Signal < Analog / Digital

Advantages of digital signal over Analog.

- More immune to noise
- More secure and reliable
- High data rate.
- Long distance transfer reliable
- Storage
- Cheaper than analog.

Numbers.

i. Binary -> It's base is 2
ii. Octal -> It's base is 8 from (0-7)
iii. Decimal -> It's base is 10 from (0-9)
iv. Hexa-Decimal -> It's base is 16 from (0-9, A, B, C, D, E, F)

* $(26)_{10} = (\quad)_2$.

$$
\begin{array}{r|l}
2 & 26 \to 0 \\
2 & 13 - 1 \\
2 & 6 - 0 \\
2 & 3 - 1 \\
 & 1
\end{array}
$$

$(26)_{10} = 11010$ //

* $(26.6125)_{10} = (\,?\,)_2$

$$\begin{array}{r|l} 2 & 26 - 0 \\ 2 & 13 - 1 \\ 2 & 6 - 0 \\ 2 & 3 - 1 \\ & 1 \end{array}$$

$0.6125 \times 2 = 1.234$
$0.234 \times 2 = 0.468$
$0.468 \times 2 = 0.936$
$0.936 \times 2 = 1.872$
$0.872 \times 2 = 1.744$

$\times$ $(26.6125)_{10} = 11010.110011$

* $(65.364)_8 = (\,?\,)_{10}$

$6 \times 8^1 + 5 \times 8^0 \cdot 3 \times 8^{-1} + 6 \times 8^{-2} + 4 \times 8^{-3}$

$48 + 5 \cdot 4761$

$(53.4761)_{10}$

* $(65.364)_8 = (\quad)_2$

$(110101 \cdot 011110100)_2$

* $(65.364)_8 = (\quad)_{16}$

$- \; 110101 \cdot 011110100$

$= \; 0011 \;\; 0101 \cdot 0111 \;\; 1010 \;\; 0000$

$= \; 3 \;\; 5 \cdot 7 \;\; A$

$= \; (35.7A)_{16}$

Find the value of n from number.

* $(239)_x = (645)_8$

$2n^2 + 3n + 9 = 6 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$

or, $2n^2 + 3n + 9 = 6 \times 64 + 32 + 5$

or, $2n^2 + 3n + 9 = 421$

or, $2n^2+3n-172=0$

* $(211)_n = (152)_8$

or, $2n^2+n+1 = 1\times8^2+5\times8^1+2\times8^0$

or, $2n^2+n+1 = 64+32+2$

or, $2n^2+n+1 = 106$

or, $2n^2+n-105 = 0$

or, $2n^2-14n+15n-105=0$

or, $2n(n-7)+15(n-7)=0$

or, $(n-7)(2n+15)=0$

Hence, $n-7=0$ $\quad$ $2n+15=0$

$\therefore n=7$ $\quad$ $2n=-15$

$n=\frac{-15}{2}$

Binary addition

$2\,|\,12 - 0$
$2\,|\,6 - 0$
$2\,|\,3 - 1$
1

* $(26)_{10} = (11010)_2$

$(12)_{10} = -(1100)_2$

1110

* ## Binary subtraction

$(26)_{10} = (11010)_2$

$(12)_{10} = +(1100)_2$

100110

Binary Multipication

```
        11010
       ×1100
        00000
       00000×
      11010××
   + 11010×××
  (100111000)₂
```

Representing of BCD, Gray code, Excess-3 code, ASCII

Binary to Gray-code

$$b_3 + b_2 + b_1 + b_0$$

$1000 \rightarrow 1 \longrightarrow 0 \longrightarrow 0 \longrightarrow 0$

1 1 0 0

$G_3 = b_3$ $G_2 = b_3 + b_2$ $G_1 = b_1 + b_2$ $G_0 = b_0 + b_1$

Gray code to binary

$1000 \rightarrow 1100$

$1001 \rightarrow 1101$

$(1101)_{Gray} = (1001)_2$

b_3	b_2	b_1	b_0
1	1	0	1
1	0	0	1

(A)

$(1100)_{Gray} = (1000)_2$

$$\begin{array}{cccc} 1 & 1 & 0 & 0 \\ \downarrow & + & + & + \\ 1 & 0 & 0 & 0 \end{array}$$

Excess-3 code

0 → +3 → 3
1 → +3 → 4
2 → +3 → 5
3 → +3 → 6
4 → +3 → 7.
5 → +3 → 8
6 → +3 → 9
7
8
9

10 +3

1 0
↓ ↓
3 3
↓ ↓
4 3
0100 0011

Alpha numeric code

→ ASCII (American standard code for information interchange)
→ EBCDIC (Extended BCD interchange code)
BCD = Binary coded decimal.

Addition

$(115)_{10} = (1110011)_2$

$(27)_{10} = + (11011)_2$

$(10001110)_2$

2 | 115 - 1
2 | 57 - 1
2 | 28 - 0
2 | 14 - 0
2 | 7 - 1
2 | 3 - 1
1

2 | 27 - 1
2 | 13 - 1
2 | 6 - 0
2 | 3 - 1
1

11011 (1's complement is 00100)

Substraction

```
 1110011
+  00100
 1110111
      +1
 1111000 -> (2's complement)
```

Binary substraction using 1's complement

$(26)_{10} - (12)_{10}$

Binary equivalent → 1's complement

binary equivalent → 1's complement

+

$(115)_{10} - (27)_{10}$

```
 1110011
 0011011 -> 1's complement
(1100100)
```

2 | 115 - 1
2 | 57 - 1
2 | 28 - 0
2 | 14 - 0
2 | 7 - 1
2 | 3 - 1
1

```
  1110011
- 1100100
 [1] 1001 11
       +1
 (1011000)₂
```

2 | 27 - 1
2 | 13 - 1
2 | 6 - 0
2 | 3 - 1
1

$(534)_8 \rightarrow BCD$

$5\times8^2 + 3\times8^1 + 4\times8^0$

$320 + 24 + 4$

$(348)_{10}$

$(001101001000)_{BCD}$

Error detection and correction code:

examples:
- Parity
  - odd
  - even
- check sum
- Cyclic Redundancy check

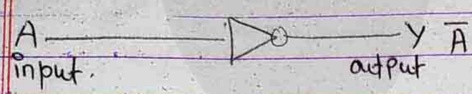
- Block codes
- Convolution codes

Combinantional logic :-

input { A → , B → } comb logic → output

logic gates
- Basic gates [eg:- Not, AND, OR]
- Universal gates [eg:- NAND, NOR]
- Special type gates [X-OR, X-NOR]

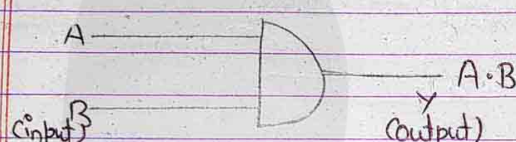
1. Not-Gate



Truth table.

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

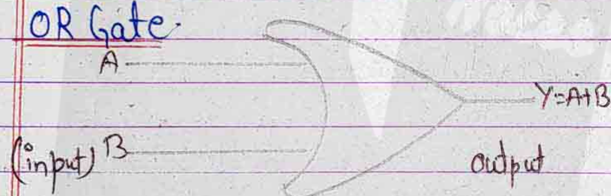
2. AND-Gate



Truthtable-

Input A	Input B	Output(Y)
0	0	0
0	1	0
1	0	0
1	1	1

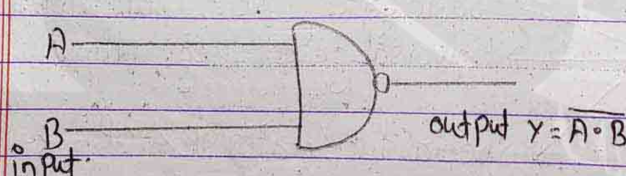
3. OR Gate.



Truthtable

Input A	Input B	OutPut(Y)
0	0	0
0	1	1
1	0	1
1	1	1

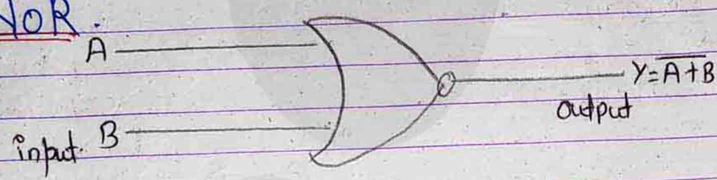
4. NAND Gate.



Truth table.

InPut A	InPut B	OutPut
0	0	1
0	1.	1
1	0	1
1	1	0

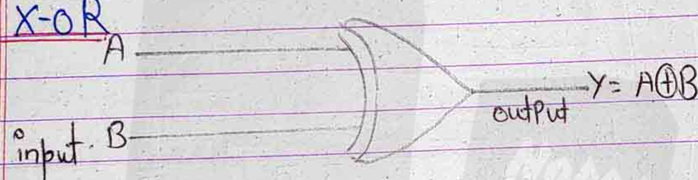
5. NOR.



Truth table

Input A	B	output (X)
0	0	1
0	1	0
1	0	0
1	1	0

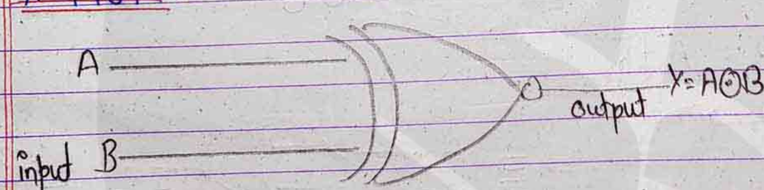
6. X-OR



Truth table.

Input A	B	output (X)
0	0	0
0	1	1
1	0	1
1	1	0

7. X-NOR.

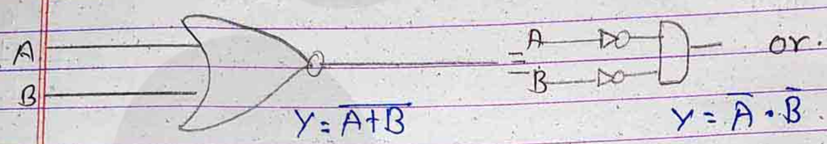


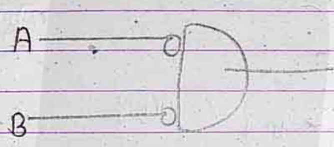
Truth table

Input A	B	output (X)
0	0	1
0	1	0
1	0	0
1	1	1

De-Morgan's Law

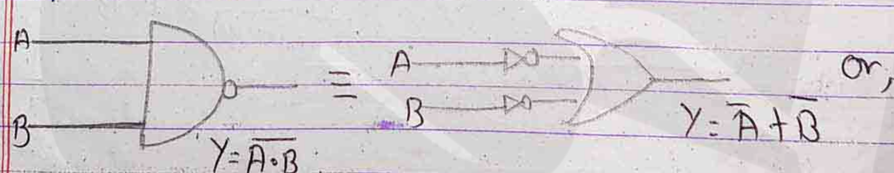
i. $\overline{A+B} = \bar{A} \cdot \bar{B}$

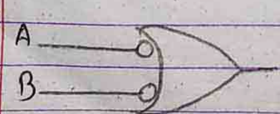




A	B	$\overline{A+B}$	$\bar{A}$ $\bar{B}$	$\bar{A} \cdot \bar{B}$
0	0	1	1 1	1
0	1	0	1 0	0
1	0	0	0 1	0
1	1	0	0 0	0

ii. $\overline{A \cdot B} = \bar{A} + \bar{B}$

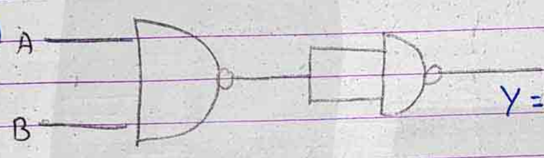




A	B	$\overline{A\cdot B}$	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

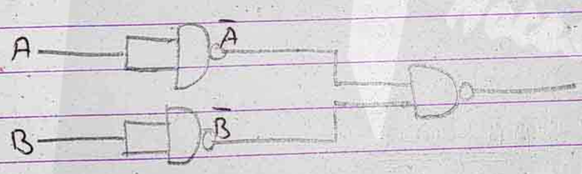
Construction of all logic gates using NAND only:

(ii) AND



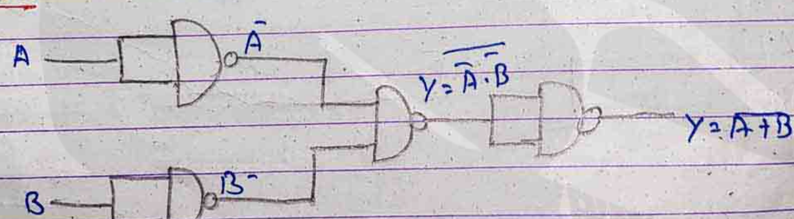
$Y = \overline{\overline{A\cdot B}} = A\cdot B$

(iii) OR



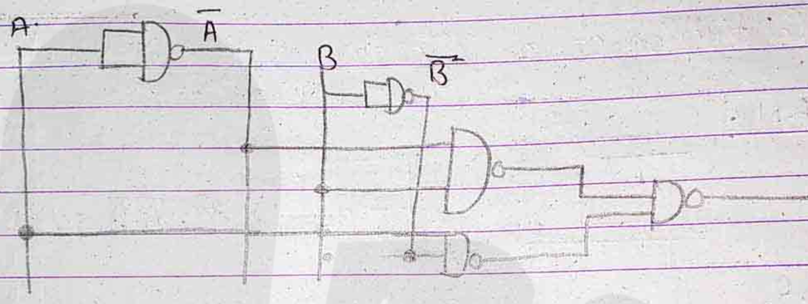
$Y = \overline{\bar{A}\cdot\bar{B}}$

iv. NOR



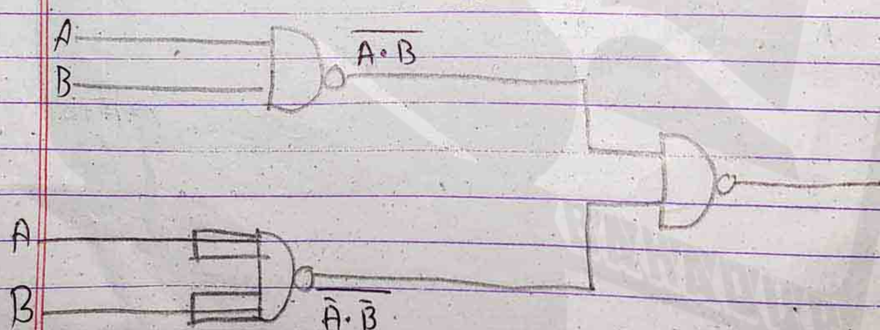
v. X-OR

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



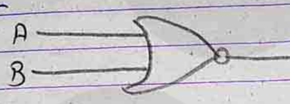
vi. X-NOR

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



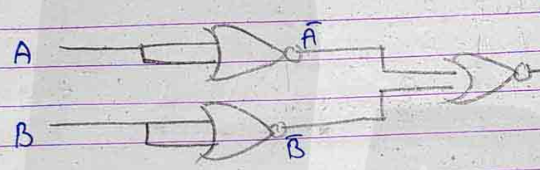
Construction of all logic gates using NOR only:

i. NOR :-



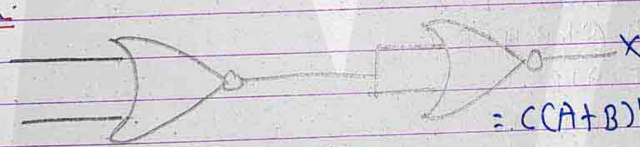
$Y = \overline{A \cdot B}$

ii. AND



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

iii. OR.



$= ((A+B)')' = A+B.$

Boolean expression or laws:

1. ~~Commutative~~ Associative law:

(i) $(A+B)+C = A+(B+C)$

(ii) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

2. Commutative law:

(i) $A \cdot B = B \cdot A$

(ii) $A+B = B+A$

3. OR Law :-

i) $A+0 = A$

ii) $A+1 = 1$

iii) $A+A = A$

iv) $A+\bar{A} = 1$

4. AND Law :-

i) $A \cdot 0 = 0$

ii) $A \cdot 1 = A$

iii) $A \cdot A = A$

iv) $A \cdot \bar{A} = 0$

5. $A + AB = A$

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

8. $A+BC = (A+B)(A+C)$

$= A \cdot A + A \cdot C + B \cdot A + B \cdot C$

$= A + AC + BA + BC$

$= A(1+C) + BA + BC$

$= A + BA + BC$

$= A(A+B) + BC$

$= A+BC$

9. $xy+xz(wz+w\bar{z})$

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

$= xy + xw$

$= x(y+w)$

10 $xy + xy' + x'y$
$= x(y+y') + x'y$
$= x \cdot 1 + x'y$
$= x + x'y$
$= x + y$

Questions.

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

ii. $xy + \bar{x}z + yz$
$= xy + \bar{x}z + yz(x+\bar{x})$
$= xy + \bar{x}z + xyz + \bar{x}yz$
$= xy(1+z) + \bar{x}z(1+y)$
$= xy(1+z) + \bar{x}z(1+y)$
$= xy + \bar{x}z$

iii. $AB + A(B+C) + B(B+C)$
$= AB + A \cdot B + AC + B \cdot B + BC$
$= AB + AC + B(B+C)$
$= B + AB + AC + BC$
$= B(1+A) + AC + BC$
$= B + BC + AC$
$= B(1+C) + AC$
$= B + AC$

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

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

$= (AC)(AC)(A\bar{C})$

$= AC \cdot A\bar{C}$

$= 0$

Boolean function

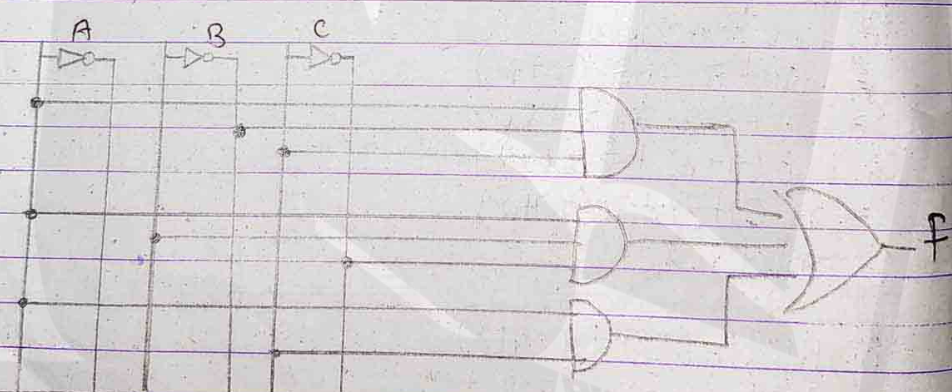
$\underbrace{f(A,B,C)}_{\text{Boolean function}} = \underbrace{\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC}_{\text{Boolean function}}$

literal

term

Sum of Production (SoP) =>

$f(A,B,C) = A\bar{B}C + AB\bar{C} + AC$



Standard (canonical form) = $A\bar{B}C + AB\bar{C} + AC(B+\bar{B})$

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

$= \underbrace{A\bar{B}C}_{\text{Minterm}} + AB\bar{C} + ABC$

Input A	B	C		minterm (mi)	F_1
0	0	0	$\bar{A}\bar{B}\bar{C}$	m0	0
0	0	1	$\bar{A}\bar{B}C$	m1	1
0	1	0	$\bar{A}B\bar{C}$	m2	0
0	1	1	$\bar{A}BC$	m3	0
1	0	0	$A\bar{B}\bar{C}$	m4	1
1	0	1	$A\bar{B}C$	m5	1
1	1	0	$AB\bar{C}$	m6	0
1	1	1	ABC	m7	1

$f_1 = m1 + m4 + m5 + m7$.

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

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

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

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

$= \underbrace{ABC}_{111} + \underbrace{AB\bar{C}}_{110} + \underbrace{A\bar{B}C}_{101} + \underbrace{\bar{A}\bar{B}C}_{001} + \underbrace{A\bar{B}\bar{C}}_{100} + \underbrace{\bar{A}\bar{B}C}_{001}$

m7 m6 m5 m4 m1.

$f = M_1, M_4, M_5, M_6, M_7$

$f = \sum m(1, 4, 5, 6, 7)$

Product of sum (Pos) =>

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

term literal

Input A	B	C		minterm	f
0	0	0	$(A+B+C)$	m0	0
0	0	1	$(\bar{A}+B+\bar{C})$	m1	1
0	1	0	$(A+\bar{B}+C)$	m2	0
0	1	1	$(\bar{A}+\bar{B}+\bar{C})$	m3	1
1	0	0	$(\bar{A}+B+C)$	m4	1
1	0	1	$(\bar{A}+B+\bar{C})$	m5	0
1	1	0	$(\bar{A}+\bar{B}+C)$	m6	0
1	1	1	$(A+B+C)$	m7	1

$f = m0 + m2 + m5 + m6$

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

$= 111 + 101 + 110 + 000$

m7 m5 m6 m0

(0, 5, 6, 7)

1, 2, 3, 4

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

P.T.O

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

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

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

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

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

$f = \Pi\ (0,2,4,5)$

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

$= \ (1,3,6,7)$

$f = \Sigma m\ (1,3,6,7)$

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

Difference between SoP and PoS

SoP	PoS
$f = (\sim) + \sim + \sim$	• $f = (\sim)\cdot(\sim)\cdot(\sim)$
$0 \to \bar{A}$	• $0 \to A$
$1 \to A$	$1 \to \bar{A}$
Min term (m_i)	• Max term (M_j)
term(output) - 1	• term(output) -> 0
$ABC \to 111$	• $(\bar{A}+\bar{B}+\bar{C}) = 111$
Canonical form $(A+\bar{A})$	• Canonical form $(A+B+C) = (A+B)(A+C)$

K-Map (Karnaugh Map)

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

$\bar{A} + C$

Questions

1. $f(x,y,z) = \Sigma(2,3,4,5)$

x \ yz	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$ 0			1	1
x 1	1	1		

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

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

2. $f(x,y,z) = \Sigma(3,4,6,7)$

x \ yz	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$ 0			1	
x 1	1		1	1

$= yz + x\bar{z}$

3. $f(n,y,z) = (0,2,4,5,6)$

$x \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	1			1
x	1	1		1

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

4. $f(n,y,z) = (1,2,3,5,7)$

$x \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$		1	1	1
x		1	1	

$= \bar{x}y + z$

5. $f(w,x,y,z) = (0,1,2,4,5,6,8,9,12,13,14)$

$wx \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	1	1		1
$\bar{w}x$	1	1		1
wx	1	1		1
$w\bar{x}$	1	1		

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

6. $Af(A,B,C,D) = \Sigma(0,1,2,5,8,9,10)$

$AB \backslash CD$	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1		1
$\bar{A}B$		1		
AB	~~1~~	~~1~~		
$A\bar{B}$	1	1		1

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

7. $f(x,y,z) = \Pi(0,2,5,7)$

$x \backslash yz$	$\bar{Y}Z$	$Y\bar{Z}$	$\overline{YZ}$	$\bar{Y}Z$
$\bar{X}$	0			0
X		0	0	

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

8. $f(w,x,y,z) = \Sigma(1,3,7,11,15)$
$D(w,x,y,z) = \Sigma(0,2,5)$

$wx \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	x	1	1	x
$\bar{w}x$		x	1	
wx			1	
$w\bar{x}$			1	

$= \bar{w}\bar{x} + yz$

9. $f(x,y,z) = \Sigma(0,1,2,4,5)$
$d(x,y,z) = \Sigma(3,6,7)$

x \ yz	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	1	1	X	1
x	1	1	X	X

$= 1$

10. $f(A,B,C,D) = \Sigma(0,6,8,13,14)$
$f(A,B,C,D) = \Sigma(2,4,10)$

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1			X
$\bar{A}B$	X			1
AB		1		1
$A\bar{B}$	1			X

$\bar{B}\bar{D} + AB\bar{C}D + C\bar{D}$

Simplify the following Boolean functions, using k-maps.

a. $f(w,x,y,z) = \Sigma(0,1,4,5,8,12)$

wx \ yz	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	1	1		
$\bar{w}x$	1	1		
wx	1			
$w\bar{x}$	1			

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

b. $f(w,x,y,z) = \Sigma(4,6,9,11,12,14)$

$wx \backslash yz$	$\bar{y}\bar{z}$	$\bar{x}z$	yz	$x\bar{z}$
$\bar{w}\bar{x}$				
$\bar{w}x$	1			1
wx	1			1
$w\bar{x}$		1	1	

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

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

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

c. $f(w,x,y,z) = \Sigma(1,3,5,6,7,9,11)$

$wx \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$x\bar{z}$
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$		1	1	1
wx				
$w\bar{x}$		1	1	

$\bar{w}z + \bar{w}xy + \bar{x}z$

$= \bar{w}z + \cancel{\bar{w}xy} + \cancel{\bar{x}z}$

d. $f(w,x,y,z) = \Sigma(2,6,7,10,14,15$

$wx \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	xz	$y\bar{z}$
$\bar{w}\bar{x}$				1
$\bar{w}x$			1	1
wx			1	1
$w\bar{x}$				1

$xy + y\bar{z}$

Implement the following Boolean function F, together with the don't care conditions, using not more than two NOR Gates

$f(A,B,C,D) = \Sigma(2,4,6,10,12)$

$d(A,B,C,D) = \Sigma(0,8,9,13)$

$(A')'$

$AB \backslash CD$	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	X			1
$\bar{A}B$	1			1
AB	1	X		
$A\bar{B}$	X	X		1

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

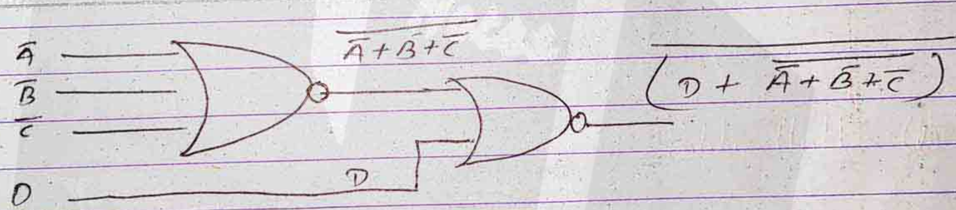
$= \left\{\overline{\overline{\bar{D}\cdot(\bar{A}+\bar{B}+\bar{C})}}\right\}$

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

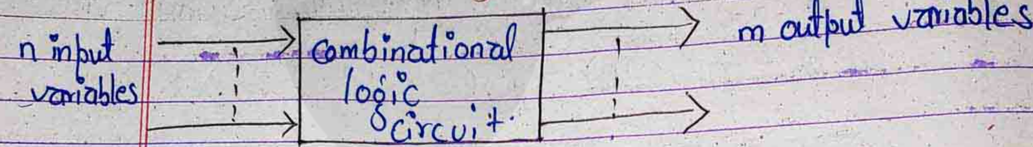
$\overline{(A+B)} = \bar{A}\cdot\bar{B}$

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

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



Combinational logic circuit

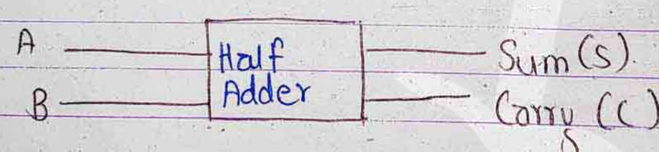


Block diagram of combinational logic circuit.

Adder → Half adder, Full adder

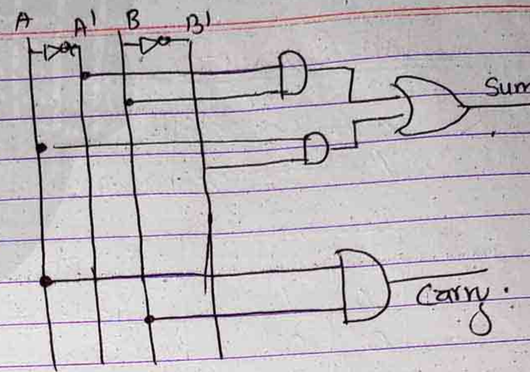
Subtractor → Half subtractor, full subtractor.

b) Half adder

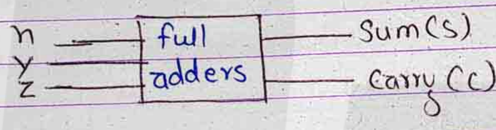


Input		Out Put	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1.

$$Sum(S) = \bar{A}B + A\bar{B}$$
$$= A \oplus B$$
$$Carry\ (C) = AB$$



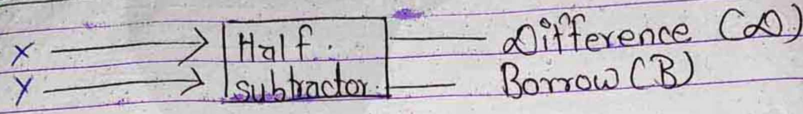
b. full adder



Input			Output	
x	y	z	Sum(S)	Carry(C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Sum(S) \Rightarrow \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$
$$\Rightarrow \bar{x}(\bar{y}z + y\bar{z}) + x(\bar{y}\bar{z} + yz)$$
$$\Rightarrow \bar{x}(y \oplus z) + x(\overline{y \oplus z})$$
$$\Rightarrow (x \oplus y \oplus z)$$

Half Subtractor

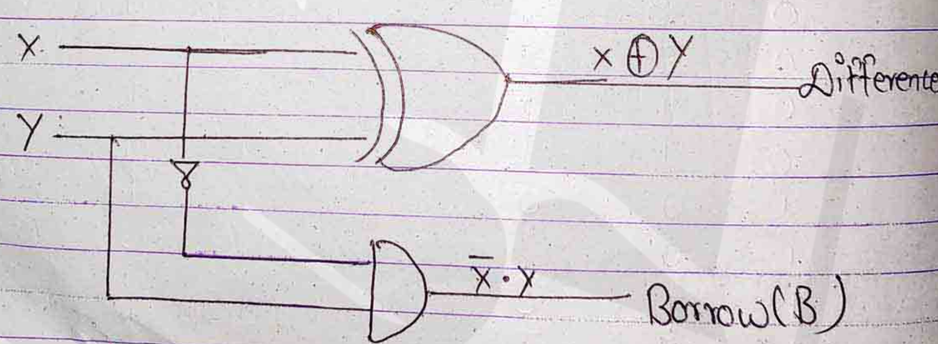


Input		Output	
X	Y	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

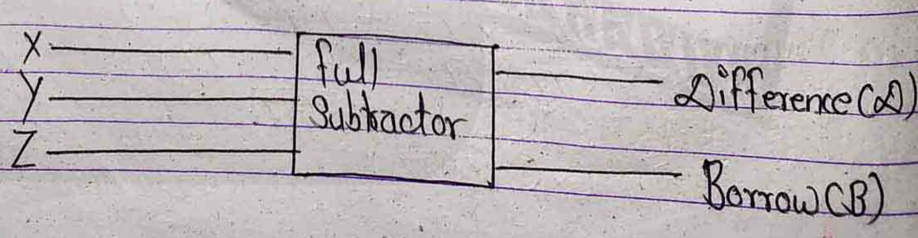
Difference (D) = $\bar{x}y + x\bar{y}$

$= x \oplus y$

Borrow (B) = $\bar{x}y$



full subtractor



Input			Output	
x	y	z	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Difference (D) $\Rightarrow x'y'z + x'yz' + xy'z' + xyz$

$\Rightarrow z'(x'y + xy') + z(x'y' + xy)$

$\Rightarrow z'(x \oplus y) + z(\overline{x \oplus y})$

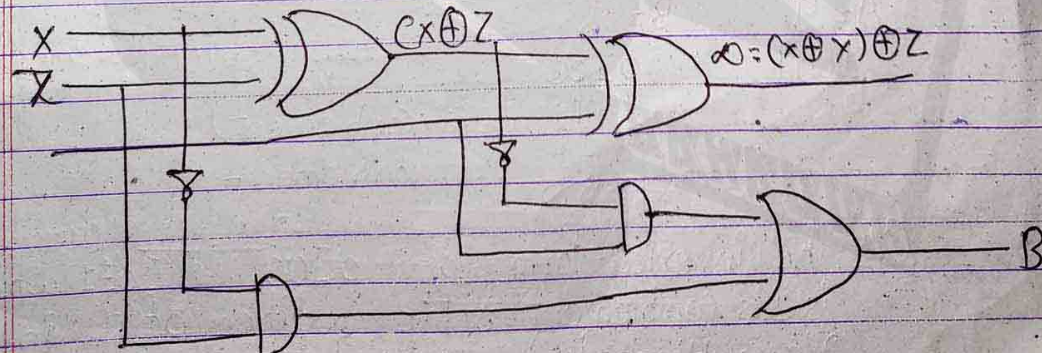
$\Rightarrow x \oplus y \oplus z$

Borrow (B) $\Rightarrow \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + xyz$

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

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

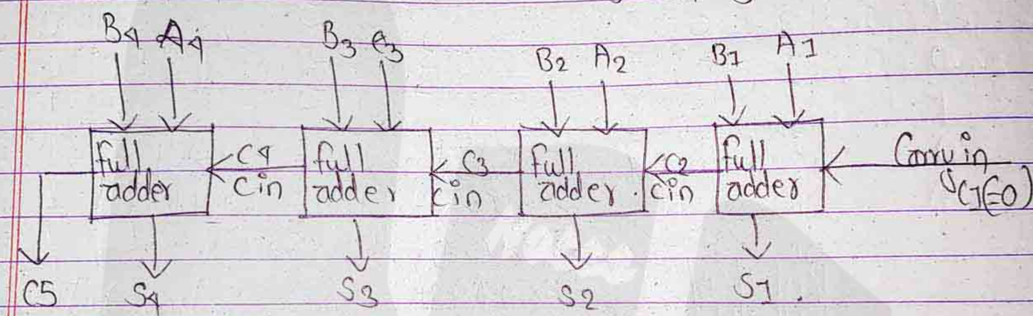
$= y(\overline{x \oplus z}) + \bar{x}z$



i) Binary Parallel adder
ii) Binary Parallel subtractor
iii) Binary adder and subtractor

i. 4-bit binary parallel adder →

A ⇒ $A_4 A_3 A_2 A_1$
B ⇒ $B_4 B_3 B_2 B_1$



[∵ Cin = Carry in]

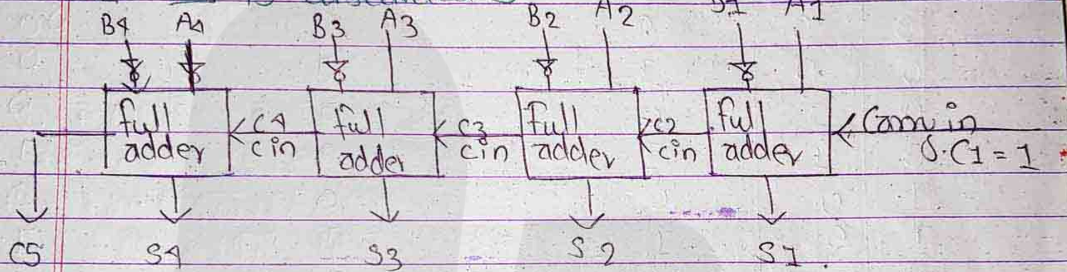
Binary Parallel subtractor.

A = 9 B = 4
1001 0100
1011 (1's complement)
+1 (2's complement.
1100

(∵ The inverted form of 0100 is 1011 which is 1's complement & the addition of 1 in LSB is 2's complement)

$$\begin{array}{r} 1011 \\ -\,1100 \\ \hline 0001 \end{array} \qquad \begin{array}{r} 1001 \\ +\,1100 \\ \hline \boxed{1}0101 \end{array}$$

[∵ $\boxed{1}$ is discarded]



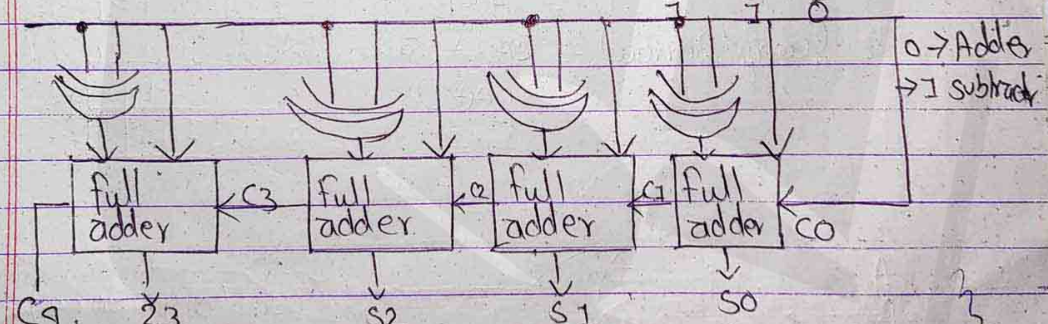
iii. Binary adder and subtractor.

$B \oplus 0 = B$

$B \oplus 1 = \overline{B}$

XOR

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



P.T.O

Adder substractor =>

C=0 , addition

C=1 subtraction

C	A	B	Sum/difference	Carry / Borrow
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	0

Y_1 (sum/difference) $= \bar{A}B + A\bar{B}$

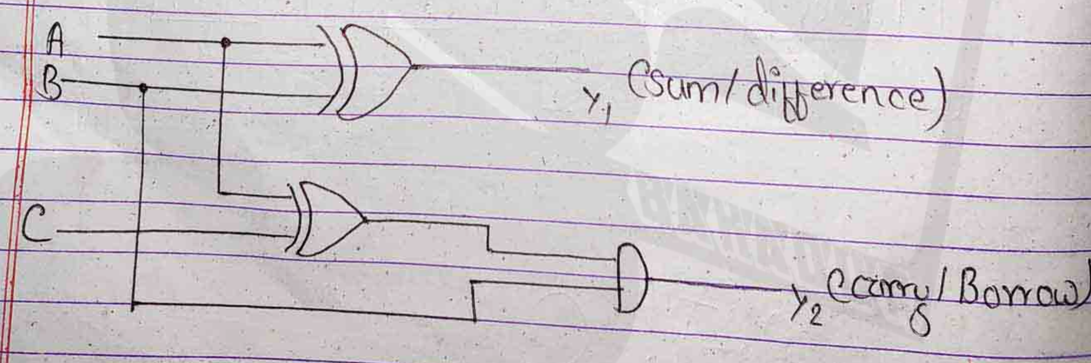
$= A \oplus B$

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

Y_2 (carry/Borrow) $= \bar{C}AB + C\bar{A}B$

$= B(A \oplus C)$

C \ AB	00	01	11	10
0			1	
1		1		

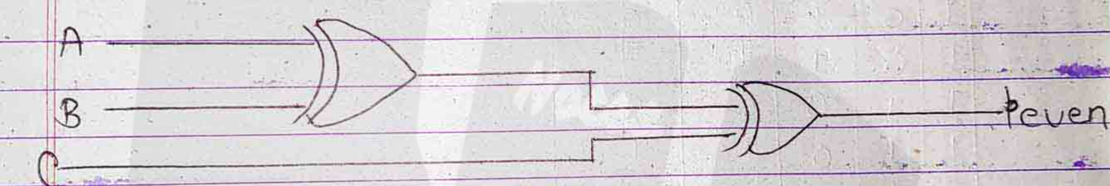


Parity Generator and checker.

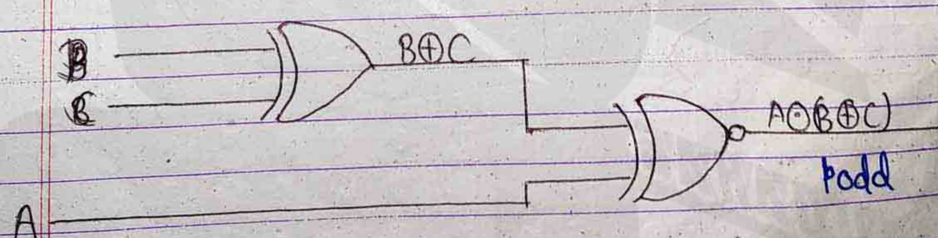
Information

A	B	C	Podd	Peven
0	0	0	1	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	0	1

$$P_{even} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$$
$$= \bar{A}(B\oplus C) + A(\overline{B\oplus C})$$
$$= A\oplus B\oplus C$$



$$P_{odd} \Rightarrow \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$$
$$= \bar{A}(\bar{B}\bar{C} + BC) + A(\bar{B}C + B\bar{C})$$
$$= \bar{A}(\overline{B\oplus C}) + A(B\oplus C)$$
$$= A\odot(B\oplus C)$$



Parity checker (even):

A	B	C	P	P Check
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

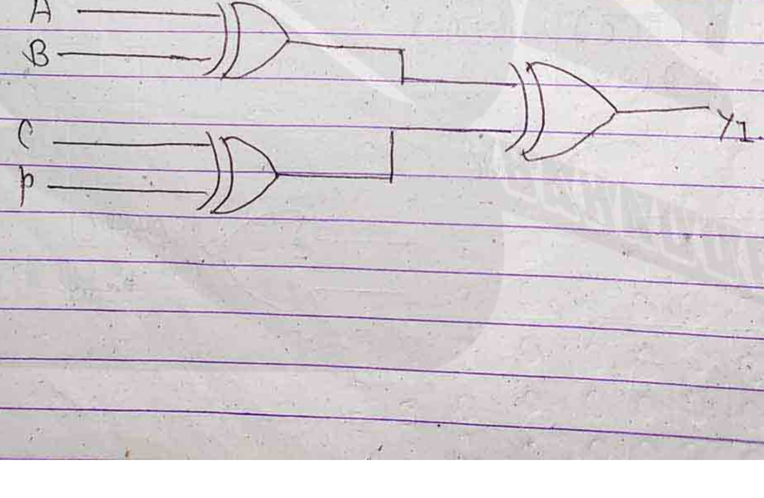
AB \ CP	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$Y_1 = \bar{A}\bar{B}\bar{C}P + \bar{A}\bar{B}C\bar{P} + AB\bar{C}\bar{P} + \bar{A}BCP +$
$AB\bar{C}P + ABC\bar{P} + A\bar{B}\bar{C}\bar{P} + A\bar{B}CP$

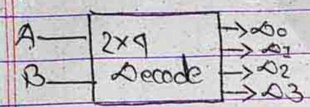
$= \bar{A}\bar{B}(C\oplus P) + \bar{A}B(\overline{C\oplus P}) + AB(C\oplus P) + A\bar{B}(C\oplus P)$

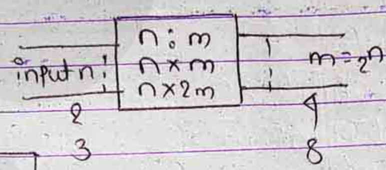
$= (C\oplus P)(\overline{A\oplus B}) + (\overline{C\oplus P})(A\oplus B)$

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



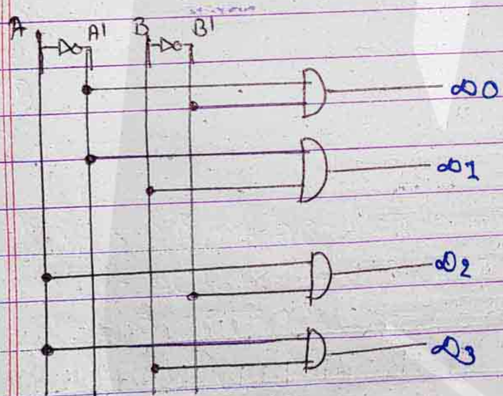
Encode & Decode :-



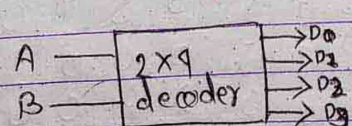


Input		Output			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$D_0 = \bar{A}\bar{B}$, $D_1 = \bar{A}B$, $D_2 = A\bar{B}$, $D_3 = AB$

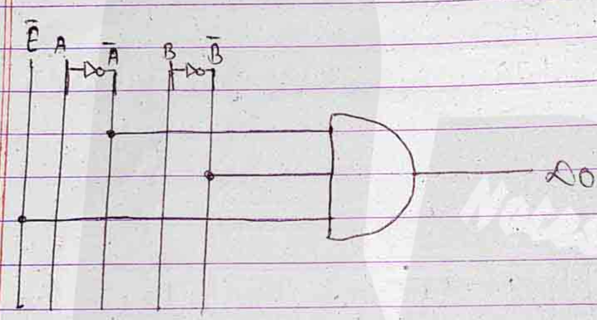


Decoder with enable pin :-

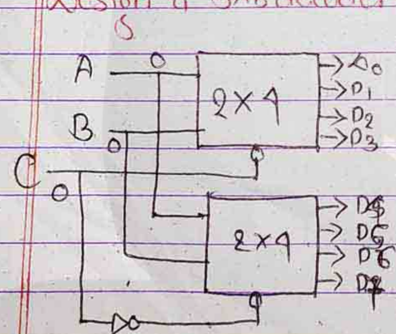


Input			Output			
E	A	B	D0	D1	D2	D3
1	x	x	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	0	0	0	0	1

$D_0 = A'B'E'$, $D_1 = A'BE$, $D_2 = AB'E'$, $D_3 = AB'E'$



Design a 3x8 decoder using 2x4 decoder.

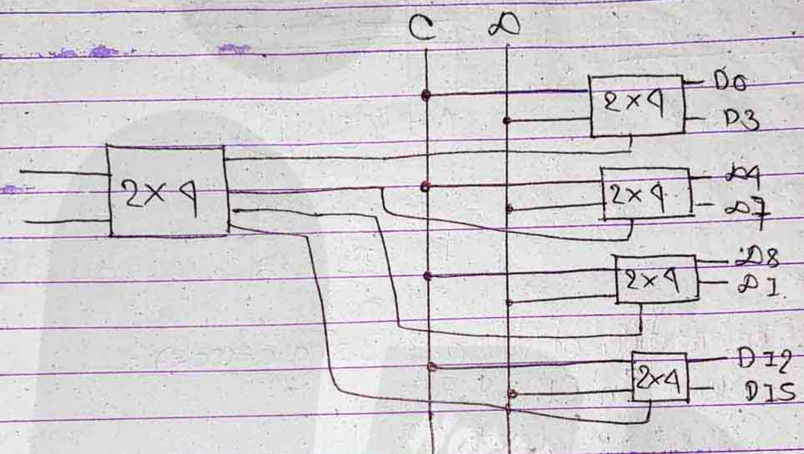


Let E be the MSB for input.

E	A	B	D0	D1	D2	D3	D4	D5	D6	D7	~~D8~~
0	0	0	1	0	0	0	0	0	0	0	
0	0	1	0	1	0	0	0	0	0	0	
0	1	0	0	0	1	0	0	0	0	0	
0	1	1	0	0	0	1	0	0	0	0	
1	0	0	0	0	0	0	1	0	0	0	
1	0	1	0	0	0	0	0	1	0	0	
1	1	0	0	0	0	0	0	0	1	0	
1	1	1	0	0	0	0	0	0	0	1	

(∴ Note :- MSB ⇒ Most significant Bits)

Design a 4×16 decoder using 2×4 decoder.



Implementation of Boolean function using decoder :-

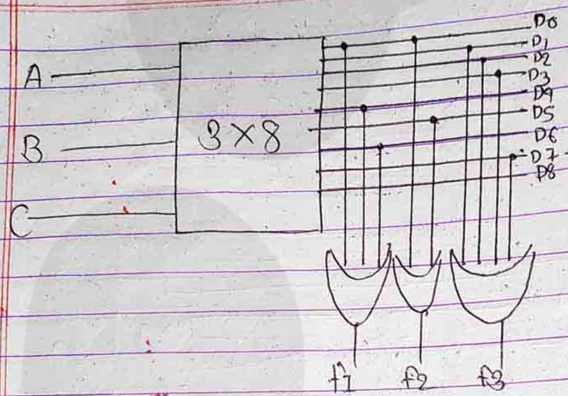
$F = \sum m(0, 4, 6)$, $f_2 = \sum m(0, 5)$ $f_3 = \sum m(1, 2, 5, 7)$

$f = \sum m(0, 4, 6)$
$= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$ ——— (i)

$f_2 = \sum(0, 5)$
$= \bar{A}\bar{B}\bar{C} + A\bar{B}C$ ——— (ii)

$f_3 = \sum m(1, 2, 5, 7)$
$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + ABC$ ——— (iii)

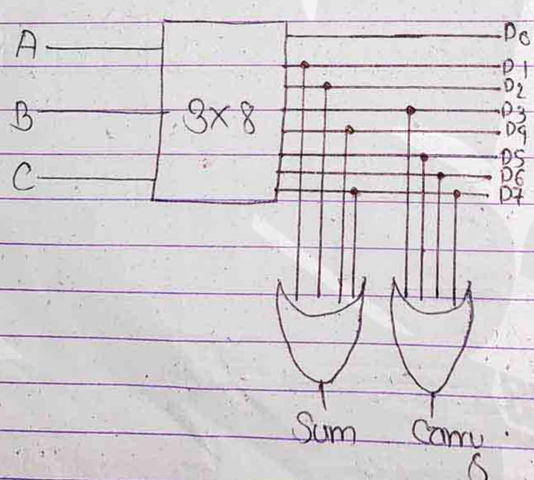
(P.T.O)



Implementation of full adder using decoder

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

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



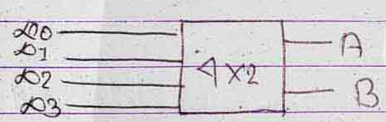
(P.T.O)

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

$Sum = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

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

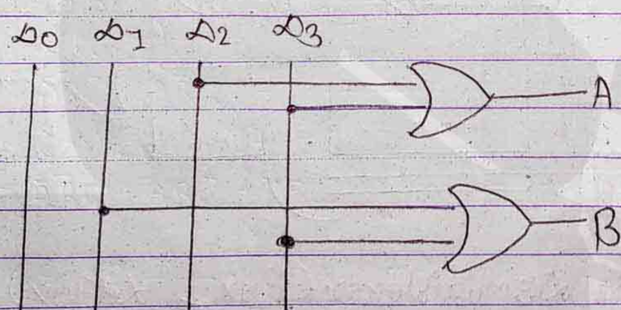
* Encoder



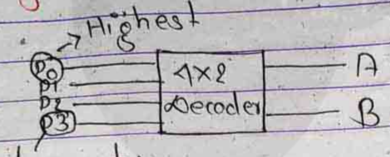
Input				output	
D0	D1	D2	D3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

$A = D_2 \oplus D_3$

$B = D_1 + D_3$



Priority Encoder :-

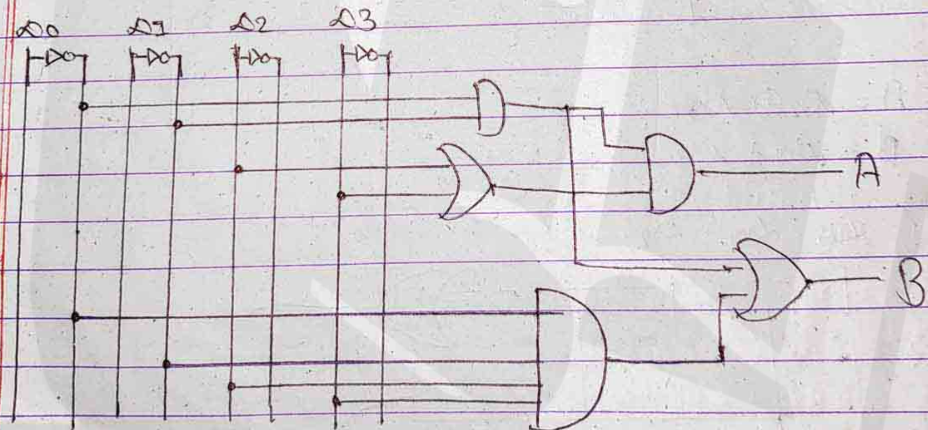


input				output	
D_0	D_1	D_2	D_3	A	B
1	X	X	X	0	0
0	1	X	X	0	1
0	0	1	X	1	0
0	0	0	1	1	1

$A = \overline{D_0}\,\overline{D_1}\,D_2 + \overline{D_0}\,\overline{D_1}\,\overline{D_2}\,D_3$

$= \overline{D_0}\,\overline{D_1}\,(D_2 + D_3)$

$B = \overline{D_0}\,D_1 + \overline{D_0}\,\overline{D_1}\,\overline{D_2}\,D_3$

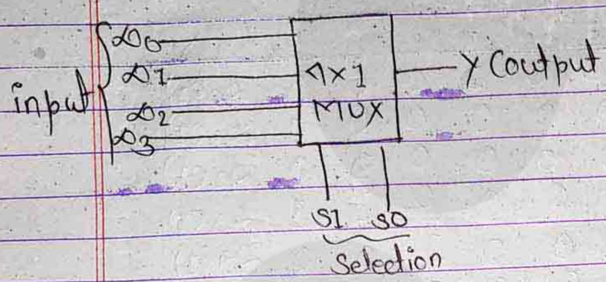


Multiplexer and Demultiplexer (Mux & Demux)

Multiplexer :

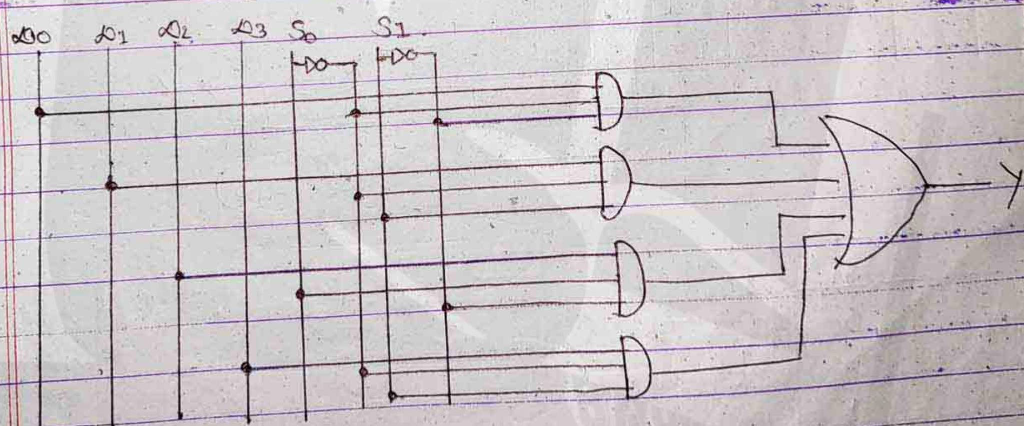
* Many to one

(4x1 Multiplexer)



Input				selection		output
D0	D1	D2	D3	S1	S0	Y
1	0	0	0	0	0	D0(1)
0	1	0	0	0	1	D1(1)
0	0	1	0	1	0	D2(1)
0	0	0	1	1	1	D3(1)

$Y = D_0\overline{S_1}\,\overline{S_0} + D_1\overline{S_1}S_0 + D_2S_1\overline{S_0} + D_3S_1S_0$



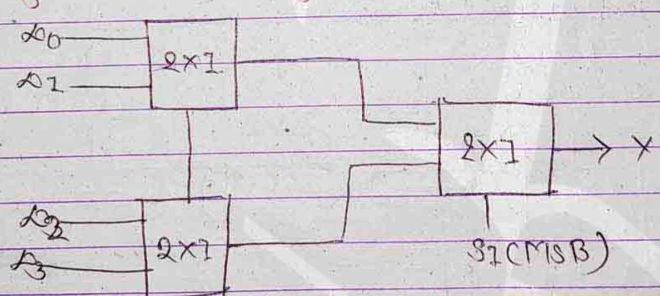
(P.T.O)

(8x1)

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	S_0	S_1	S_2	Y
1	0	0	0	0	0	0	0	0	0	0	$D_0(1)$
0	1	0	0	0	0	0	0	0	0	1	$D_1(1)$
0	0	1	0	0	0	0	0	0	1	0	$D_2(1)$
0	0	0	1	0	0	0	0	0	1	1	$D_3(1)$
0	0	0	0	1	0	0	0	1	0	0	$D_4(1)$
0	0	0	0	0	1	0	0	1	0	1	$D_5(1)$
0	0	0	0	0	0	1	0	1	1	0	$D_6(1)$
0	0	0	0	0	0	0	1	1	1	1	$D_7(1)$

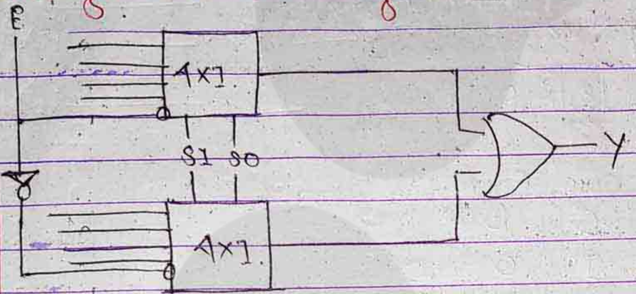
$Y = D_0\bar{S_0}\bar{S_1}\bar{S_2} + D_1\bar{S_0}\bar{S_1}S_2 + D_2\bar{S_0}S_1\bar{S_2} + D_3\bar{S_0}\bar{S_1}\bar{S_2} + D_4 S_0\bar{S_1}\bar{S_2} + D_5 S_0\bar{S_1}S_2$
$+ D_6 S_0 S_1\bar{S_2} + D_7 S_0 S_1 S_2$

* Design 4x1 Mux using 2x1.

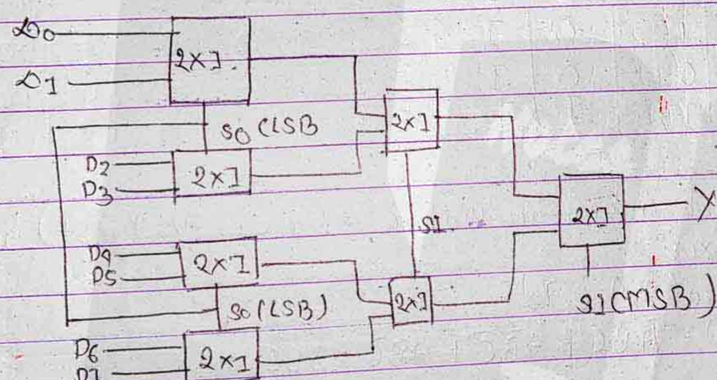


S_1	S_0	Input
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

* Design 8x1 MUX using 4x1.



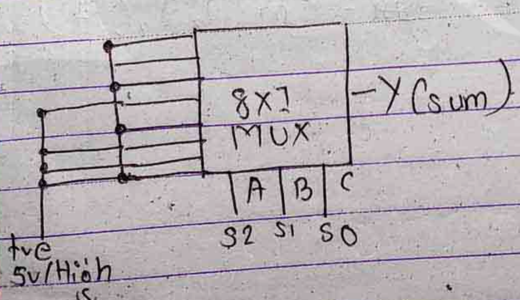
using 2x1

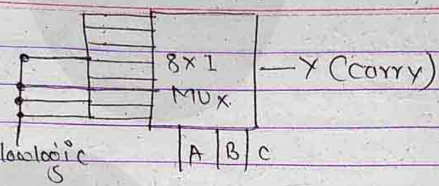


\# Design full adder using multiplexer

fsum = (1,2,4,7)

fcarry = (3,5,6,7)



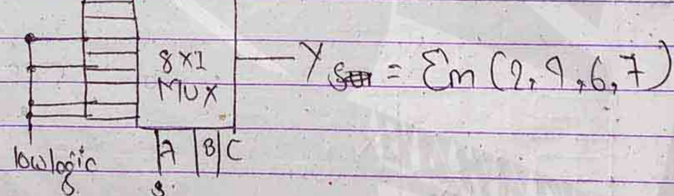


A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

\# Implement a boolean function f= ~~So~~ $\Sigma m(2,4,6,7)$ by using Multiplexor

$f = \Sigma m(2,4,6,7)$

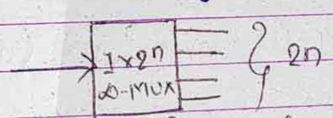
$f(ABC) = \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$



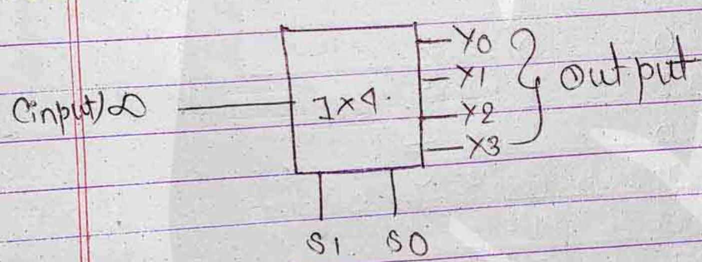
A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

DeMultiplexer (DeMux)

→ One to Many



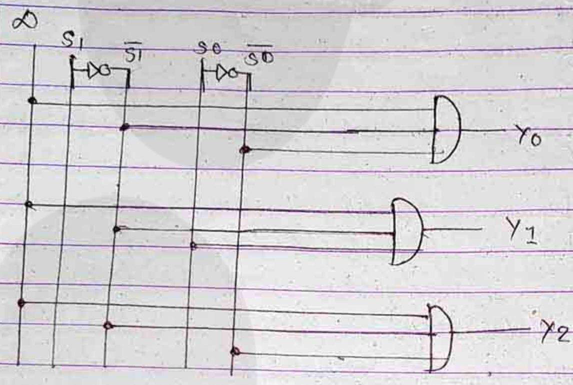
n: no of selection



Input	Selection		Output			
			Y0	Y1	Y2	Y3
D	0	0	D_1	0	0	0
D	0	1	0	D_2	0	0
D	1	0	0	0	D_3	0
D	1	1	0	0	0	D_4

$Y_0 = D\bar{S_1}\bar{S_0}$ $\quad$ $Y_2 = D S_1 \bar{S_0}$

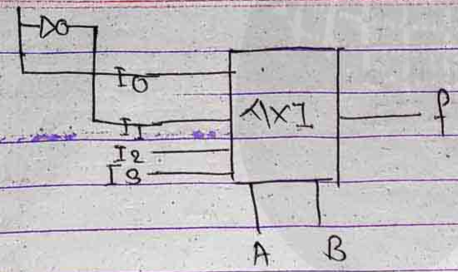
$Y_1 = D\bar{S_1}S_0$ $\quad$ $Y_3 = D S_1 S_0$



1. Implement $f(A,B,C) = \Sigma m(1,2,4,5)$ using Mux (using 4×1 Mux)

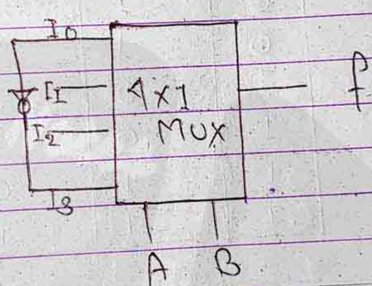
$f(A,B,C) = \Sigma m(1,2,4,5)$

	A	B	C	f	
I_0	0	0	0	0	C
	0	0	1	1	
I_1	0	1	0	1	$\bar{C}$
	0	1	1	0	
I_2	1	0	0	1	1
	1	0	1	1	
I_3	1	1	0	0	0
	1	1	1	0	

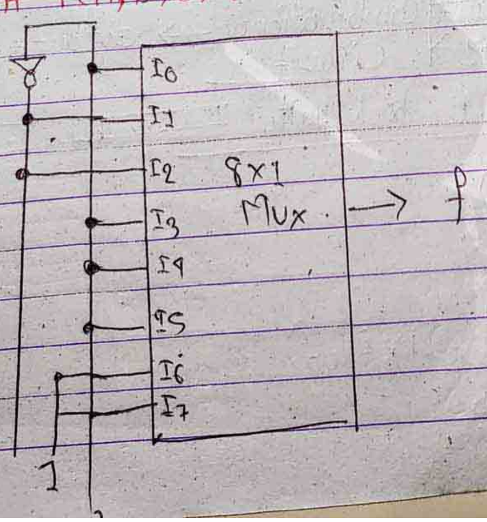


2. Implement $f(A,B,C) = \sum m(1,3,5,6)$ using MUX using 4x1.

A	B	C	f	
0	0	0	0	C
0	0	1	1	
0	1	0	0	C
0	1	1	1	
1	0	0	0	C
1	0	1	1	
1	1	0	1	$\overline{C}$
1	1	1	0	

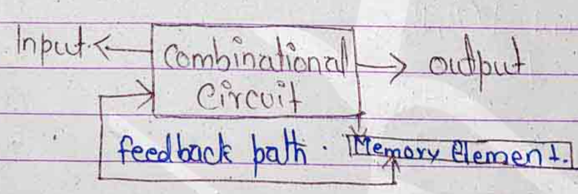


3. Implement $f(A,B,C,D) = \sum(1,2,4,11,12,13,14,15)$



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

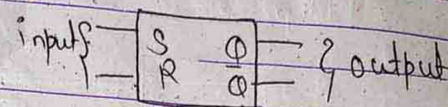
Sequential Circuit :-

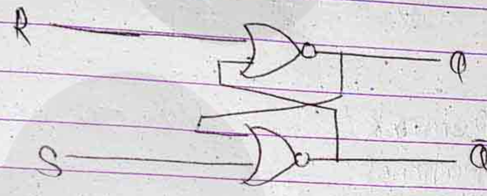


1. S-R (Set-Reset)
2. Delay (D)
3. J-k.
4. Toggle flip flop (T-flip flop)
5. Master slave flip flop

1. S-R flip flop ⇒

S-R latch
- NOR based
- NAND based.



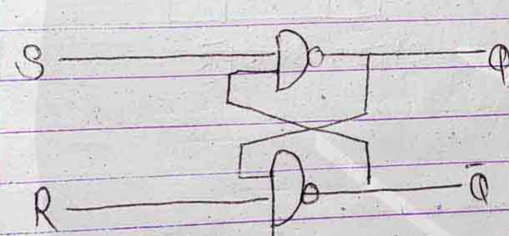


NOR

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

S	R	Q_t	$\bar{Q}_t$	Remarks
0	0	Q_{t-1}	$\bar{Q}_{t-1}$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	Invalid.

NAND based.

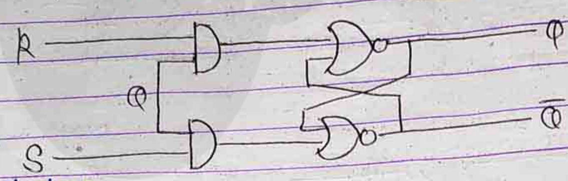


NAND

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

S	R	Q_t	$\bar{Q}_t$	Remarks
0	0	?	?	Invalid.
0	1	1	0	set
1	0	0	1	Reset
1	1	Q_{t-1}	$\overline{Q_{t-1}}$	No change

S-R flip flop ⇒



Truth table

S	R	Q_t	$\overline{Q_t}$	Remark
0	0	Q_{t-1}	$\overline{Q_{t-1}}$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	?	?	Invalid

Characteristic table

Q_t	S	R	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

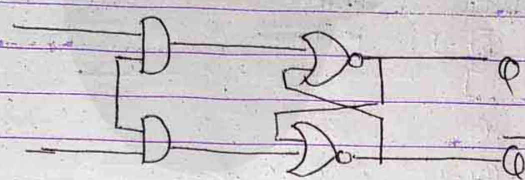
Q_t \ SR	00	01	11	10
0			X	1
1	1		X	1

$Q_{t+1} = S + Q_t\overline{R}$

Execution table

Q_t	Q_{t+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D flip flop :-



Truthtable

D	Qt
0	0
1	1

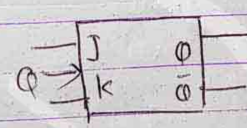
characteristics table

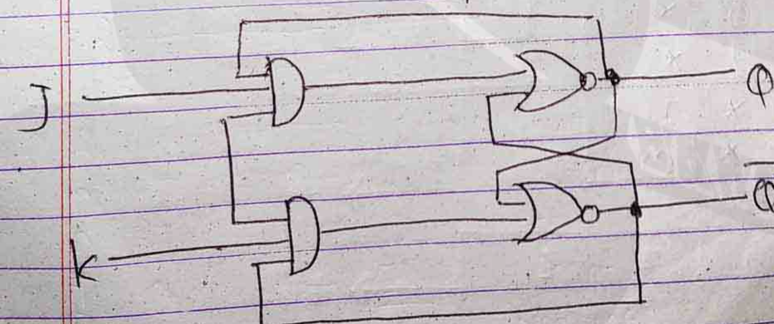
Qt	D	Qt+1
0	0	0
0	1	1
1	0	0
1	1	1

Execution table

Qt	Qt+1	D
0	0	0
0	1	1
1	0	0
1	1	1

Jk flip-flop



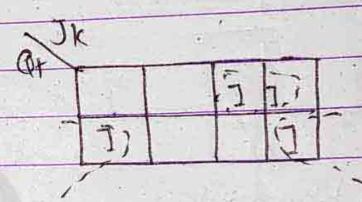


Truth table :-

J	k	Q_t	$\overline{Q_t}$	Remarks
0	0	Q_{t-1}	$\overline{Q_{t-1}}$	Nochange
0	1	0	1	Reset
1	0	1	0	Set
1	1	$\overline{Q_{t+1}}$	Q_{t-1}	toggle.

Characteristics table

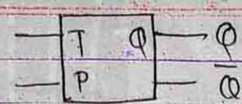
Q_t	J	k	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

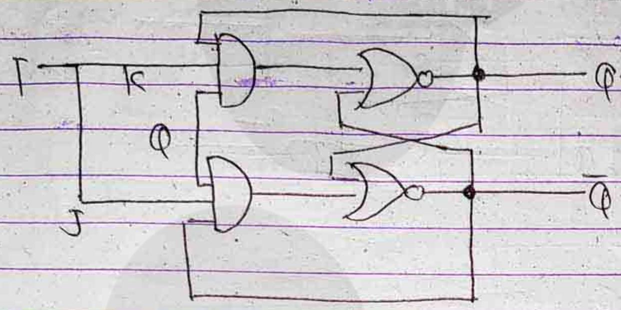


Execution table

Q_t	Q_{t+1}	J	k
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

T-flipflop (Toggle) ⇒





Truth table

T	Qt
0	0
1	1

Characteristic table

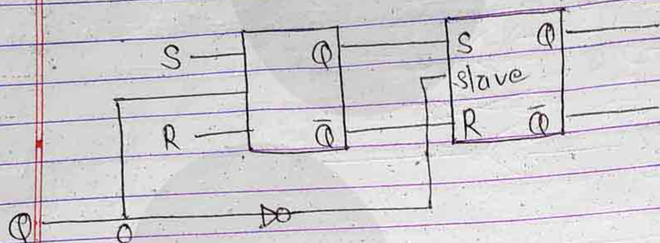
Qt	T	Qt+1	Remarks
0	0	0	No change
0	1	1	Toggle
1	0	1	No change
1	1	0	Toggle

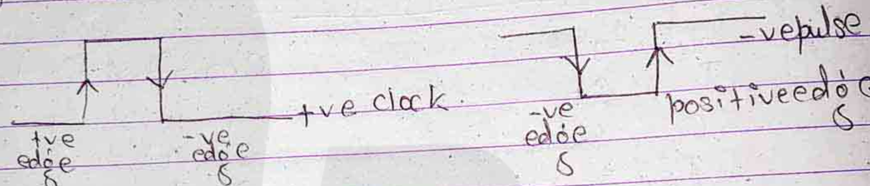
Execution table

Qt	Qt+1	T
0	0	0
0	1	1
1	0	1
1	1	0

(p. T. O)

Master-slave flip-flop

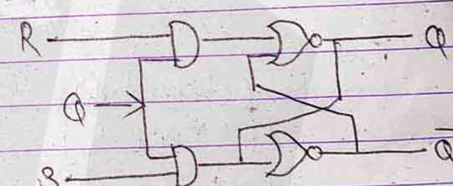


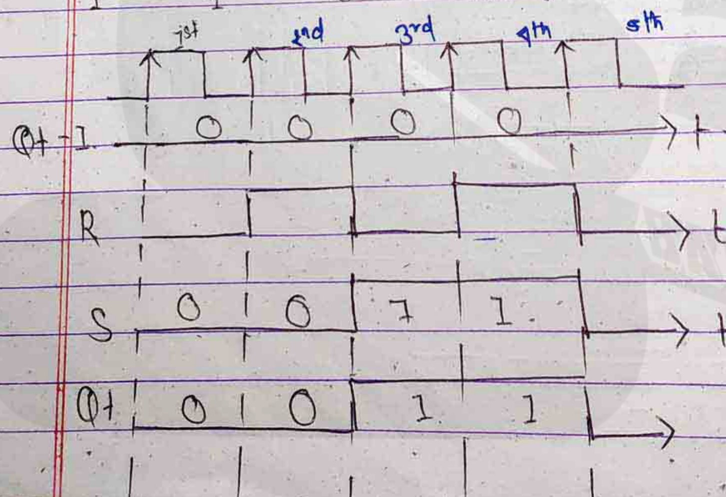


Timing diagram of flip-flop :-

S-R :

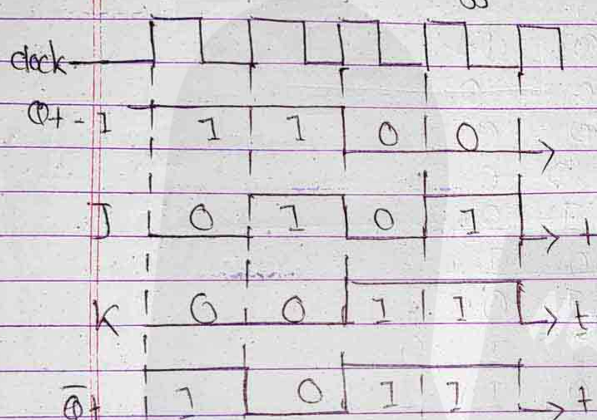
S	k	Qt	Remarks
0	0	Qt-1	No change
0	1	0	Reset
1	0	1	Set
1	1	0	Invalid





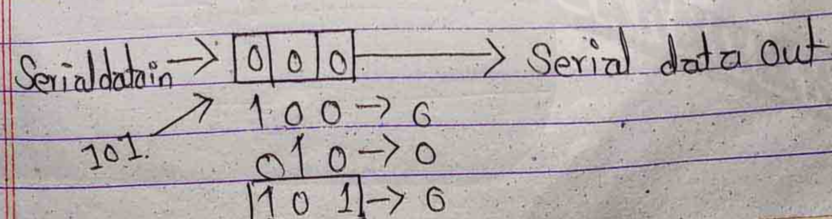
J-k

J	k	Q_t	Remarks
0	0	Q_{t-1}	No change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q_{t+1}}$	toggle

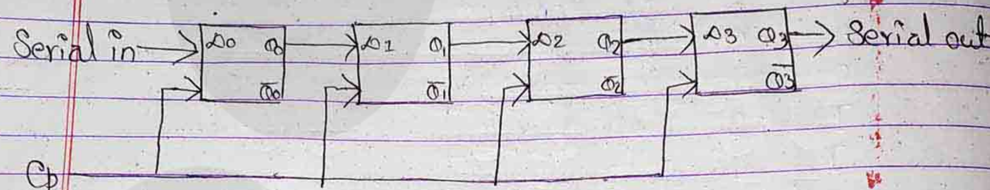


Shift - Register

i) Serial In Serial out (SISO)
ii) Serial In Parallel out (SIPO)
iii) Parallel In Parallel out (PIPO)
iv) Parallel In Serial out (PISO)

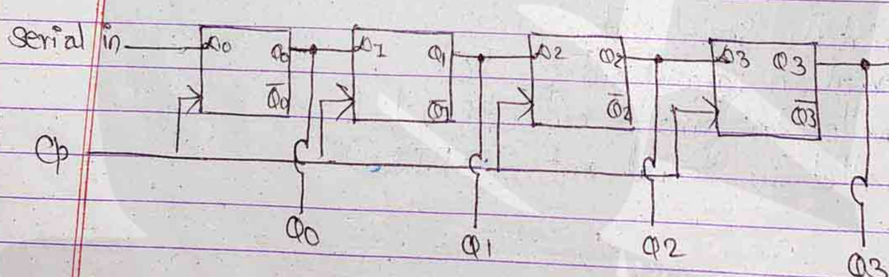


1. Serial In Serial out



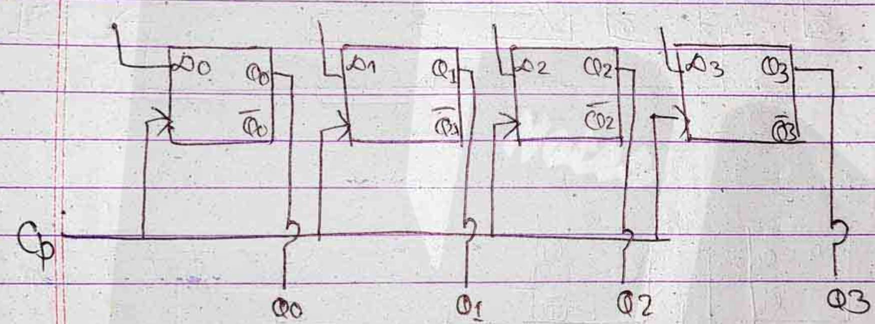
Clock	Q_0	Q_1	Q_2	Q_3	
Initial	0	0	0	0	
1st	1	0	0	0	
2nd	0	1	0	0	
3rd	1	0	1	0	
4th	1	1	0	1	
5th	0	1	1	0	1
6th	0	0	1	1	0
7th	0	0	0	1	1
8th	0	0	0	0	1

2. Serial In Parallel Out :-



Clock	Q_0	Q_1	Q_2	Q_3	(0011)
Initial	0	0	0	0	
1st	1	0	0	0	
2nd	1	1	0	0	
3rd	0	1	1	0	
4th	0	0	1	1	
	Q_0	Q_1	Q_2	Q_3	

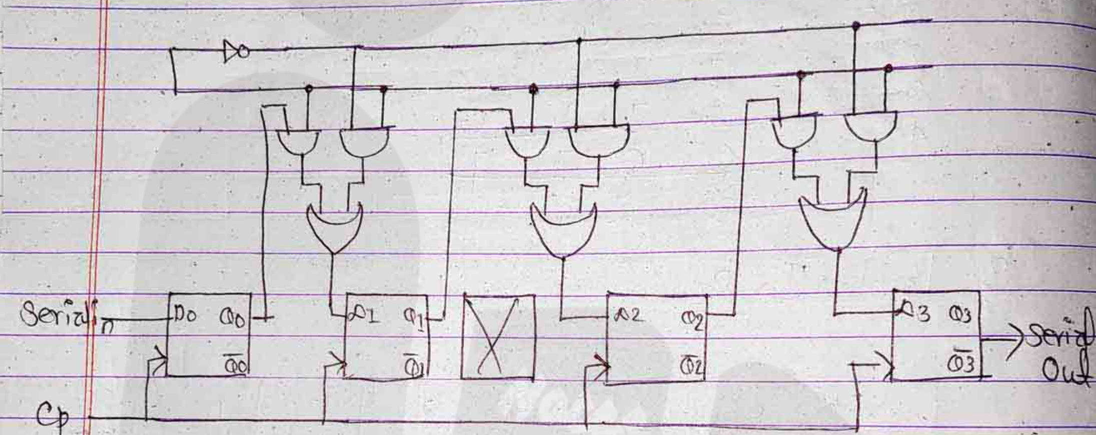
3. Parallel In Parallel out



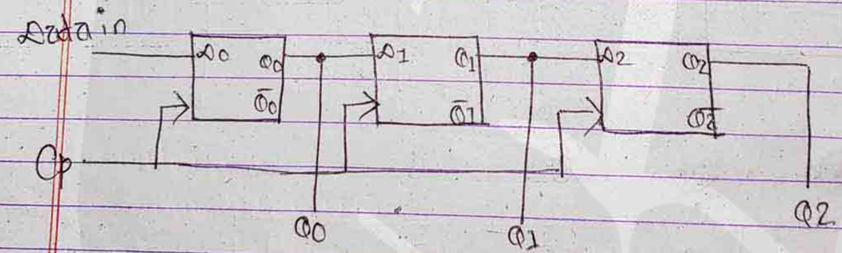
Clock	Q_0	Q_1	Q_2	Q_3
Initial	0	0	0	0
1st	0	0	1	1
	Q_0	Q_1	Q_2	Q_3

(P.T.O)

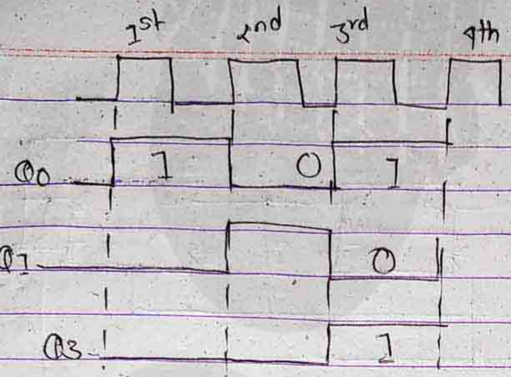
4. Parallel in Serial out :-



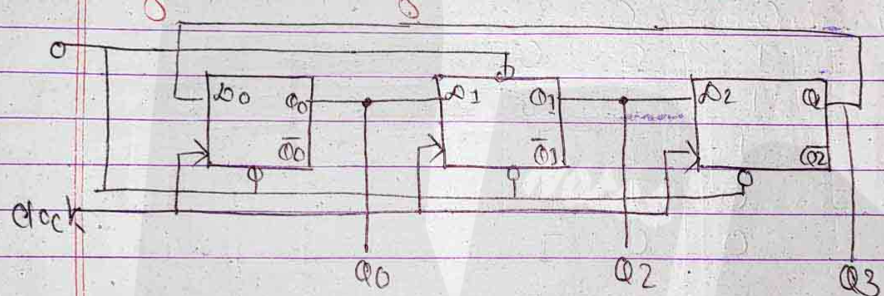
SIPO :-



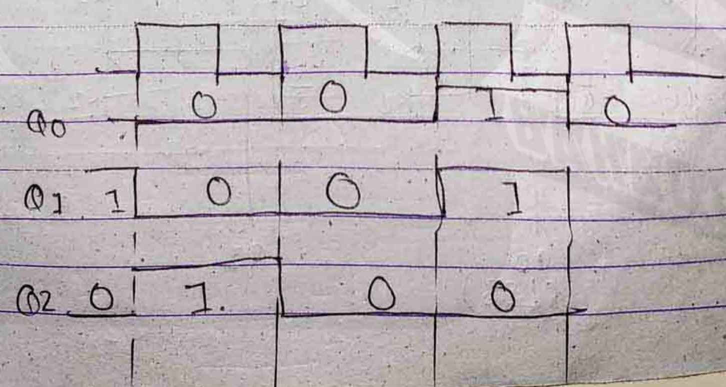
Clock	Q0	Q1	Q2
Initial	0	0	0
1st	1	0	0
2nd	0	1	0
3rd	1	0	1



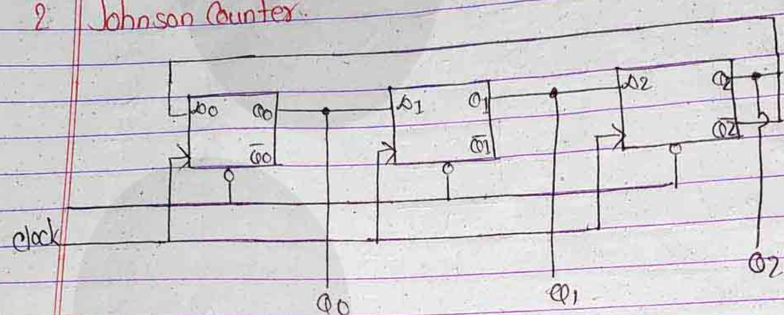
(1) Ring Counter (Straight Counter) (010)



Clock	Q_0	Q_1	Q_2
Initial	0	1	0
1st	0	0	1
2nd	1	0	0
3rd	0	1	0



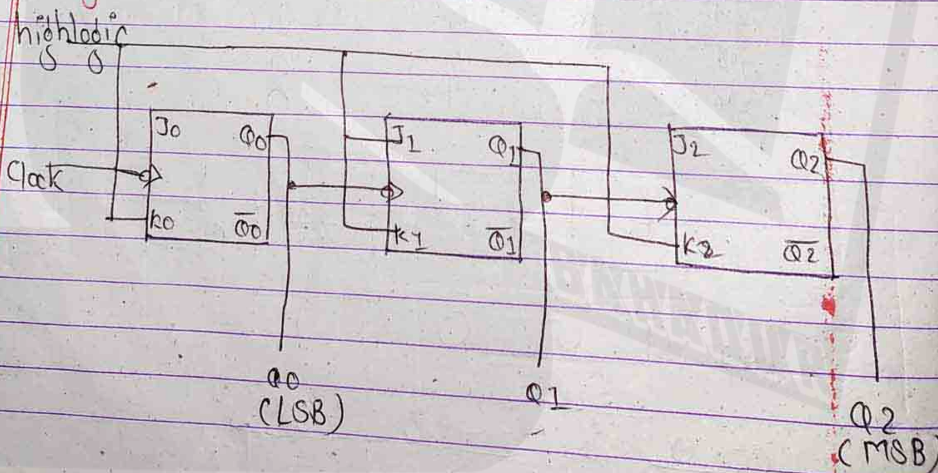
2. Johnson Counter.



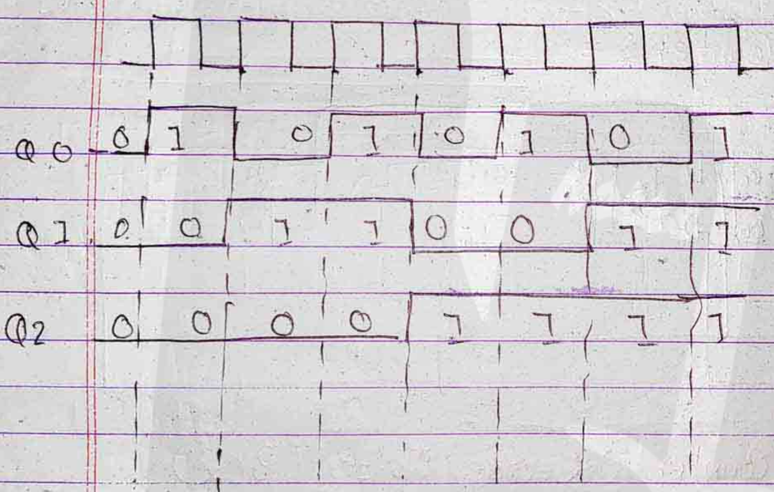
Pulse	Q0	Q1	Q2
Initial	0	0	0
1st	1	0	0
2nd	1	1	0
3rd	1	1	1
4th	0	1	1
5th	0	0	1

Counter :-

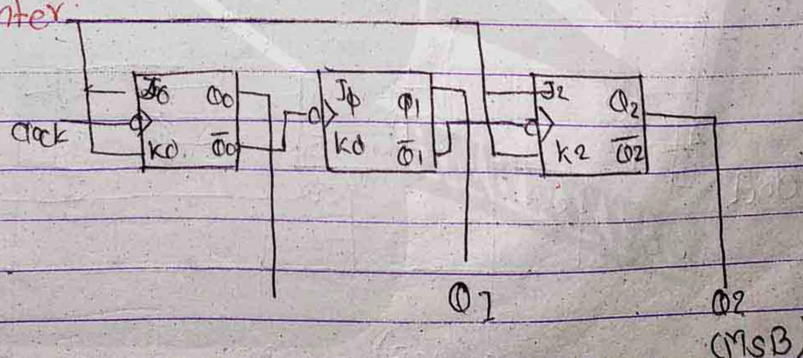
1. Asygnorous Counter :-



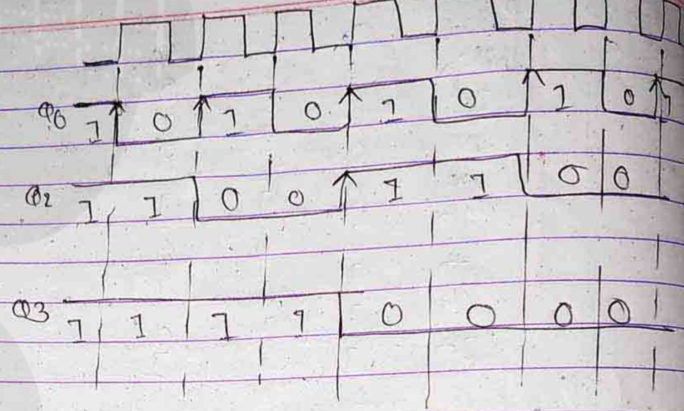
Pulse	Q2	Q1	Q0
Initial	0	0	0
1st	0	0	1
2nd	0	1	0
3rd	0	1	1
4th	1	0	0
5th	1	0	1
6th	1	1	0
7th	1	1	1



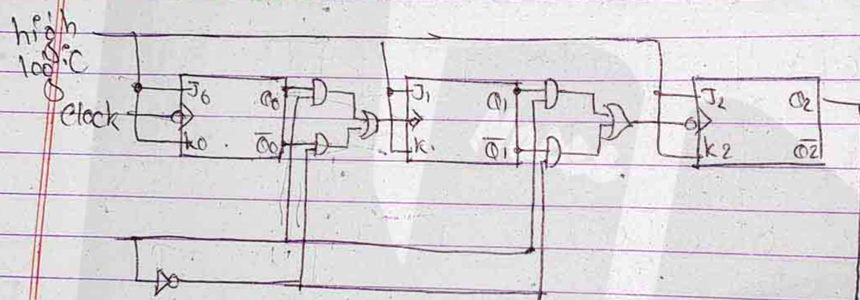
* down counter.



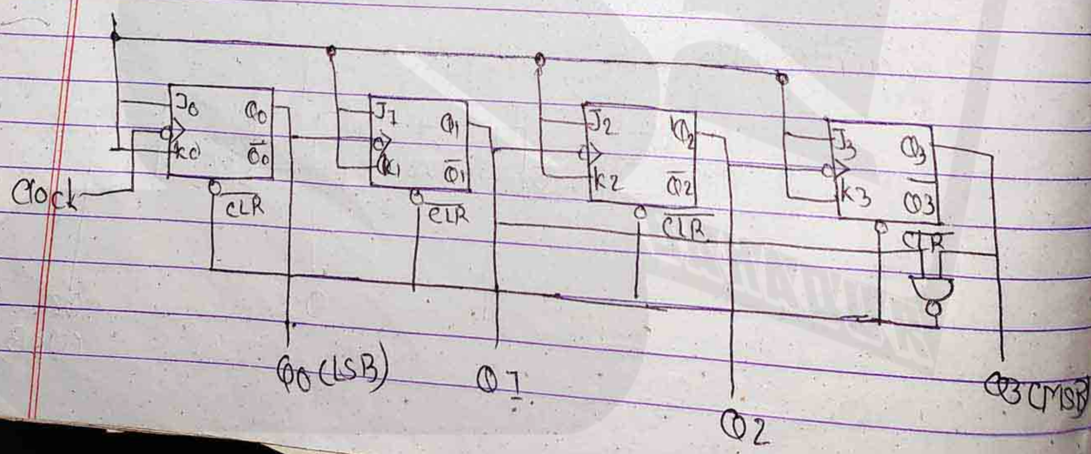
Q2	Q1	Q0
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0



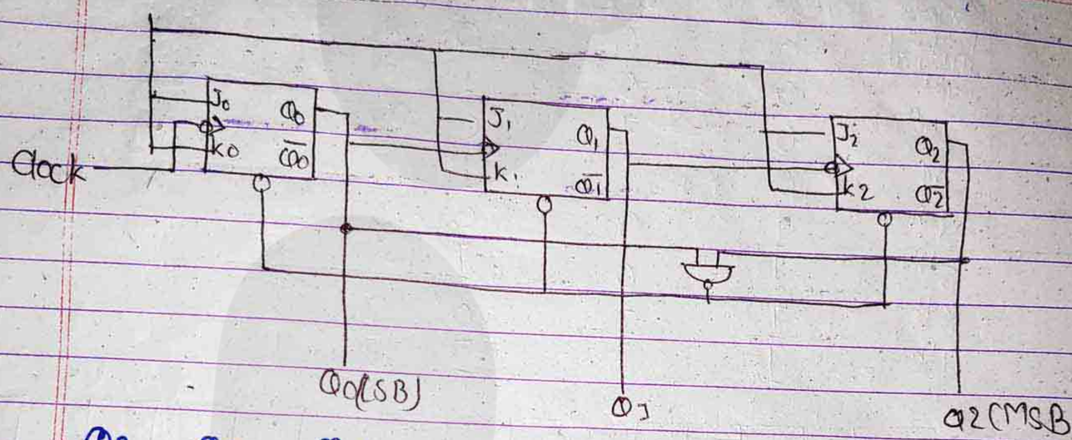
Up-down Counter :-



Modulus 10 Counter (Decade / Mod-10 / BCD)

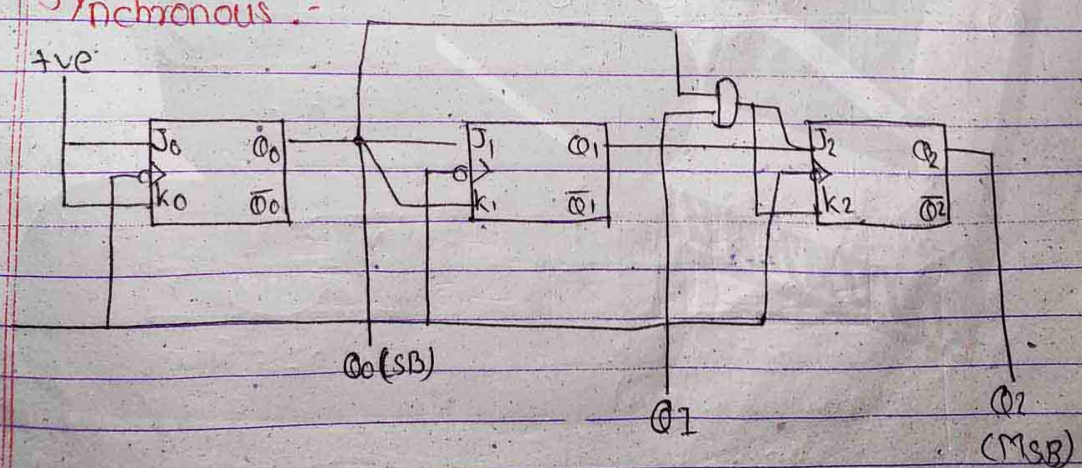


Mod-5

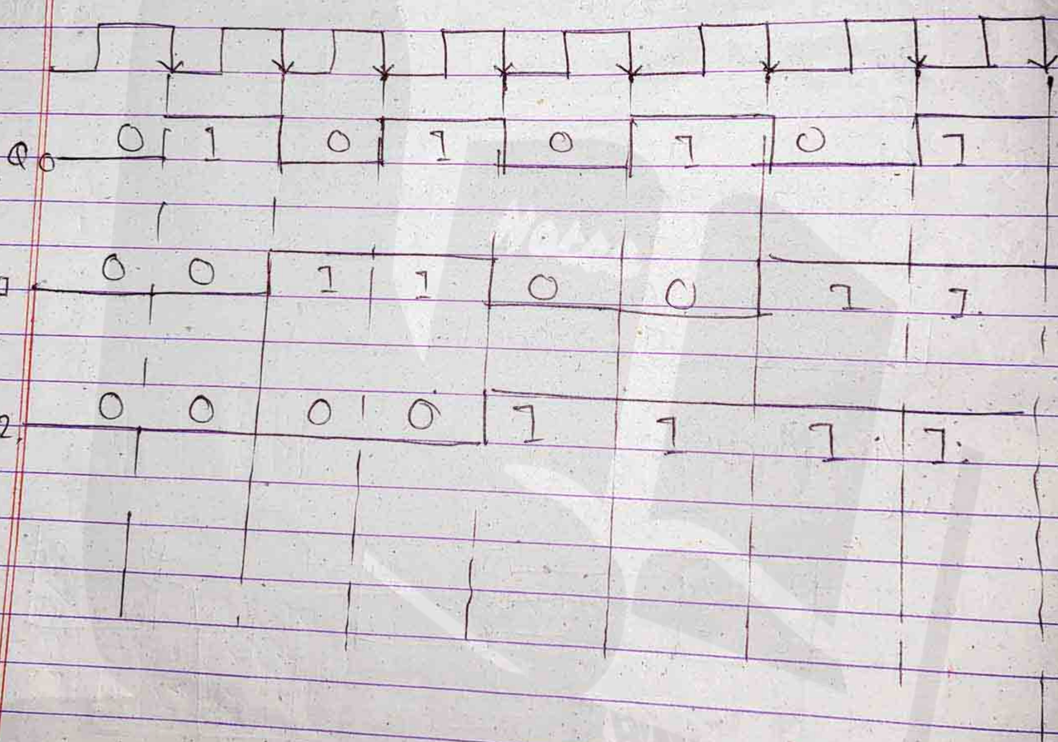


Q2	Q1	Q0	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	-> Reset

Synchronous :-

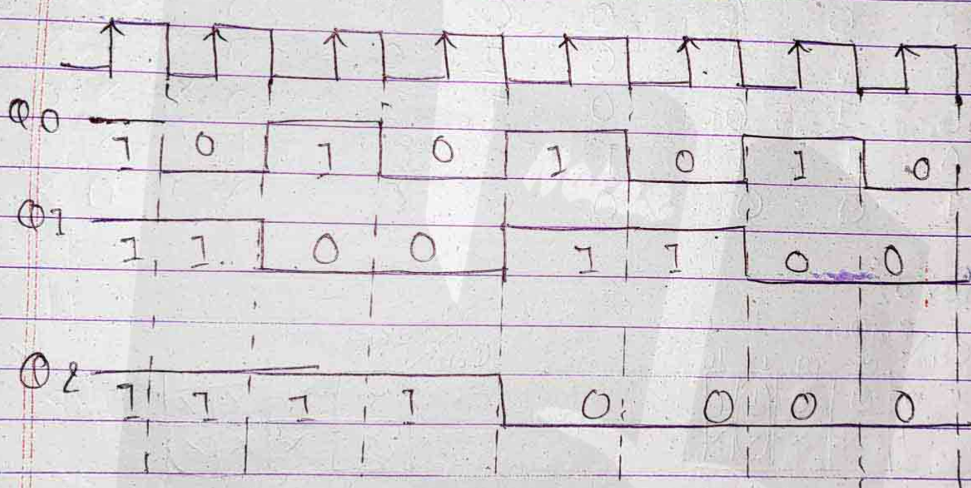


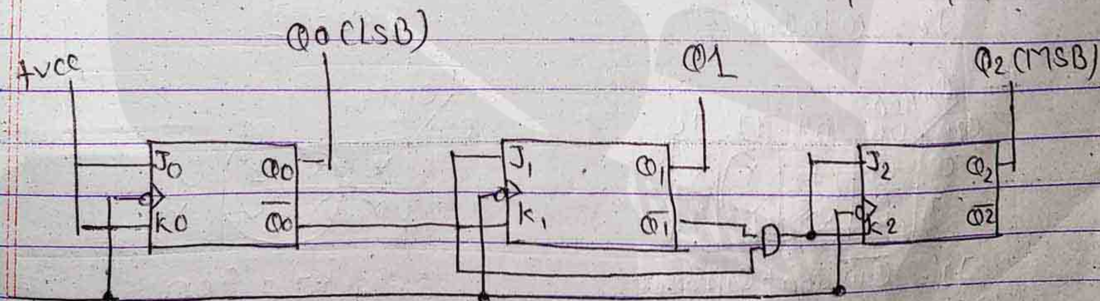
Clock	Q_2	Q_1	Q_0
Initial	0	0	0
1st	0	0	1
2nd	0	1	0
3rd	0	1	1
4th	1	0	0
5th	1	0	1
6th	1	1	0
7th	1	1	1



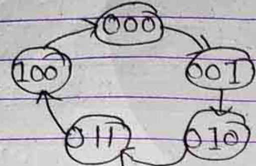
* Down Counter

Clock	Q_2	Q_1	Q_0
Initial	1	1	1
1^{st}	1	1	0
2^{nd}	1	0	1
3^{rd}	1	0	0
4^{th}	0	1	1
5^{th}	0	1	0
6^{th}	0	0	1
7^{th}	0	0	0





Synchronous mod-5 Counter :-



Execution table of J-K flip flop

Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present state			Next state			flip flop					
Q_{2n}	Q_{1n}	Q_{0n}	Q_{2n+1}	Q_{1n+1}	Q_{0n+1}	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	0	X	1	0	X	0	X

for J_2

Q_{2n} \ $Q_{1n}Q_{0n}$	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_2 = Q_{1n}Q_{0n}$

Q_{2n} \ $Q_{1n}Q_{0n}$				
	X	X	X	X
	1	X	X	X

$K_2 = 1$

Q_{2n} \ $Q_{1n}Q_{0n}$	00	01	11	10
0	0	1	X	X
1	0	X	X	X

$J_1 = \overline{Q_{1n}}Q_{0n} + Q_{1n}Q_{0n}$

$= Q_{0n}(\overline{Q}_{1n} + Q_{1n})$

$= Q_{0n}$

Q_{2n} \ $Q_{1n}Q_{0n}$				
	X	X	1	0
	X	X	X	X

$K_1 = Q_{0n}$

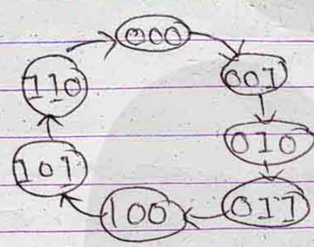
Q_{2n} \ $Q_{1n}Q_{0n}$				
	1	x	x	1
	0	x	x	x

$J_0 = \bar{Q}_{1n}\bar{Q}_{0n}\bar{Q}_{2n} + Q_{1n}\bar{Q}_{0n}\bar{Q}_{2n}$
$= \bar{Q}_{2n}$

Q_{2n} \ $Q_{1n}Q_{0n}$				
	x	1	1	x
	x	x	x	x

$k_0 = 1$

Synchronous mod-7 counter :-



Excitation table of J-k flip flop

Q_t	Q_{t+1}	J	k
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Present state			Next state			flip flop					
Q_{2n}	Q_{1n}	Q_{0n}	Q_{2n+1}	Q_{1n+1}	Q_{0n+1}	J_2	k_2	J_1	k_1	J_0	k_0
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	1	1	0	x	0	1	x	x	1
1	1	0	0	0	0	x	1	x	1	0	x

for J_2

Q_{2n} \ $Q_{1n}Q_{0n}$				
	0	0	1	0
	x	x	x	x

$J_2 = Q_{1n}Q_{0n}$

for k_2,

Q_{2n} \ $Q_{1n}Q_{0n}$				
	X	X	X	X
	1	X	X	X

$k_2 = 1$

Q_{2n} \ $Q_{1n}Q_{0n}$				
	0	1	x	x
	0	1	x	x

$J_1 =$

Q_{2n} \ $Q_{1n}Q_{0n}$				
	x	x	1	0
	x	x	x	1

$K_1 =$

Q_{2n} \ $Q_{1n}Q_{0n}$				
	1	x	x	1
	1	x	x	0

$J_0 =$

Q_{2n} \ $Q_{1n}Q_{0n}$				
	x	1	1	x
	x	1	0	x

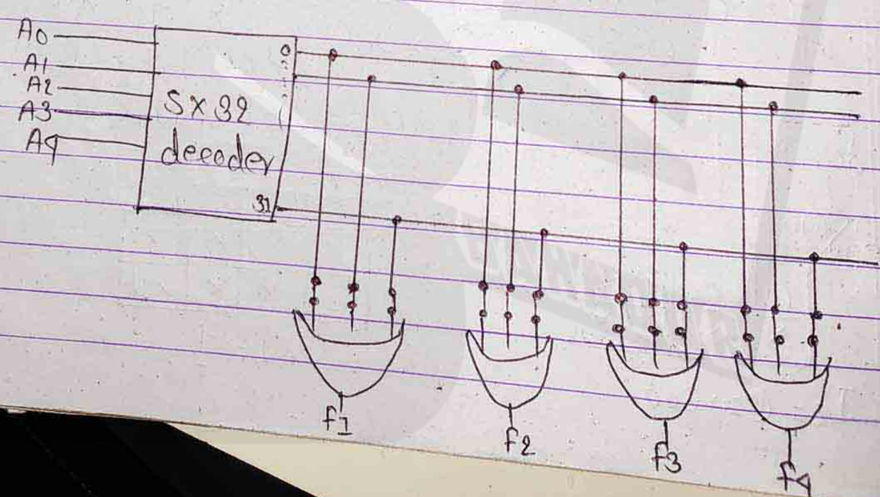
$K_0 =$

chapter-3

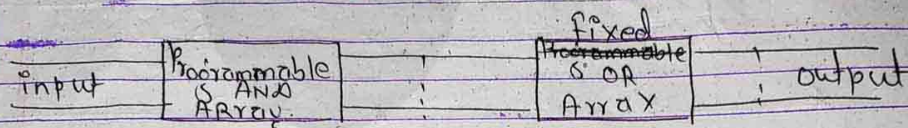
Basic concept of programmable logic :-

- → PROM
- → EPROM
- → PAL
- → PLA

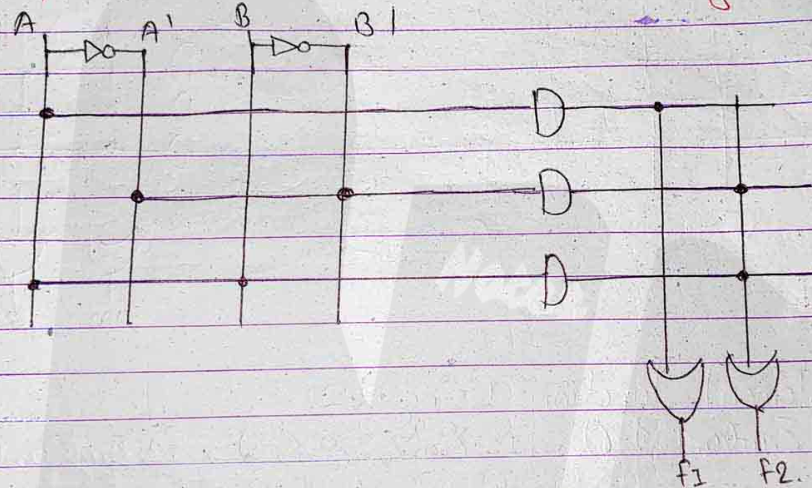
32 x 4 ROM



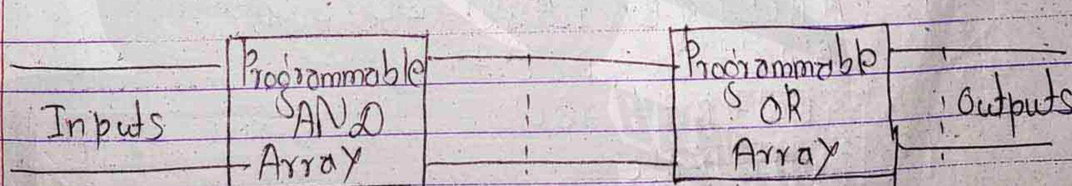
PAL (Programmable ~~logic~~ array Logic)

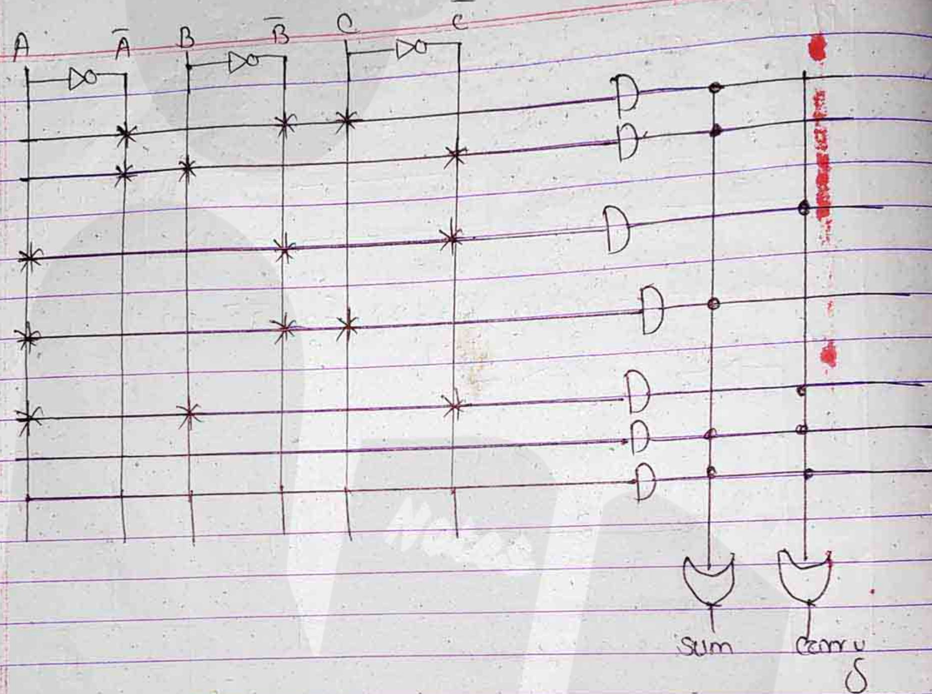


Implement function $f_1 = A_2$ $f_2 = \overline{A}\overline{B} + AB$ by using PAL



PLA (Programmable Logic array)





Implement function Using PLA.

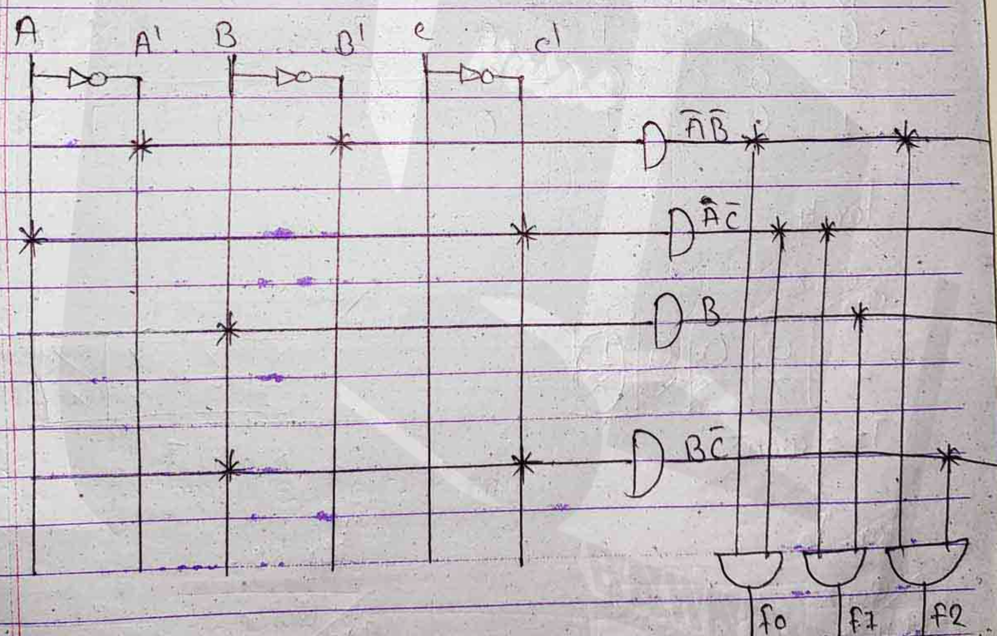
$f_0 = \Sigma m(0,1,4,6)$, $f_1 = \Sigma m(2,3,4,6,7)$, $f_2\ \Sigma m(0,1,26)$

$f_0 = A'B'C' + A'B'C + AB'C' + ABC'$
$\quad = A'B' + AC'$

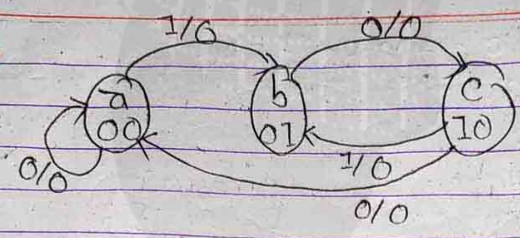
$f_1 = A'BC' + A'BC + AB'C' + ABC' + ABC$
$\quad = A'BC' + BC + AB'C' + ABC'$
$\quad = BC' + BC + ABC'$
$\quad = B + AB'C'$
$\quad = B + AC'$

$f_2 = A'B'C' + A'B'C + A'BC' + ABC'$
$= A'B' + BC'$

Product	Inputs A	B	C	Outputs f_0	f_1	f_2
$\bar{A}\bar{B}$	0	0	-	1	0	1
$A\bar{C}$	1	-	0	1	1	0
B	-	1	-	0	1	0
$B\bar{C}$	-	1	0	0	0	1



100



Present state Q_{1n}	Present state Q_{0n}	Input X	Nextstate Q_{1n+1}	Nextstate Q_{0n+1}	Output Y	J_0	K_0	J_1	K_1
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	1	X	0	X
0	1	0	1	0	0	X	1	1	X
0	1	1	0	1	0	X	0	0	X
1	0	0	0	0	1	0	X	X	1
1	0	1	0	1	0	1	X	X	1

for J_1

Q_{1n} \ $Q_{0n}X$				
0	0	0	0	1
1	X	X	X	X

$J_1 = Q_{0n}\overline{X}$

for K_1

Q_{1n} \ $Q_{0n}X$				
0	X	X	X	X
1	1	1	X	X

$K_1 = 1$

for J_0

Q_{1n} \ $Q_{0n}X$				
0	0	1	X	X
1	0	1	X	X

$J_0 = X$

for K_0

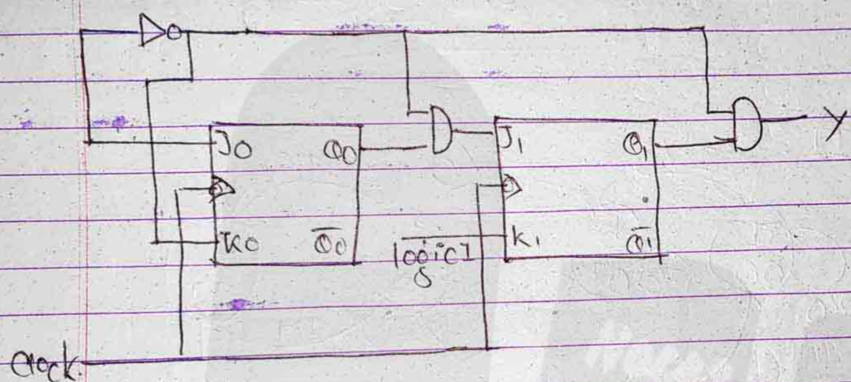
Q_{1n} \ $Q_{0n}X$				
0	X	X	0	1
1	X	X	X	X

$K_0 = \overline{X}$

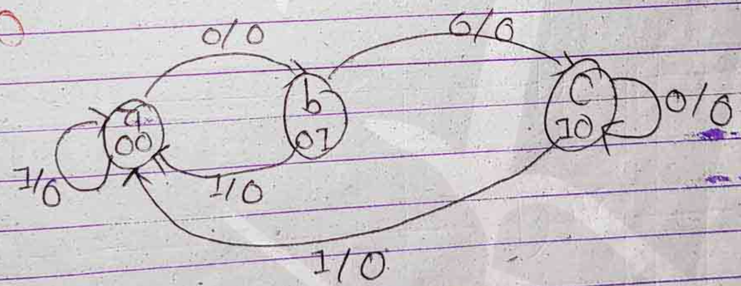
for Y

Q_{1n} \ $Q_{0n}X$			
Y	0	0	0
1	0	X	X

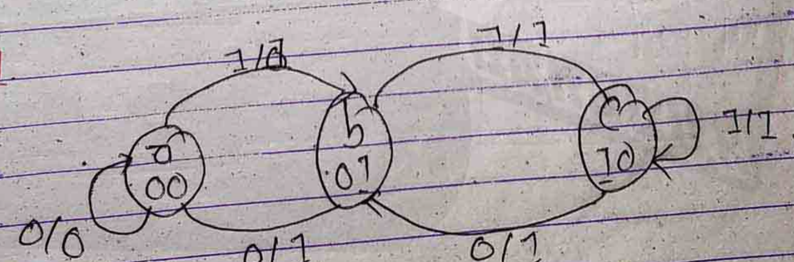
$Y = Q_{1n}\bar{X}$

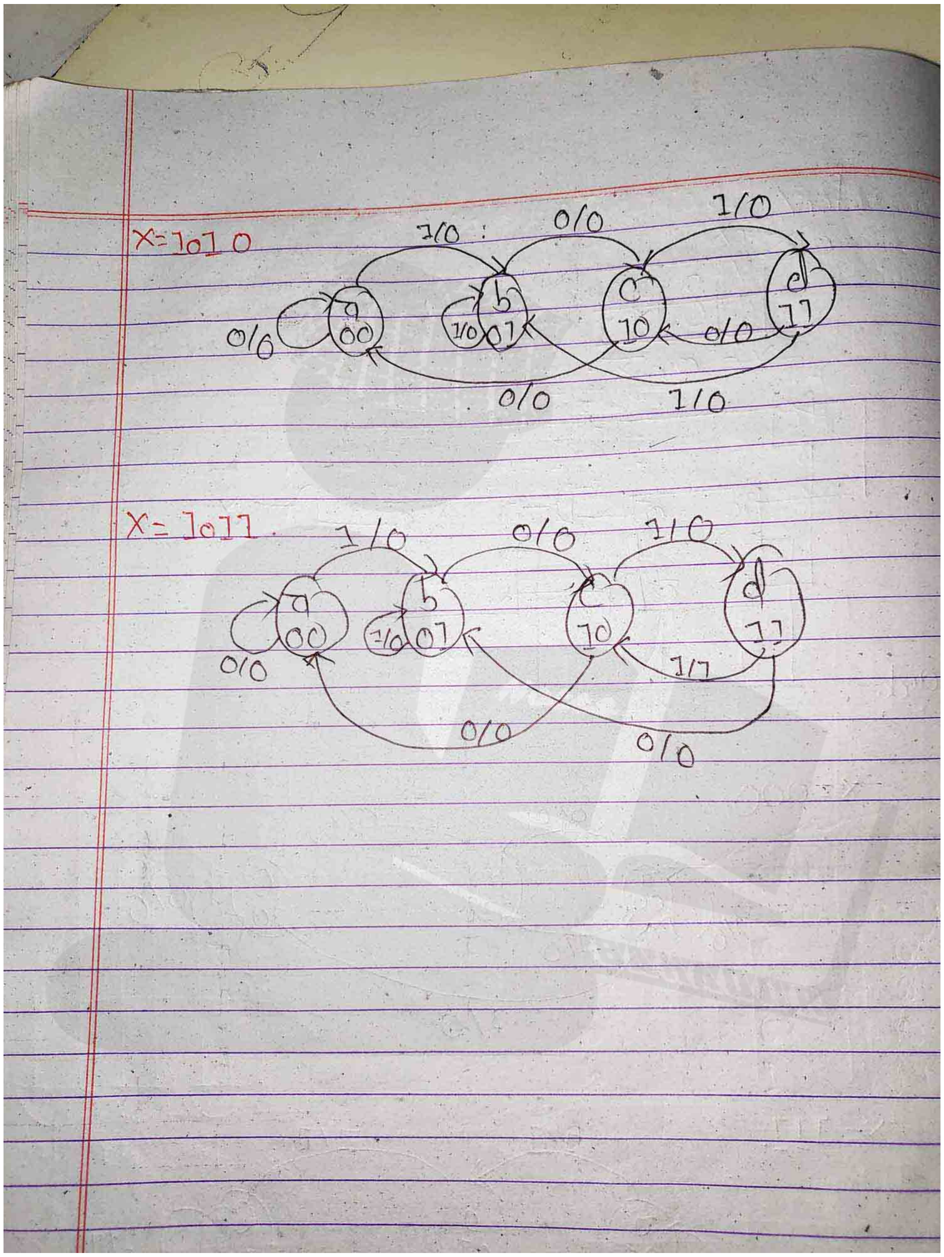


X = 000

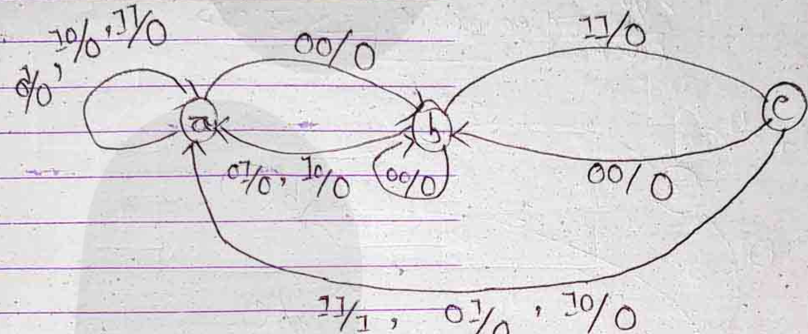


X = 111




X=1010
a
00
b
01
c
10
d
11
1/0
0/0
1/0
0/0
1/0
0/0
0/0
1/0
X= 1011
a
00
b
01
c
10
d
11
1/0
0/0
1/0
0/0
1/0
1/1
0/0
0/0

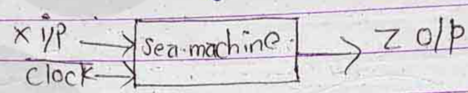
Design a machine has two several input x_1 & x_2 and one output 'Z'. The machine is required to give an output $Z = 1$ when both x_1 & x_2 contain the message '011'

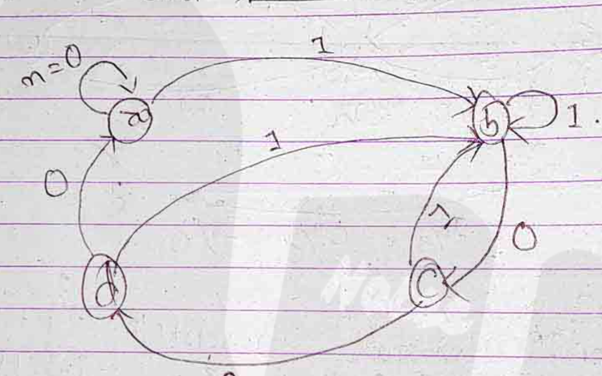


let a = 00, b = 01 and c = 10

Present state		Present IP		Next state		Present O/P	FF excitation			
Q_{1n}	Q_{0n}	x_1	x_2	Q_{1n+1}	Q_{0n+1}	Z	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	0	X	1	X
0	0	0	1	0	0	0	0	X	0	X
0	0	1	0	0	0	0	0	X	0	X
0	0	1	1	0	0	0	0	X	0	X
0	1	0	0	0	1	0	0	X	X	0
0	1	0	1	0	0	0	0	X	X	1
0	1	1	0	0	0	0	0	X	X	1
0	1	1	1	1	0	0	1	X	X	1
1	0	0	0	0	1	0	X	1	1	X
1	0	0	1	0	0	0	X	1	0	X
1	0	1	0	0	0	0	X	1	0	X
1	0	1	1	0	0	1	X	1	0	X

A synchronous machine has one bit serial input 'n' the output Z of a machine is to be set high when live input contains the message '100'. Draw the state diagram, derive the transition table or state table and design the circuit diagram.





Transition table.

Previous state	Next state		output	
	n = 0	n = 1	n = 0	n = 1
a	a	b	0	0
b	c	b	0	0
c	d	b	0	0
d	a	b	1	1

(P.T.O.)

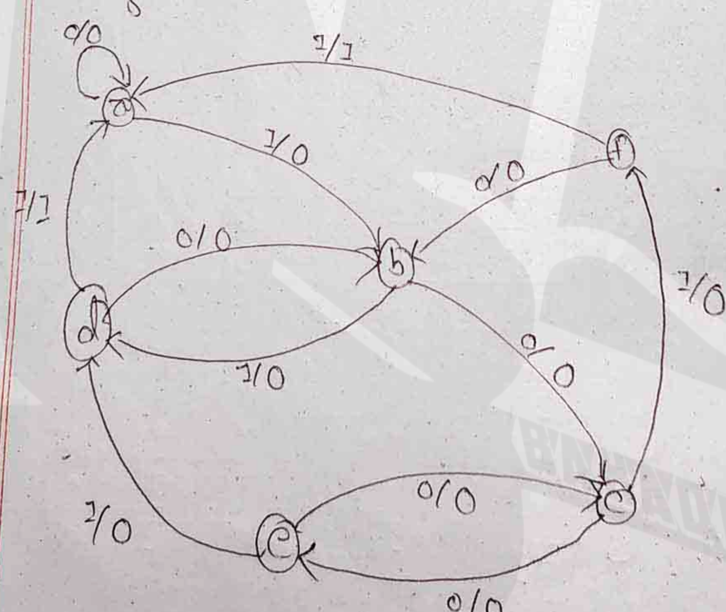
Present state		Present I/P	Nextstate		output	FF excitation			
Q_{1n}	Q_{0n}	x_i	Q_{1n+1}	Q_{0n+1}	Z	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	0	X	1	X
0	1	0	1	0	0	1	X	X	1
0	1	1	0	1	0	0	X	X	0
1	0	0	1	1	0	X	0	1	X
1	0	1	0	1	0	X	1	1	X
1	1	0	0	0	1	X	1	X	1
1	1	1	0	1	1	X	1	X	0

State reduction technique :-

- Raw elimination methods
- Implication table
- partitioning the state diagram.

Present state	Next state		Present o/p	
	n=0	n=1	n=0	n=1.
a	a	b	0	0
b	c	d	0	0
c	e	f	0	0
d	b	a	0	1
e	c	d	0	0
f	b	a	0	1.

(original table)



B and E are same, so 'e' raw is eliminated and subsitute e=b

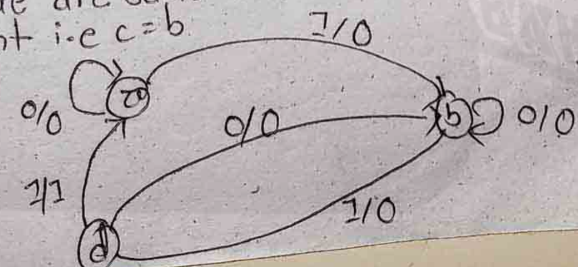
Present state	Next state		Present o/p	
	n=0	n=1	n=0	n=1
a	a	b	0	0
b	c	d	0	0
c	b	f	0	0
d	b	a	0	1
f	b	a	0	1

Here for n=0, b->c and c->b These two are equivalent if their next state are same. Here next state are same so they are equivalent i.e c=b.

Present state	Next state		Present o/p	
	n=0	n=1	n=0	n=1
a	a	b	0	0
b	c	d	0	0
c	b	d	0	0
d	b	a	0	1

Present state	Next state		Present o/p	
	n=0	n=1	n=0	n=1
a	a	b	0	0
b	b	d	0	0
d	b	a	0	1

Here for n=0, b->c and c->b. These two are equivalent if their next state are same. Here next state are same so they are equivalent i.e c=b



State reduction by Implication table method

Present state	Next state		present output
	x=0	x=1	
a	e	e	1
b	c	e	1
c	i	h	0
d	h	a	1
e	i	f	0
f	e	g	0
g	h	b	1
h	c	d	0
i	f	b	1

P.T.O

	a	b	c	d	e	f	g	h
b	c-e							
c	X	X						
d	~~e-h, e-a~~	~~c-h, e-a~~	X					
e	X	X	h-f	X				
f	X	X	~~i-e, h-g~~	X	~~i-e, f-g~~			
g	~~e-h, e-b~~	~~c-h, e-b~~	X	a-b	X	X		
h	X	X	~~i-e, h-d~~	X	~~i-d, f-d~~	~~e-c, g-d~~	X	
i	~~e-f, e-b~~	~~c-f, e-b~~	X	h-f, a-b	X	X	h-f	X

c ≡ e
h ≡ f
a ≡ b
d ≡ g

Present state	Next state x = 0	x = 1	Present O/P
a	c	c	1
c	d	f	0
d	f	a	1
f	c	d	0