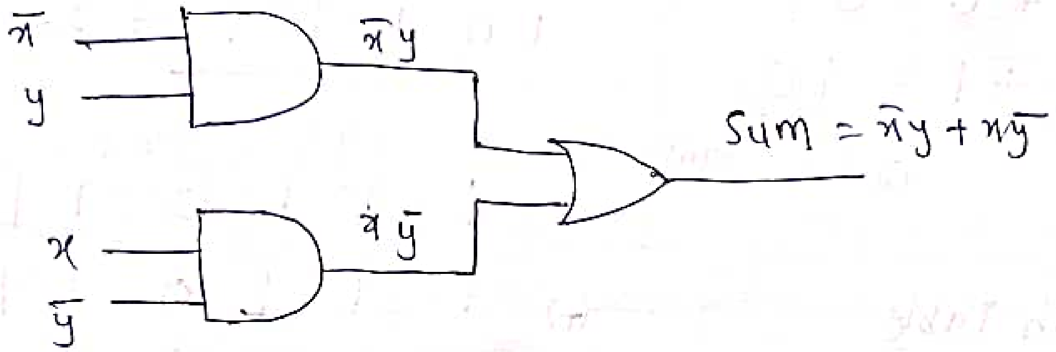
Sum = $\bar{x}y + x\bar{y}$ ↓ K-map

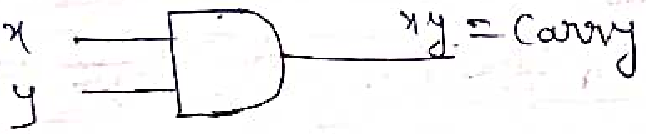
Carry = xy ↓ using Karnaugh-map

(iii) Implementation of Boolean function with logic gates.

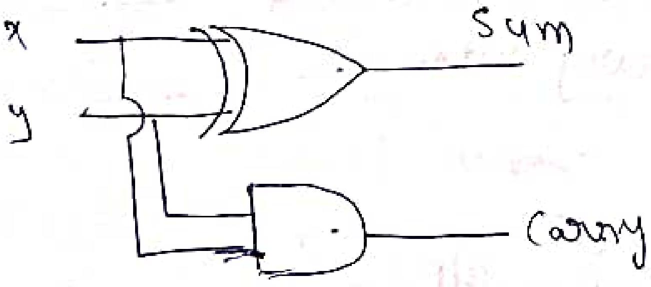
logic circuit :-

I/p available in both complements & true form.





(or)



Full Adder :-

It is a combinational circuit which will add Three binary digits (or) bits.

No of inputs = 3 (x, y, z).

No of outputs = 2 (sum, carry).

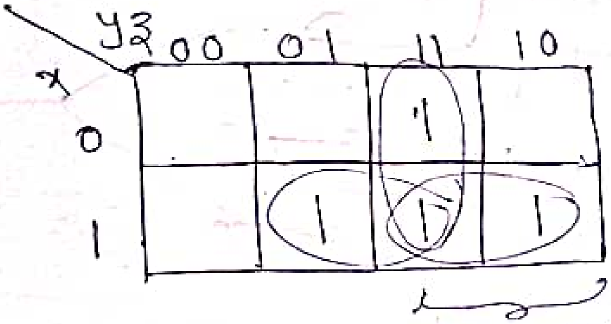
(i) Truth Table

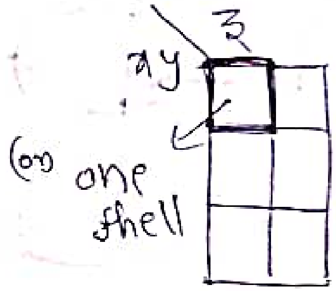
Dec	Input			Output	
	x	y	z	Carry	sum
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Sum = $\Sigma(1,2,4,7)$

Carry = $\Sigma(3,5,6,7)$

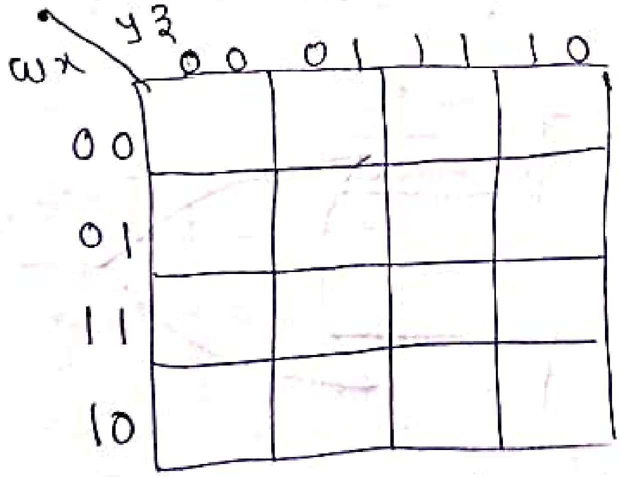
(ii) Boolean function.





pair, Quad, Octet.

$2^3 = 8 = 8$ sheels



two cells are said to be ajacent if they are deffered by single bit (or) position.

x	y	z
1	0	1
1	1	1
x		z

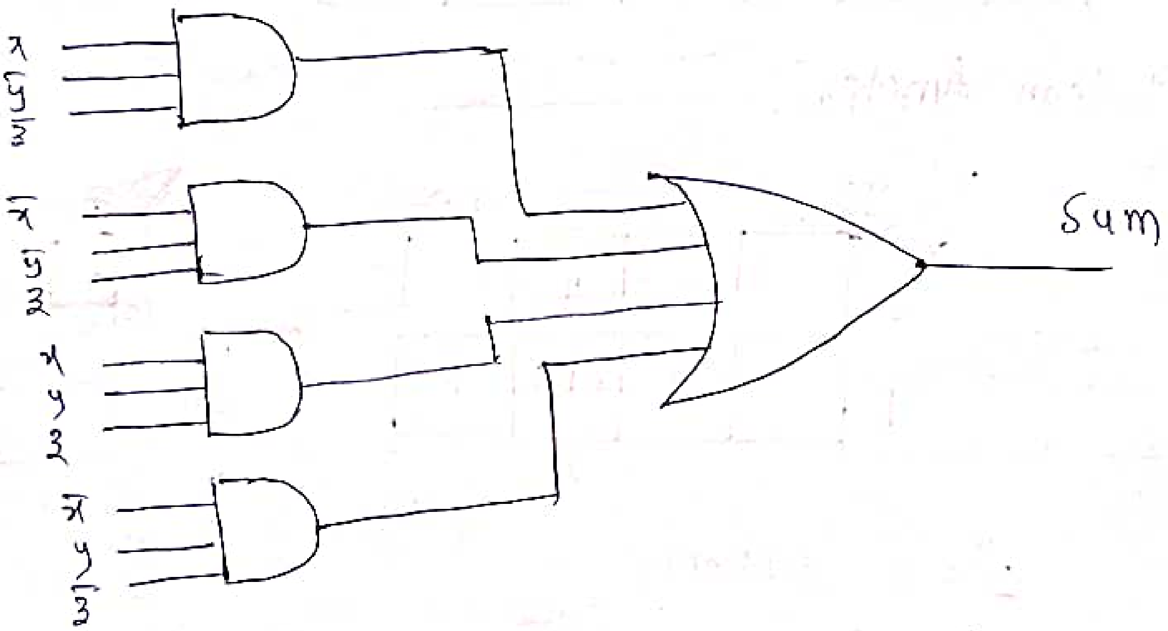
x	y	z
1	1	1
1	1	0
x	y	

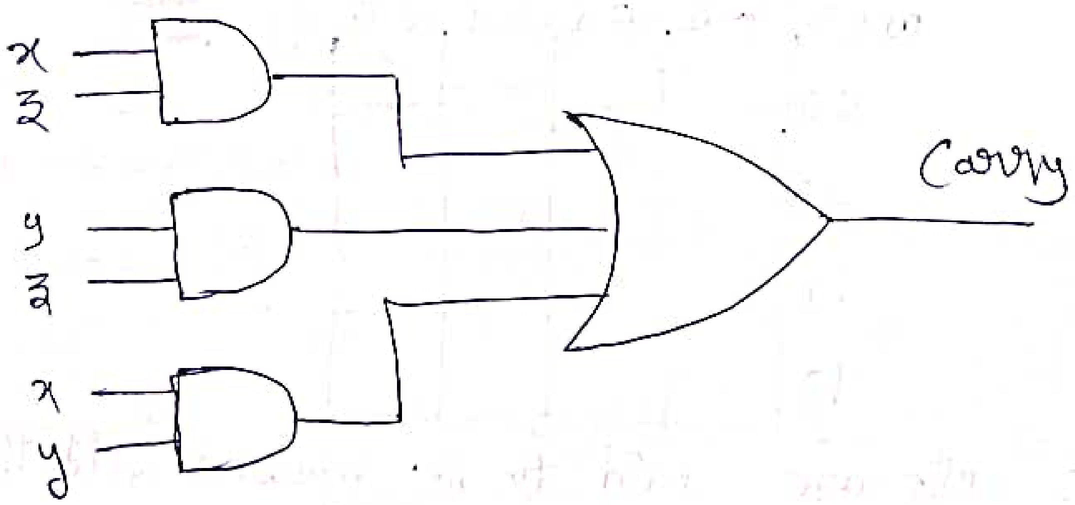
x	y	z
1	1	1
0	1	1
	y	z

Carry = $xz + xy + yz$.

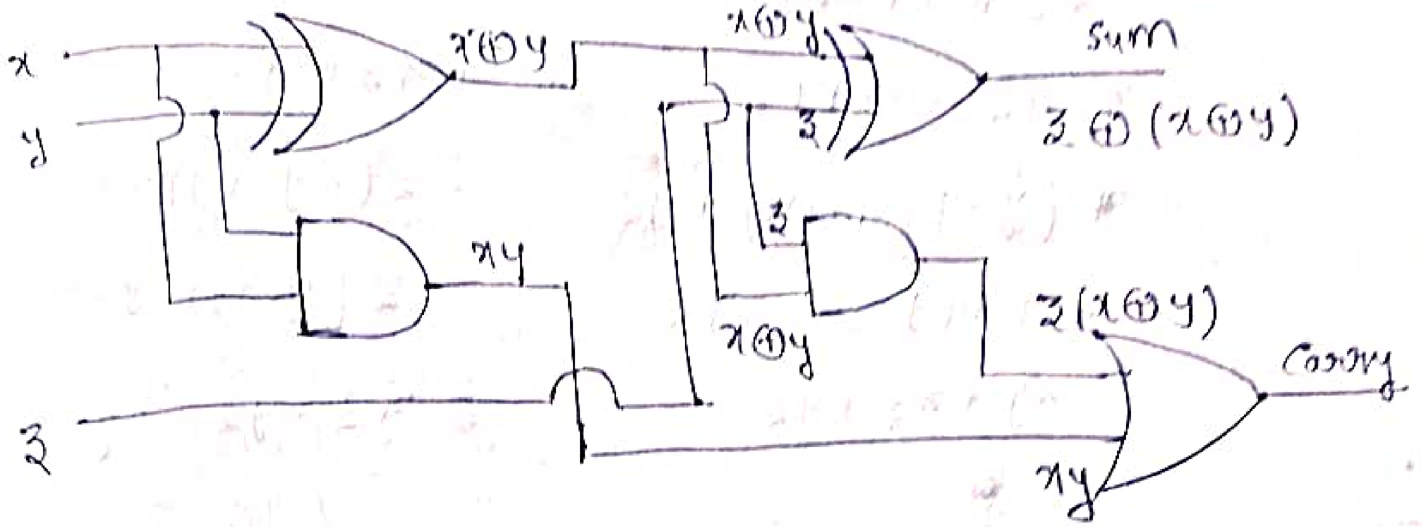
x \ yz	00	01	11	10
0		1		1
1	1		1	

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





Full adder with two half adders-



$\Rightarrow A\bar{z} + \bar{A}z$

$= (x \oplus y)\bar{z} + \overline{(x \oplus y)}\, z$

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

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

$= x\bar{y}\bar{z} + \bar{x}y\bar{z} + (xy + \bar{x}\bar{y})z.$

Sum $= x\bar{y}\bar{z} + \bar{x}y\cdot\bar{z} + xyz + \bar{x}\bar{y}z.$

Carry $= z(x \oplus y) + xy$

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

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

according to distribution theorem.

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

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

$\downarrow$ $x + \bar{x} = 1, \quad x\bar{x} = 0$

$= x\bar{y}z + y(1(x+z))$

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

$= x(\bar{y}z + y) + yz$ (or) $x\bar{y}z + yz + xy$

$= x((\bar{y}+y)\cdot(z+y)) + yz$ $= z(x\bar{y} + y) + xy$

$= x(1(y+z)) + yz$ $= z((x+y)\cdot(y+\bar{y})) + xy$

$= xy + xz + yz$ // $= xz + yz + xy$ //

(11) Simplify $f(w,x,y,z) = \Sigma(0,1,2,3,4,5,6,7,8,9,10,12)$.

Sol^n

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

BCD Adder :-

BCD – Binary Coded Decimal

Binary code for Decimal digits.

i.e, 0,1,2,3, - - - →, 9 is known as BCD code.

15 → 1111 35 → 100011.

15 → 1111 ⇐ Binary Code

15 → 00010101 ⇐ BCD Code

② → 0010 → Augend

③ → 0011 → Addend

Table for BCD Adder :-

Dec	Binary Sum					Dec	BCD Sum				
	K	Z_8	Z_4	Z_2	Z_1		C	S_8	S_4	S_2	S_1
0.	0	0	0	0	0	0.	0	0	0	0	0
1.	0	0	0	0	1	1.	0	0	0	0	1
2.	0	0	0	1	0	2.	0	0	0	1	0
3.	0	0	0	1	1	3.	0	0	0	1	1
4.	0	0	1	0	0	4.	0	0	1	0	0
5.	0	0	1	0	1	5.	0	0	1	0	1
6.	0	0	1	1	0	6.	0	0	1	1	0
7.	0	0	1	1	1	7.	0	0	1	1	1
8.	0	1	0	0	0	8.	0	1	0	0	0
9.	0	1	0	0	1	9.	0	1	0	0	1
10.	0	1	0	1	0	10.	1	0	0	0	0
11.	0	1	0	1	1	11.	1	0	0	0	1
12.	0	1	1	0	0	12.	1	0	0	1	0
13.	0	1	1	0	1	13.	1	0	0	1	1
14.	0	1	1	1	0	14.	1	0	1	0	0
15.	0	1	1	1	1	15.	1	0	1	0	1
16.	1	0	0	0	0	16.	1	0	1	1	0
17.	1	0	0	0	1	17.	1	0	1	1	1
18.	1	0	0	1	0	18.	1	1	0	0	0
19.	1	0	0	1	1	19.	1	1	0	0	1

(i) Binary sum & BCD sum are the same up to 01001.

(ii) If there is a carry in BCD sum then binary sum and BCD sum are not equal.

(iii) In order to differentiate first 10 combinations with remaining combinations, there must be correction factor.

⇒ To differentiate last 4 combinations with first 10 combinations K must be 1. (K=1)

⇒ Ily, To differentiate the combinations 01010 to 01111 from first 10 combinations Z_8 must be 1 and along with Z_8 either Z_4 or Z_2 must be 1.

(iv) From that correction we can have the following equation.

$$C = K + Z_8 Z_4 + Z_8 Z_2$$

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

1. $f(w,x,y,z) = \Sigma(0,1,2,3,4,5,7,8,10,12)$.

Sol:

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

pair
Quad
Octet.

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

$\bar{y}\bar{z}$

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

$\bar{w}z$

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

$\bar{x}\bar{z}$

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

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

$f(w,x,y,z) = \bar{y}\bar{z} + \bar{w}z + \bar{x}\bar{z}$ //

2. $f(w,x,y,z) = \Sigma(0,3,4,6,7,8,9) + \Sigma_d(5,10,14,15)$.

Sol: Σ_d అంటే d-don't care ని X గుర్తుతో indicate చేస్తాము.

wx \ yz	00	01	11	10
00	1	●	1	
01	1	X	1	1
11			X	X
10	1	1		X

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

$\bar{w}\bar{y}\bar{z}$

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

$w\bar{x}\bar{y}$

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

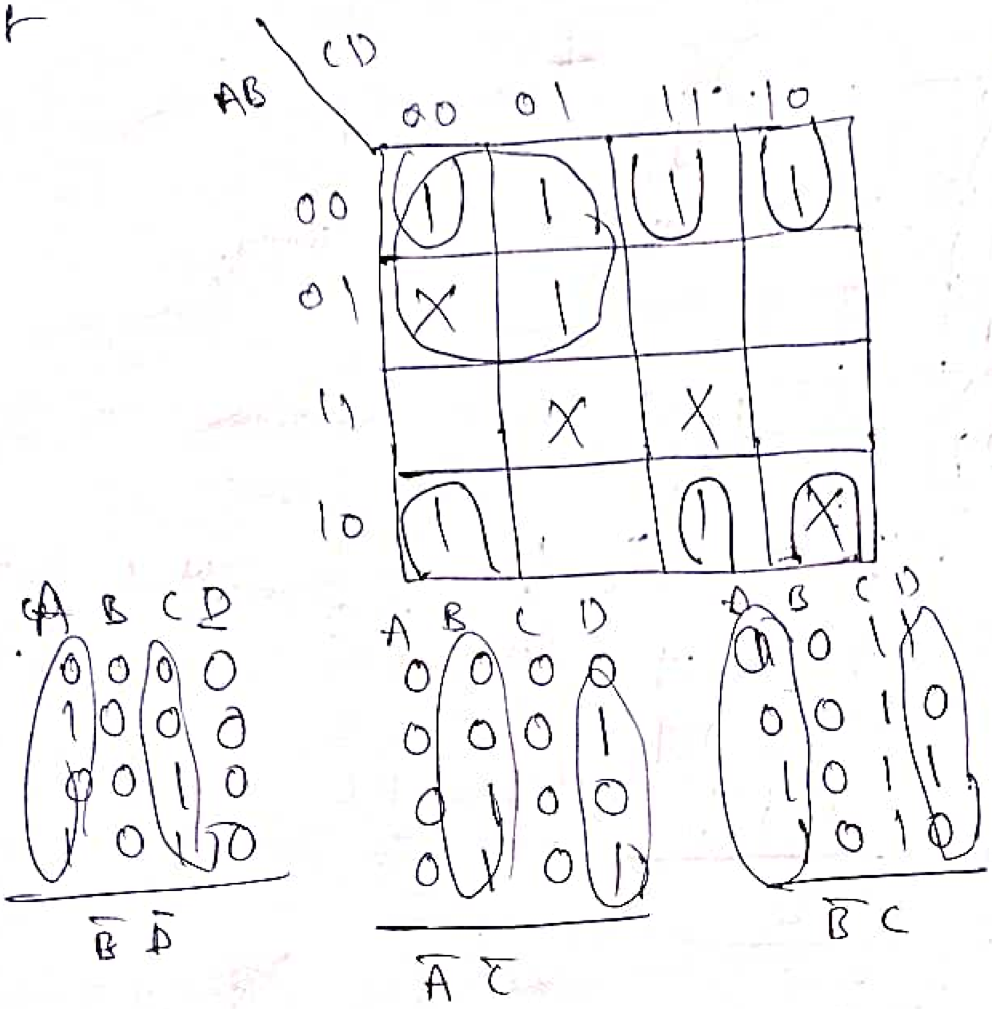
$\bar{w}yz$

0	1	0	0
0	1	0	1
0	1	1	1
0	1	1	0

$\therefore f(w,x,y,z) = \bar{w}\bar{y}\bar{z} + w\bar{x}\bar{y} + \bar{w}x + \overset{\bar{w}\,x}{\bar{w}yz}$

③ · $f(A,B,C,D) = \Sigma(0,1,2,3,5,8,11) + \Sigma_d(4,10,13,15)$

Sol:



AB \ CD	00	01	11	10
00	1	1	1	1
01	X	1		
11		X	X	
10	1		1	X

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

$\bar{B}\bar{D}$

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

$\bar{A}\bar{C}$

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

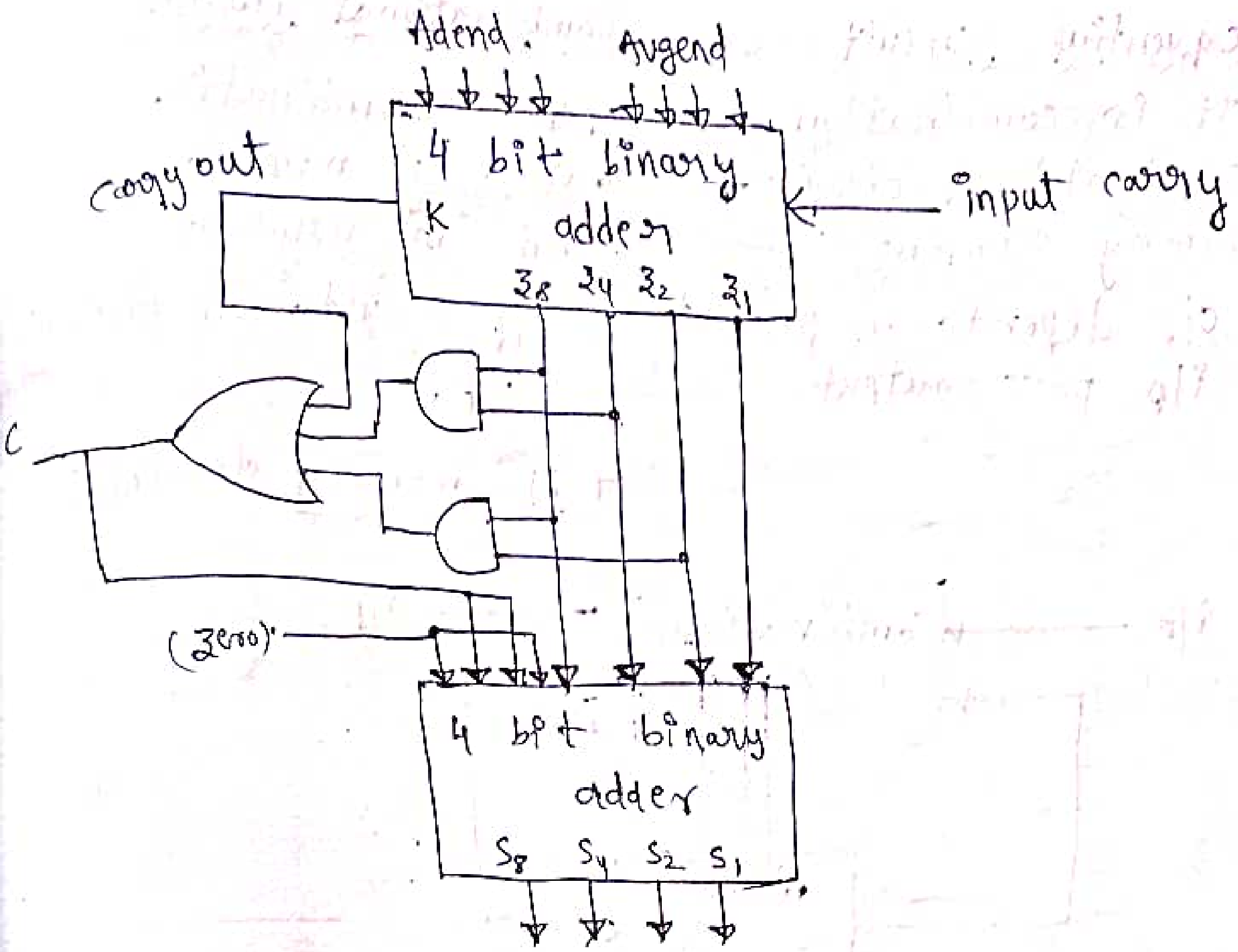
$\bar{B}C$

$\therefore f(A,B,C,D) = \bar{B}\bar{D} + \bar{A}\bar{C} + \bar{B}C$ //

Pair → one variable will be eliminated.

Quad → Two variables will be eliminated.

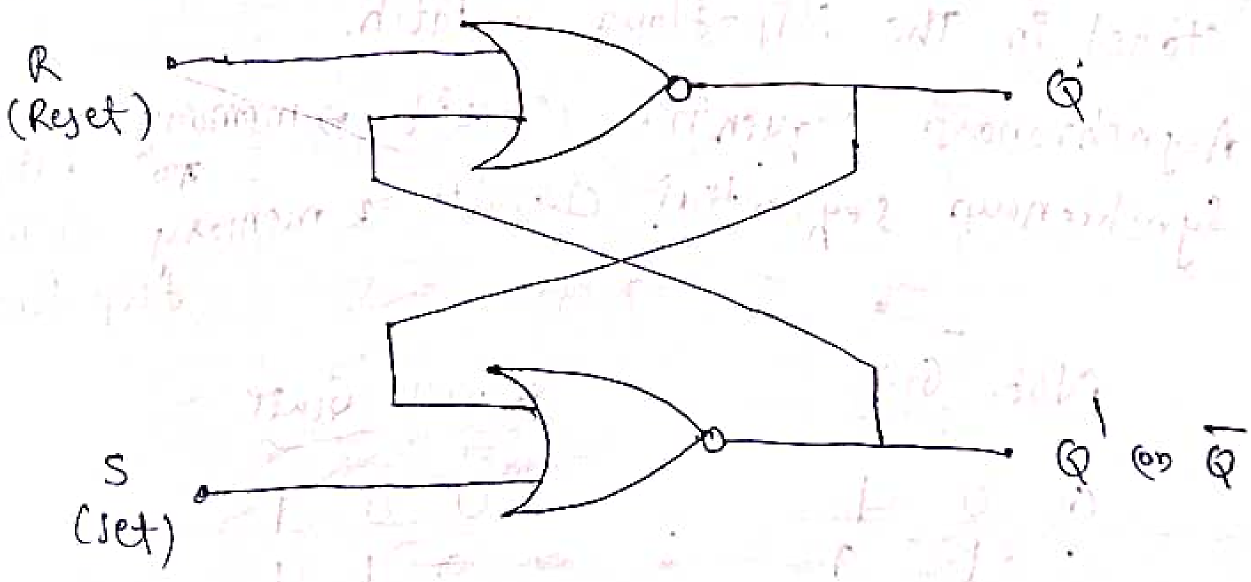
Octet → Three variablen will be eliminated.



$$C = K + Z_8 \cdot Z_4 + Z_8 \cdot Z_2$$

Latches & Flip-Flops:-

Latch:- It is 1 bit memory element.



Sequential circuit.

⇒ It is combination of combinational circuit + memory elements

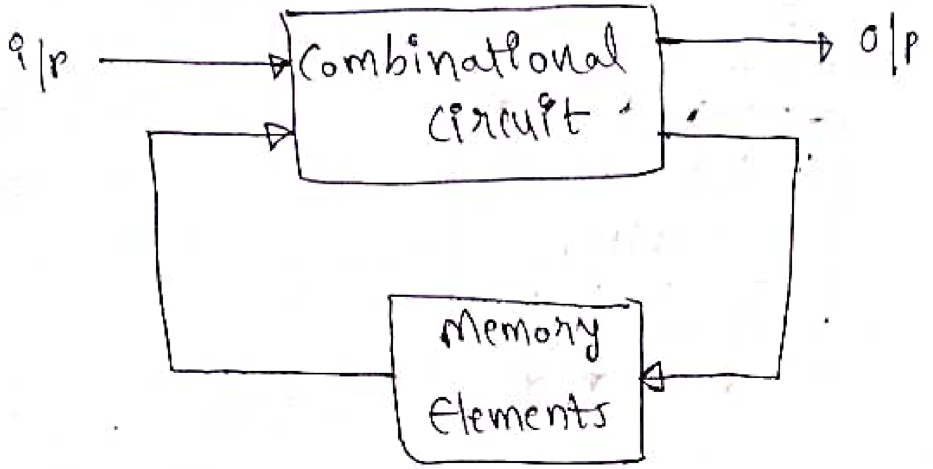
⇒. o/p depends on present i/p past output

Combinational circuit.

⇒ It is combination of logic gates 'n' number of i/ps and 'm' number of o/ps.

⇒ o/p depends on present i/p.

⇒ No memory element.



⇒ Memory elements store 1 bit of information,

⇒ present o/p depends on present i/p as well as past o/p

State (or) Status:- At any given time the information stored in the flip-flops (or) latch.

Asynchronous sequential circuit ⇒ memory elements ⇒ latch

Synchronous sequential circuit ⇒ memory elements ⇒ flip flops

NOR Gate

0	0	1
0	1	0
1	0	0
1	1	0

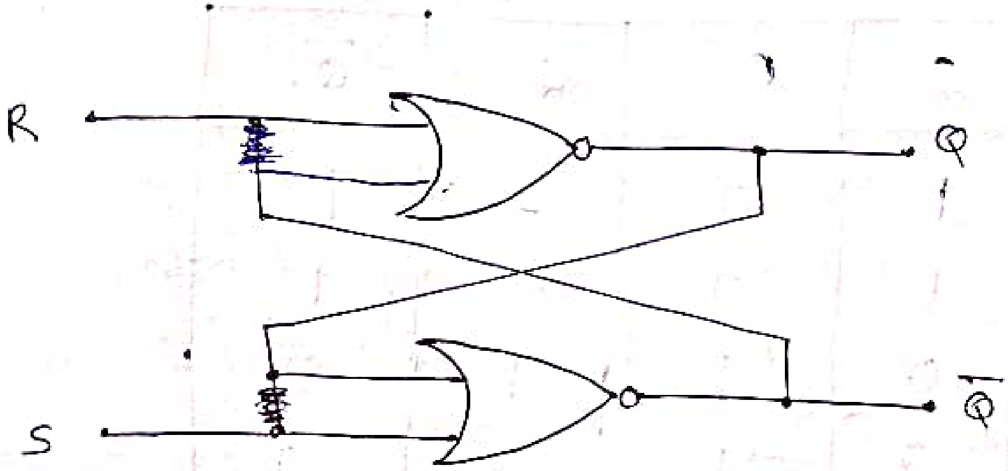
NAND Gate

0	0	1
0	1	1
1	0	1
1	1	0

⇒ For NOR Gate if any one i/p is 1 then o/p will be zero irrespective of other i/p.

⇒ For NAND Gate if any one i/p is zero then o/p will be 1 irrespective of other i/p.

Latch ⇒ Cross coupled NOR Gate (or NAND Gate.



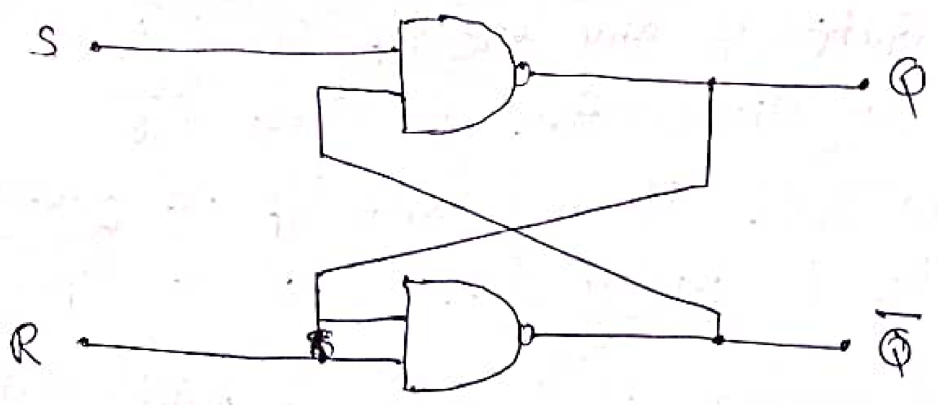
SR Latch with NOR Gates.

⇒ +ve logic

S	R	Q	Q̄
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	Indeterminate (or) Forbidden	

⇒ in such a way that ⇒ remember that

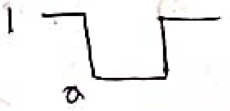
⇒ depends on the situation ⇒ that also you need to consider.



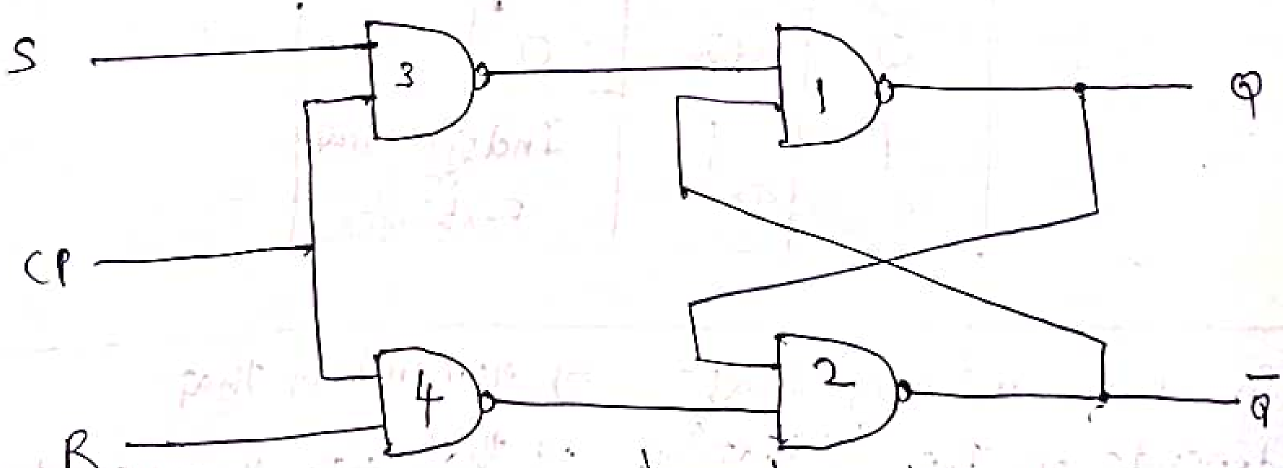
SR Latch with NAND Gates.

S	R	Q	Q̄
1	0	0	1
1	0	0	1
0	1	1	0
0	1	1	0
1	1	Indeterminate (or) Forbidden	

⇒ -ve logic



SR Flip Flop:- CP is always 1.



CP = Clock pulse ⇒

Q(t)	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Indeterminate

Characteristic table (or) Truth Table.

⇒ Q(t) represents present o/p

⇒ Q(t+1) represents next o/p.

(i) when clock pulse (CP) is 1 then the latch produces o/p.

(ii) When CP is '0' then the latch will not produce any o/p.

Q(t) \ SR	00	01	11	10
0			1	1
1	1		X	1

Q(t)	S	R
1	1	1
1	1	0
0	1	1
0	1	0
	S	

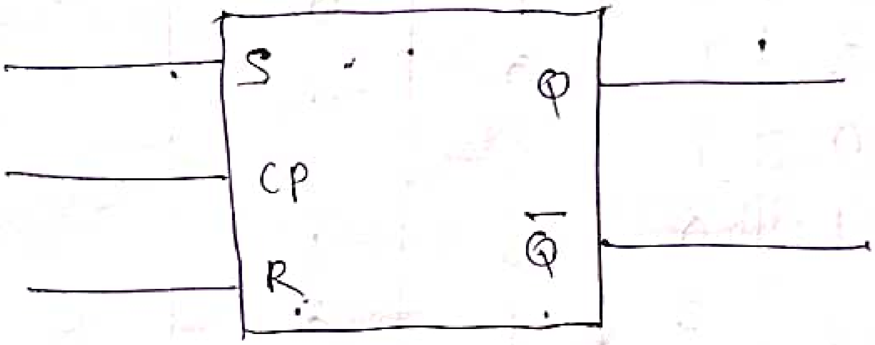
1	0	0
1	1	0
Q(t)		$\bar{R}$

$\therefore$ $Q(t+1) = S + Q(t)\bar{R}$

It is characterstic equation.

⇒ For Flip-Flop there is clock signal.

⇒ For latch there is no clock signal.



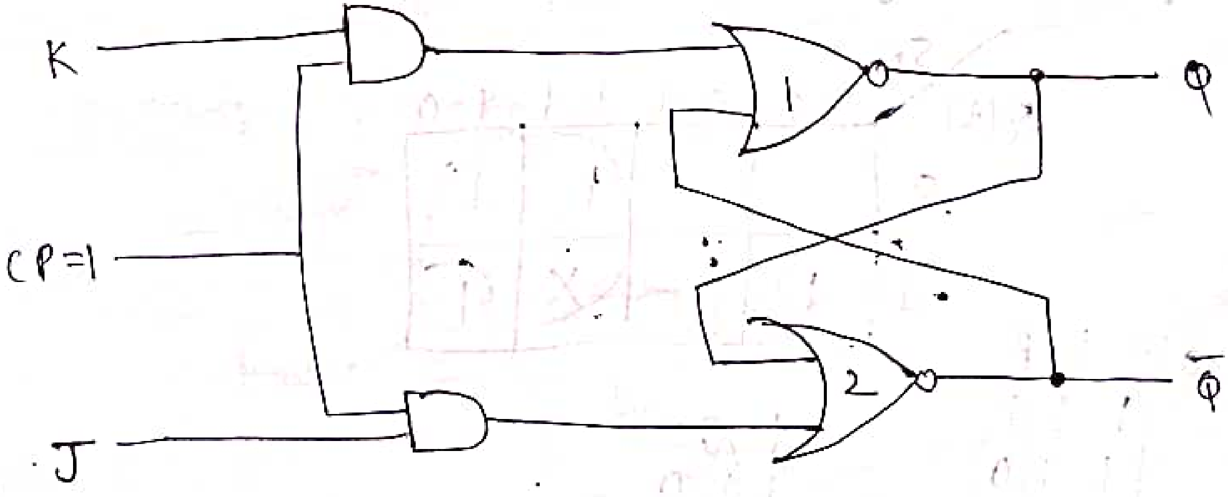
Symbol for SR flip-flop.

Draw back of SR Flip-Flop:-

⇒ when both i/ps are 1 (i.e, S=1, R=1) then the o/p will be indeterminate state (undefine state), (or forbidden state.

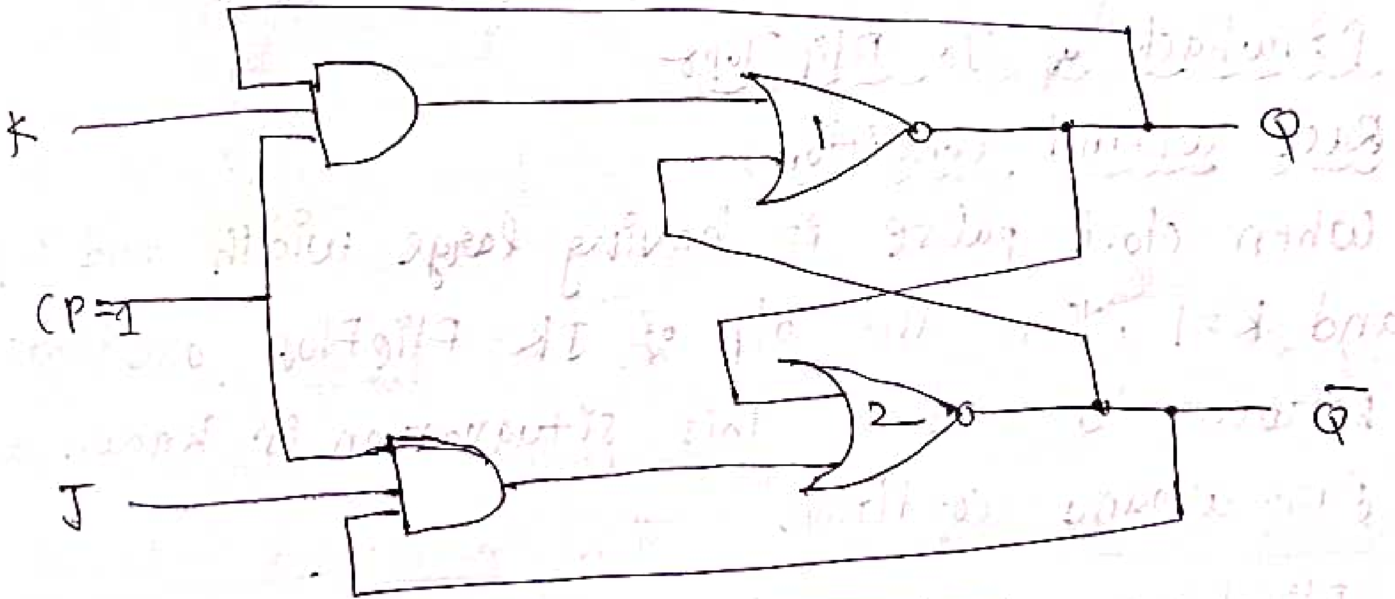
⇒ To avoid the drawback of SR Flip Flop, JK Flip Flop is used.

JK-Flip Flop:-



clock pulse (CP) is always one (1).

Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Q(t) \ JK	00	01	11	10
0			1	1
1	1			1

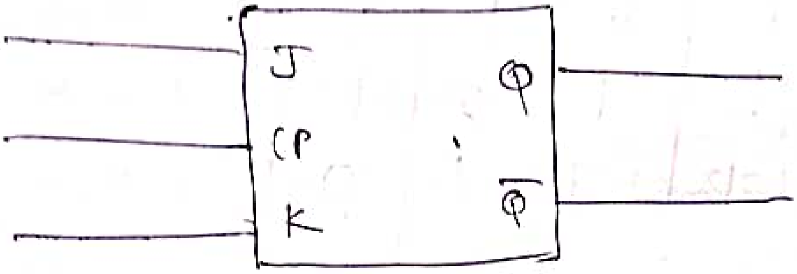
0 1 1
0 1 0

$\overline{Q(t)}$ J

1 0 0
1 1 0

Q(t) $\overline{K}$

$$\boxed{Q(t+1) = \overline{Q(t)}\,J + Q(t)\,\overline{K}}$$ ← Characterstic equation by using K-map.



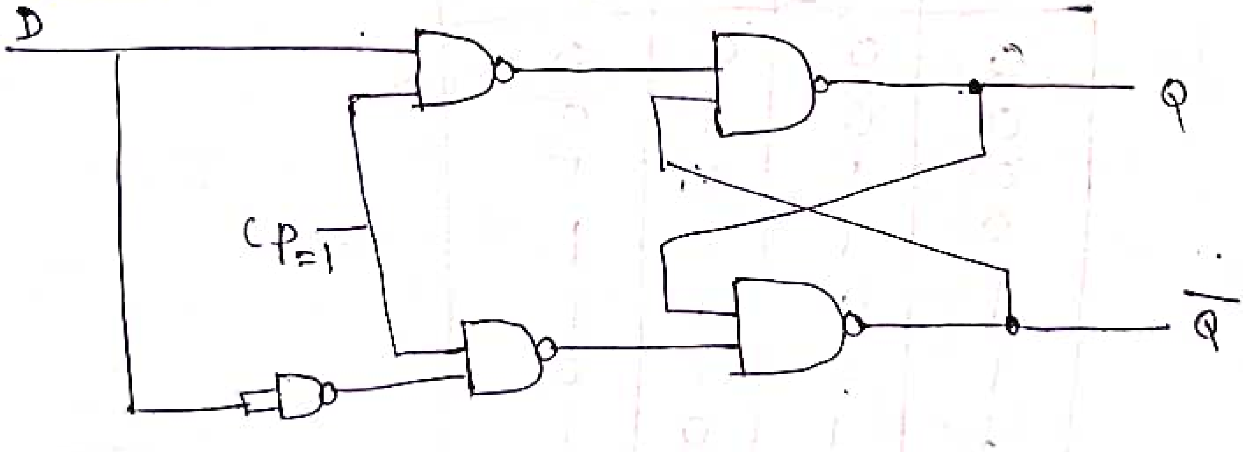
Symbol for JK Flip Flop

Drawback of JK Flip Flop:-

Race around condition:-

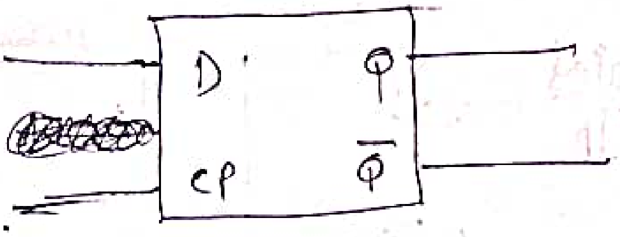
When clock pulse is having large width and j=1 and K=1, then the o/p of JK Flip Flop oscillate between '0' and '1' this situavation is known as "Race around condition."

D-Flip-Flop:-



Q(t)	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

same o/p.



Symbol for D Flip Flop.

D = Data Flip Flop.

Q(t) \ D	0	1
0		1
1		1

0 1
1 1

D//

∴ $\boxed{Q(t+1) = D}$ ← characterstic equation

Shift Registors and Countors:-

Registors:-

(i) It is a group of flip flops each store 1 bit of binary information.

(ii) n-bit register will have n-number of flip flops.

Shift registor:-

It is a registor capable of shifting binary information held in each cell ~~to~~ its neighbouring cells in a selected direction.

Counter:- (i) Counter is a registor that goes through pre-determined sequence of binary states.

(ii) n-bit counter will have n-number of flipflops and counts '0' through 2^n-1.

Shift Register:-

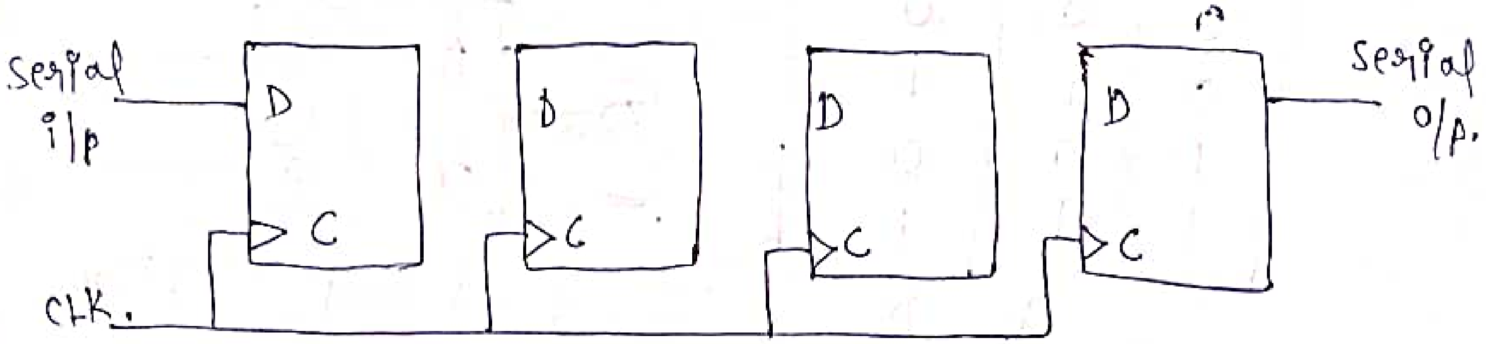


fig: 4 bit shift register.
shift right operation.

Counters:-

There are two types of counters
(i) Ripple counter (ii) Sychronous counter.

A3	A2	A1	A0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

MSB

LSB = Least Significant bit

Most Significant Bit

Logic-1:-

(i) A_0 is complements for every clock pulse.

(ii) A_1 is complements when A_0 changes from '1' to '0'.

(iii) A_2 is complements when A_1 changes from 1 to 0.

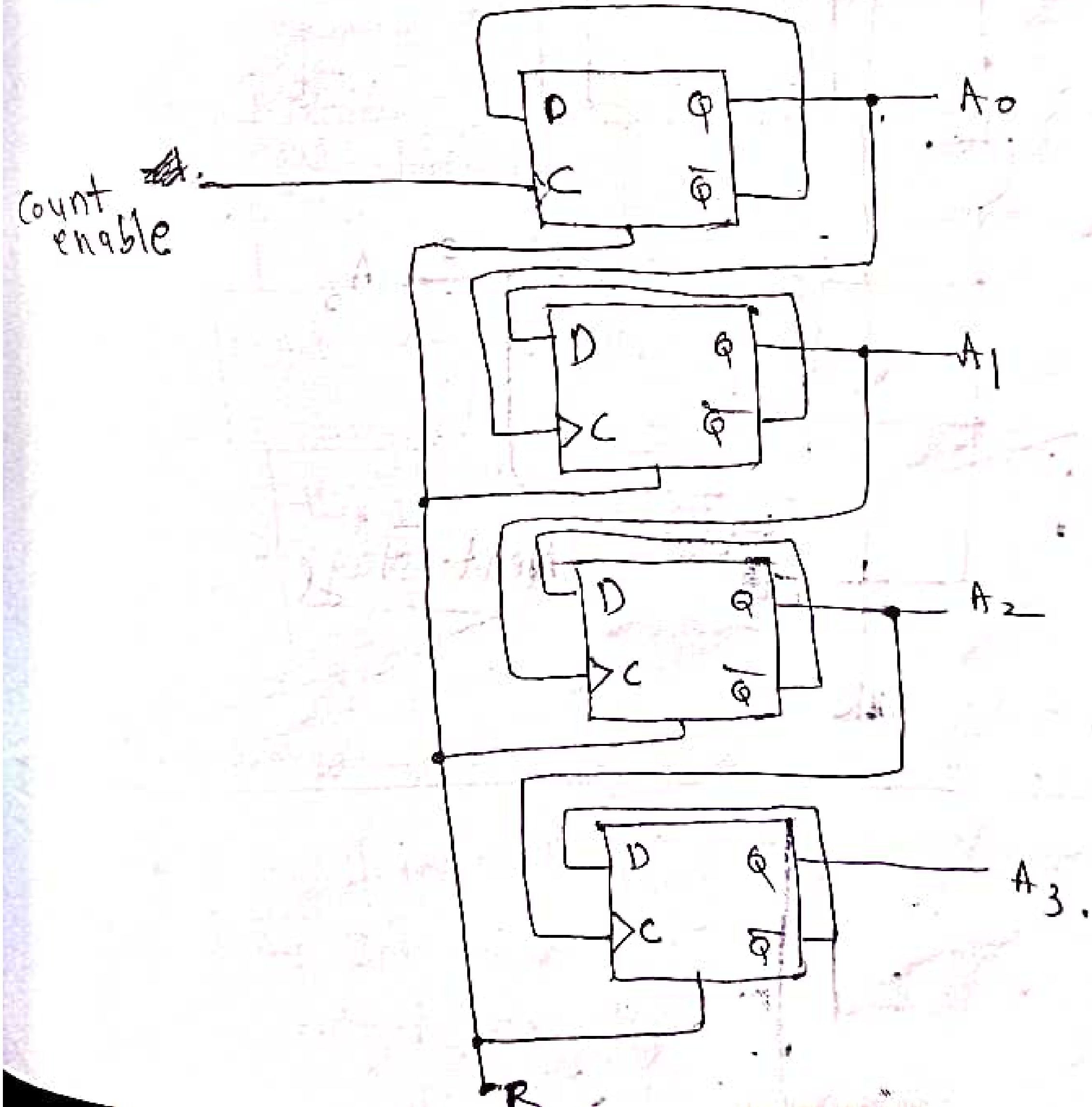
(iv) A_3 is complements when A_2 changes from 1 to 0.

Logic-2:-

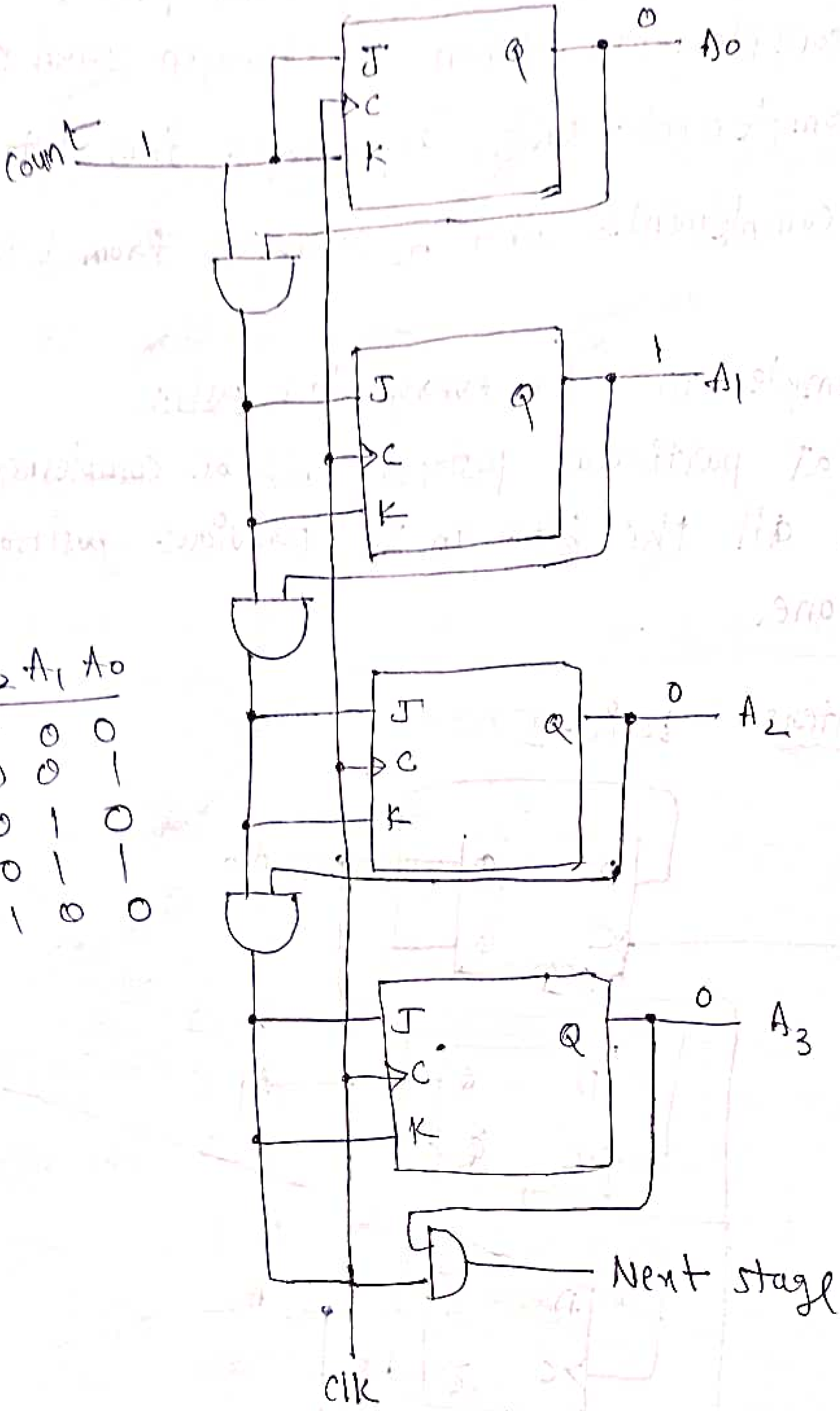
(i) A_0 is complements for every clock pulse.

(ii) A bit at particular position will be complemented if at all all the bits in the previous positions must be one.

Ripple Counter:- Logic-1



Binary Synchronous Counters:-



A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0

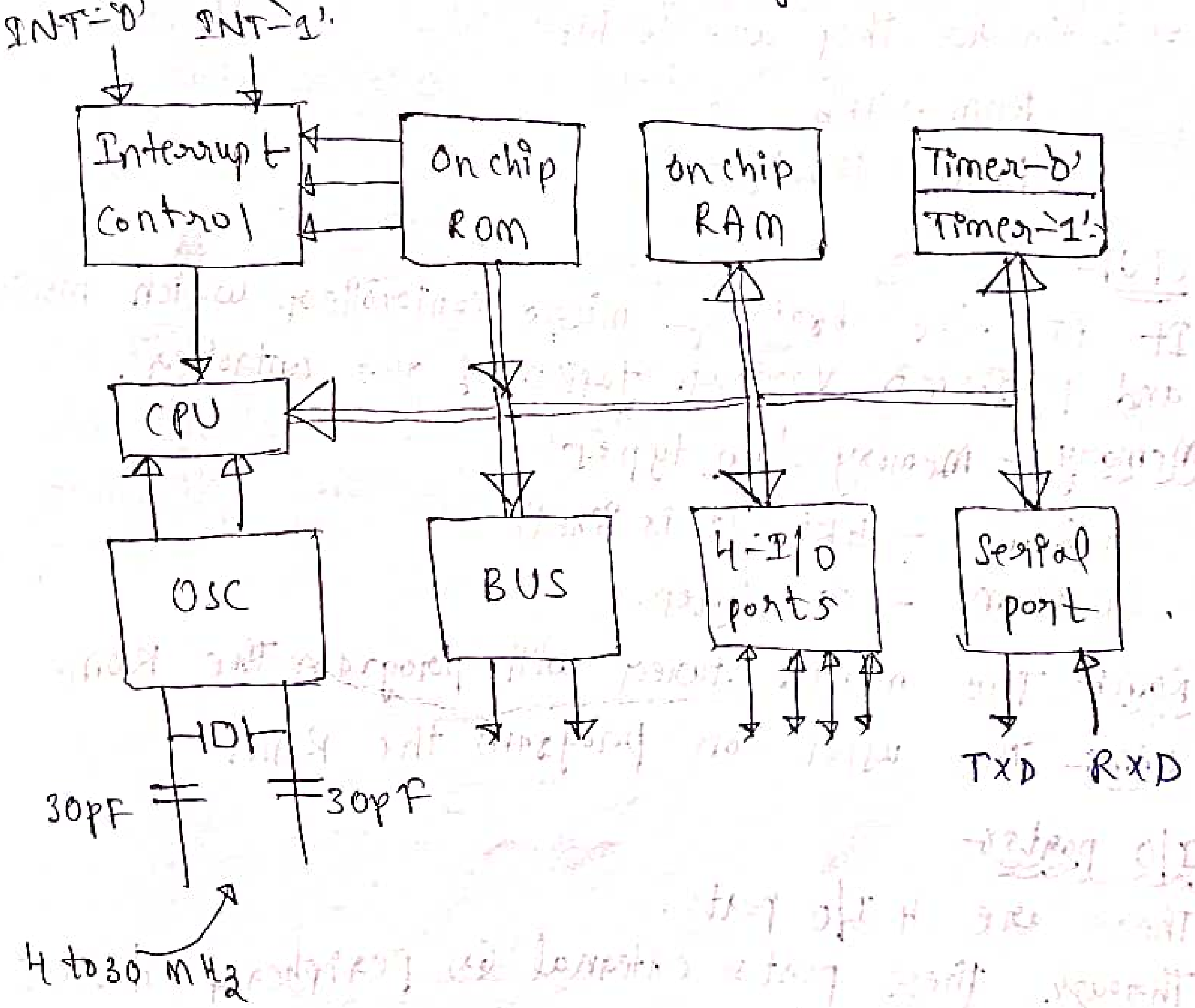
Micro Controller:-

Embaded system uses combination of Hardware & Software.

(i) It is small computer which consists CPU, Memory, I/O ports ..etc.

(ii) Applications → Home appliances
→ industrial
→ Agriculture
→ Defence.

→ 8051 Micro Controller block diagram



INT'0' → Interrupt '0'.

INT'1' → Interrupt '1'.

CPU → Central Processing Unit

TXD → Transmit Data

RXD → Recieve Data

OSC → oscillator.

I/O → Input /output

RAM → Random Access Memory.

ROM → Read Only Memory. ⇐ constant

⇒ 8051 micro controller was developed by INTEL in 1981.

⇒ It is a 8-bit micro controller.

⇒ 12 MHz clock signal.

⇒ 2 timers they are 16-bit

ROM → 4kb

RAM → 128 bytes.

CPU:-

It is like brain of micro controller which monitors and performs various tasks of the controller.

Memory:- Memory two types

(i) ROM - 4kb (It is fixed).

(ii) RAM - 128 bytes.

Rom:- The manufacturer will program the Rom.

RAM:- The user can program the RAM

I/O ports:-

There are 4 I/O ports.

Through these ports external (or) peripheral devices will be connected.

↕ ⇒ Bidirectional   ↓ unidirectional.

Serial port:-

In order to transmit (or) receive data serial port will be used.

Timers:-

It is having two timers each has 16 bit.

To find out pulse width (or) interval a timer will be used.

BUS:-

Group (or) Collection of wires.

It acts as channel (or) medium b/w different blocks

→ Address bus - 16 bit (16 wires)

→ Data bus - 8 bit (8 wires)

OSC:- Oscillator

To generate clock signal frequency.

Interrupts:-

To hold the current program and to run some other program the interrupts will be used.