

Digital Logic

M T W T F S S	
Page No.:	YOUVA
Date:	

* Number System

Page: - 2

- introduction ✓
- conversion ✓
- Arithmetic operation ✓
- Compliments ✓
- codes → BCD ✓
 - X-3 ✓
 - self complementing codes ✓
 - Grey code ✓
 - Alphanumeric codes ✓
- Representation of Numbers in computers → integer ✓
→ real no ✓

* Boolean Algebra page: - 38

- Introduction ✓
- Boolean law ✓
- Logic gate
- principle of duality
- 2 ways to represent Boolean functions → SOP
→ POS
- Kmaps.

* Combinational Circuits

- Introduction
- BCD to X-3 code conversion
- Half adder
- full adder
- BCD adder
- * - Look ahead carry adder
- Half subtractor
- full subtractor
- multiplexer (MUX)
- DEMUX
- encoder
- Decoder
- ROM

* Sequential Circuits

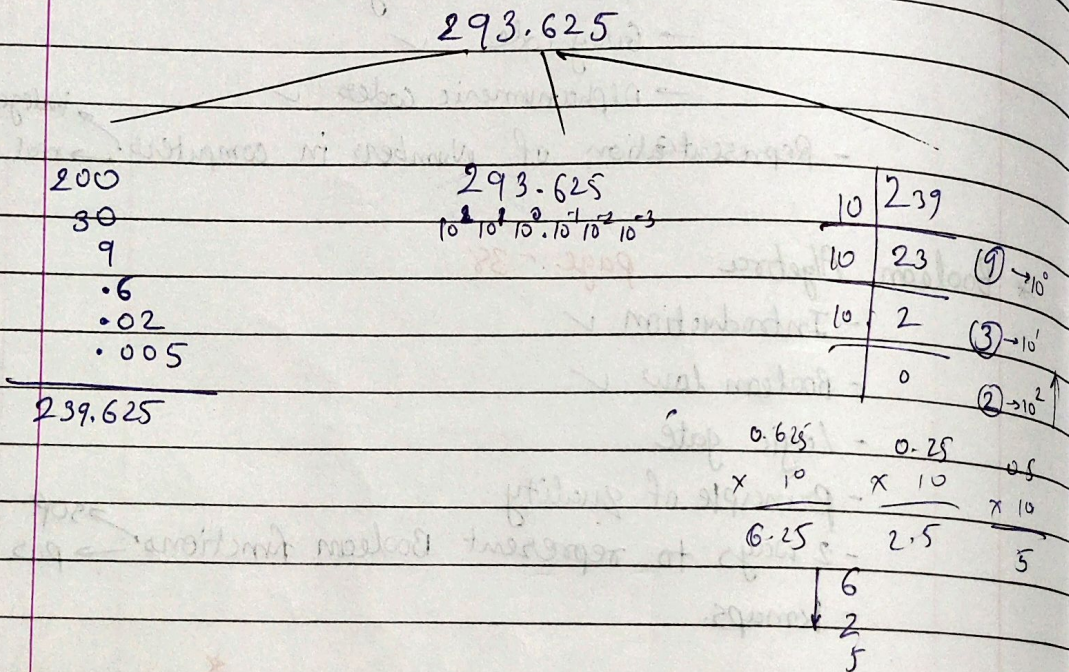
- Introduction
- Latch
- SR flip flop
- JK flip flop
- Master slave flip flop
- ~~combinational~~ Counter
- Registers
- Booth's algorithm for multiplication

① Number System

② Introduction

- 1st number system invented was unary Number system
- 2nd number system was decimal Number system

Decimal $\rightarrow$ Decimal number system is weighted code.



2 basic term

① Base :- Total no. of digit supported by system
 $\hookrightarrow n > 0$

② digit :- remainder generated when divided by base digit.

Base 2 $\rightarrow 0, 1$

8 $\rightarrow 0, 1, 2$

$n \rightarrow 0 \rightarrow n-1$

Base 11 $\rightarrow 0$ to 10

0, 1, ..., 9, (10) $\rightarrow A$

Base 16 $\rightarrow 0, \dots, 9, A, B, C, D, E, F$

Q7 234.52 base 9 ; 237.42
 $\downarrow \quad \downarrow$
 $5 \quad 7$
 base ≥ 6 base ≥ 8

Counting

0	0
1	1
(10)	10
(11) $\rightarrow$ max	9 $\rightarrow$ max
(100)	10
	99 $\rightarrow$ max
	(100)

Conversion

① Decimal to x. (base 10 to base x)

Q8 $(25.625)_{10} \rightarrow (\quad)_2$

$(11001.101)_2$

2	25	1	0.625
2	12	0	X 2
2	6	0	1.250
2	3	1	0.250
	1	1	X 2
			0.5
			X 2
			1.0

② $(36.5)_{10} \rightarrow (\quad)_3$

3	36	0
3	12	0
3	4	1
3	1	1
	0	

$(1100.1)_3$

	0.5
X 3	
1.5	

Binary number ke piche.
"one" add karne se

$$\text{num} = (\text{num} \times 2 + 1)$$

Q. $(2.6)_{10} \rightarrow ()_2$
 $(10.1001)_2$

$$\begin{array}{r} 0.6 \\ \times 2 \\ \hline 1.2 \\ \times 2 \\ \hline 0.4 \\ \times 2 \\ \hline 0.8 \\ \times 2 \\ \hline 1.6 \\ \times 2 \\ \hline 1.2 \end{array}$$

Q. $(189.75)_{10} \rightarrow ()_{32}$

$$\begin{array}{r|l} 32 & 189 \quad 29 \\ 32 & 5 \quad 5 \\ & 0 \end{array}$$

$$\begin{array}{r} 0.75 \\ \times 32 \\ \hline 24.0 \end{array}$$

$(5 \ 29.24)_{32}$

→ How many steps are needed to convert $(x)_{10}$ to $()_2$

$x = 2^n \rightarrow n \text{ steps}$

$3 \rightarrow 2 \text{ step}$

$5 \rightarrow 3 \text{ step}$

→ no. of bits

② Base x to decimal (Base x to Base 10)

$(11001.101)_2 \rightarrow ()_{10}$

$$\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 1 & . & 1 & 0 & 1 \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} & 2^{-3} \end{array}$$

$$\begin{aligned} & 1 + 0 + 0 + 8 + 16 + \frac{1}{2} + \frac{1}{4} \\ & 25 + (0.5 + 0.125) \\ & (25.625)_{10} \end{aligned}$$

Binary number ke piche "zero" add karne se num = (num) x 2

$$(1020)_3 \rightarrow (\quad)_{10}$$

$3^3 \ 3^2 \ 3^1 \ 3^0$

$$27 + 0 + 6 + 0 = (33)_{10}$$

Q)

$$(36)_x = (30)_{10} \text{ find } x$$

$$3x + 6 = 30$$

$$x = 8$$

Q)

$$(x1)_y = (36)_{10}$$

$$x < y$$

$$xy + y^0 = 36$$

$$xy = 35$$

$$x = 7 \rightarrow y = 5 \rightarrow \text{not valid}$$

$$x = 5 \rightarrow y = 7$$

$$x = 35 \rightarrow y = 1 \rightarrow \text{not valid}$$

$$x = 1 \rightarrow y = 35$$

Trick $(\quad)_x \rightarrow (\quad)_{10}$

8 4 2 1

$$112 \rightarrow$$

$$1100$$

$$5 = 101$$

$$1010 \rightarrow 10$$

$$10100 \rightarrow 20$$

$$101000 \rightarrow 40$$

left shift
 $\Rightarrow (num) \times 2$

$$101 \rightarrow 5$$

$$1011 \rightarrow 11$$

$$10111 \rightarrow 23$$

$$(num)_2 + 1$$

$$1011010$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

$$(((11 \times 2) + 1) \times 2)$$

$$45 \times 2$$

$$= 90$$

$$100 \rightarrow 5$$

$$100 \rightarrow 4$$

$$101 \rightarrow 5$$

$$110 \rightarrow 6$$

$$111 \rightarrow 7$$

$$1000 \rightarrow 8$$

$$1001 \rightarrow 9$$

$$1010 \rightarrow 10$$

$$1011 \rightarrow 11$$

$$1100 \rightarrow 12$$

$$1101 \rightarrow 13$$

$$1110 \rightarrow 14$$

$$1111 \rightarrow 15$$

① $()_x \rightarrow$ add 0 at last $\rightarrow \text{num} \times x$
 add 1 at last $\rightarrow (\text{num} \times x) + 1$
 add 2 at last $\rightarrow (\text{num} \times x) + 2$
 add 3 at last $\rightarrow (\text{num} \times x) + 3$

$$(127)_8 \Rightarrow (1270)_8$$

$\underbrace{\hspace{10em}}_{(\text{num} \times 8)}$

③ base x to base y $\{ (x)_x \rightarrow ()_y \}$

$$(12.13)_4 \rightarrow ()_5$$

$\underbrace{\hspace{10em}}_{()_{10}}$

$$2 \mid 2.13$$

$4^2 \ 4^1 \ 4^0 \quad 2^1 \ 4^2$

$$32 + 4 + 2 + 0.25 + 0.1875$$

$$(38.4375)_{10}$$

$$(38.4375)_{10} \rightarrow ()_5$$

$$123.20432$$

$$0.4375$$

$$\begin{array}{r} \times \quad 5 \\ \hline 2.1875 \\ \times \quad 5 \\ \hline 0.9375 \\ \times \quad 5 \\ \hline 4.6875 \end{array}$$

$$\begin{array}{r} 0.6875 \\ \times \quad 5 \\ \hline 3.4375 \end{array}$$

$$\begin{array}{r|l} 5 & 32 \\ \hline 5 & 7 \\ \hline 5 & 1 \\ \hline & 6 \end{array}$$

Q. $(11)_2 + (22)_3 + (33)_4 + (44)_5 = (abc)_6$

$(3)_{10} + 8 + 15 + 24 = \cancel{ab^2 + b^2 + c} (abc)_6$

$\phi(50) = \cancel{ab^2 + b^2 + c} (abc)_6$

$(122)_6 = (abc)_6$

$a=1, b=2, c=2$

6	50	2
6	78	2
6	1	1
	0	

Q. $(123)_5 = (xy)_y$ How many possible value of x & y ?

$(38)_{10} = xy + 8$ $(38)_{10} = (x, y)$

$x < y$

$xy \geq 30$

- 30, 1 ✗
- 1, 30 ✗
- 5, 6 ✓
- 6, 5 ✗
- 10, 3 ✗
- 3, 10 ✓
- 15, 2 ✗
- 2, 15 ✓

4 values

Q. $(43)_x = (y3)_8$ possible value of x & y

$4x + 3 = 8y + 3$
 $x = 2y$

$x > 4$

$y < 8$

- $y = 7 \rightarrow x = 14$
- $y = 6 \rightarrow x = 12$
- $y = 5 \rightarrow x = 10$
- $y = 4 \rightarrow x = 8$
- $y = 3 \rightarrow x = 6$

5 value possible

Q. > $\sqrt{(121)}_2 = (.)_2$

$\sqrt{(9^2 + 2 \cdot 9 + 1)}_{10}$

$\sqrt{(9+1)^2}_{10} = (9+1)_{10} = (11)_2$

Que > Roots of $x^2 - 11x + 22 = 0$ are 3 & 6 in which base?



base $\rightarrow b$

$x^2 - (b+1)x + (2b+2) = 0$

Sum of roots = $b+1$

$b+1 = 3+6$

$b = 8$

Que > $(\frac{66}{6})_6 = (11)_6$ is true in which base

$\frac{(66)_6}{(6)_6} = (11)_6 \Rightarrow \frac{6b+6}{6} = b+1$

$(b+1) = b+1$

~~base~~ in all cases base ≥ 7

Que >

In number system ranges of numbers are from -3 to 3, represented as c, D, A, 0, 1, 2, 3. Represent 18, 26 & 102 in this number system.

Base = 7

7	18
7	3
	0

-3 $\rightarrow$ c
 3 $\rightarrow$ 3

$(18)_{10} = (\frac{36}{2})_7$ here

Remember suggested by

$$\begin{array}{l} 2 \rightarrow 8 \\ 2 \rightarrow 16 \end{array}$$

$3 \rightarrow 9$

$$\begin{array}{c|c} 7 & 26 \\ \hline 7 & 4 \end{array} \quad -2 \rightarrow B$$

$$\begin{array}{c|c} 7 & 1 \\ \hline & 0 \end{array} \quad -3 \rightarrow C \quad (1CB)$$

$$1 \nearrow$$

$$\begin{array}{r} 7 \overline{) 102} \\ 7 \overline{) 15} \quad -3 \rightarrow C \\ \underline{7 \overline{) 2}} \quad 1 \\ \underline{ 6} \quad 2 \end{array}$$

→ r_L to r_K conversion

$2 \rightarrow 8 \Rightarrow 2 \rightarrow 2(3) \rightarrow \text{slice}$
 $2 \rightarrow 16 \Rightarrow 2 \rightarrow 2(4) \rightarrow \text{slice}$

$$\rightarrow (10110.10111)_2 \rightarrow ()_8$$

$$\begin{array}{r} 10110.1011 \\ \hline \end{array}$$

$$\downarrow \downarrow \downarrow \downarrow$$

$$(26.56)_8$$

→ 10111010.1001101]
01010.1188010
(572.0466)₈

→ 10110111000010 . 100111011 ₂ → () ₁₆

5BC2.9D8

→ $(122 \cdot 2(1))_3 \rightarrow ()_9$

$$0122 \cdot 2110$$

(18.73)9

→ $2^k \rightarrow 2$ conversion

$$(\quad)_8 \rightarrow (\quad)_2$$

(143) $\rightarrow$ (23714.12)

every 1 ke 3 bit

$$(1001111001100.00101)_2$$

→

$$(23612.121)_8 \rightarrow (\quad)$$

01001110001010.001010001

$$(63284.123)_9 \rightarrow (\quad)_3$$

$$20 \overset{10}{0} 022211.010210$$

Q) $(728)_9 = (\quad)_{27}$

$(\cancel{20})_3$

$(210222)_3 \rightarrow (2126)_{27}$

(3)

210 222

$$(7\ 8\ 1\ 2\ 3\ 4\ 6)_9 = (\)_{27}$$

$$(21\ 22\ 01\ 02\ 10\ 11\ 20)_3 \rightarrow (\)_{27}$$

$$(7 \cdot 2^4\ 11\ 10\ 15)_{27}$$

Q7) $(24\ 26\ 22)_{27} \rightarrow (\)_9$

$$K=3 \rightarrow (\)_3$$

$$(220\ 222\ 211)_3 \rightarrow (\)_9$$

$$(2\ 6\ 8\ 8\ 4)_{27}$$

Q7) $(7\ 6\ 5\ 1\ 2\ 3\ 4)_8 \rightarrow (\)_{16}$

$$K=3 \rightarrow (\)_2$$

$$(11110\ 101\ 001\ 010\ 011\ 100)_2 \rightarrow (\)_{16}$$

$$(1F\ 529C)_{16}$$

(GATE)

How many 1's in binary

$$(5 \times 512) + (3 \times 64) + (7 \times 8) + 2$$

eq. of this No.

$$(5372)_8 \rightarrow (\)_2$$

$$(101\ 011\ 111\ 010)_2$$

4

Tricks for Conversion

① $(base)^x = 1$ followed by x 0's in that.

$$(2)^3 = 8 = (1000)_2$$

$$(3)^3 = 27 = (1000)_3$$

② Continuous 1's in any base.

$$\frac{b^K - 1}{b - 1}$$

$$\rightarrow (11111)_2 \xrightarrow{\substack{2^4+2^3+2^2+2^1+2^0 \\ 2^5-1}} \frac{2^5-1}{2-1} = 31$$

$$\text{G.P.} \rightarrow \begin{matrix} r^0 + r^1 + r^2 + r^3 + r^4 + \dots \\ a \quad a(r^K-1) \\ r-1 \end{matrix}$$

$$Q \rightarrow (255)_{10} \rightarrow ()_2$$

8's 1's
 2^8-1

$$\rightarrow (11111)_3 \rightarrow \frac{3^5-1}{3-2} = 121$$

$$3^4 + 3^3 + 3^2 + 3^1 + 3^0$$

③ Number is slightly less than perfect square

$$(254)_{10} \rightarrow ()_2$$

$$\begin{aligned} & 255 \rightarrow (11111111)_2 \\ & \text{1 kam he} \rightarrow (11111110)_2 \end{aligned}$$

$$\begin{aligned} & (253)_{10} \rightarrow ()_2 \\ & \text{2 kam he} \rightarrow (255)_{10} \rightarrow (11111111)_2 \\ & \rightarrow (11111101)_2 \end{aligned}$$

$$(252)_{10} \rightarrow ()_2$$

$$\begin{aligned} & \text{3 kam he} \\ & \rightarrow (11111100)_2 \end{aligned}$$

$$(500)_{10} \rightarrow ()_2$$

$(511)_2 = \begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$
 11 km $\downarrow$

11 km ↓

(500)

11110100

(14) Number is slightly greater than perfect square

$$(33)_{10} \rightarrow (\quad)_2$$

✓ (32) $\rightarrow (100000)_2$

1 Jyada ↓

$$(33) \rightarrow (100001)_2$$
$$(66)_D \rightarrow (\quad)_2$$

10:00 P.M.

$$(70)_{10} \rightarrow ()_2$$

64 → 1000 000

6 syfda $\rightarrow$ 482

$$(70)_{10} \Rightarrow 1000110$$

(5) add 0 $\rightarrow$ num $\times 2$

$$\text{add } 1 \rightarrow (\text{num} + 2) + 1$$

if all 1's = $2^{10} = 1024$

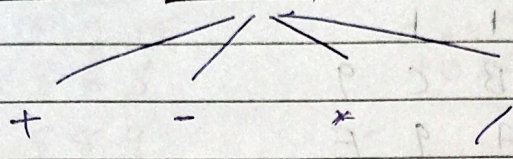
→ $((1111111101))$

⑦ Master trick

$$(150)_{10} \rightarrow (2)_{2}$$

100
12
1100/100

Arithmetic Operations.



10's place 1's place

$$\begin{array}{r} 15 \\ + 9 \\ \hline 24 \end{array}$$

① Addition

① Base 8 :-

$$\begin{array}{r} 13 \\ + 47 \\ \hline \end{array}$$

$$\begin{array}{r} 10 \\ 9 \\ \hline 7 \end{array}$$

Carry: 10 → 2, 9 → 3

$$(6 \ 2)_8$$

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \ 1 \\ 1 \ 2 \ 3 \ 6 \ 7 \ 4 \\ + 7 \ 6 \ 5 \ 6 \ 7 \ 7 \\ \hline 1 \ 1 \ 1 \ 1 \ 5 \ 7 \ 3 \end{array}$$

Carry: 1, 4, 7, 8

$$(1111573)_8$$

$$\begin{array}{r} 1 \ 1 \ 1 \\ 1 \ 1 \ 2 \ 3 \ 4 \\ 2 \ 1 \ 5 \ 6 \ 7 \\ \hline (33023)_8 \end{array}$$

② Base 12

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \ 1 \\ 1 \ A \ B \ 9 \ 7 \ 8 \\ 3 \ B \ A \ 5 \ 6 \ 9 \\ \hline (5 \ A \ A \ 3 \ 2 \ 5)_{12} \end{array}$$

A → 10, B → 11

$$\begin{array}{r} 12 \overline{) 17} \\ \underline{12} \\ 5 \ 8 \\ \underline{12} \\ 9 \\ \underline{12} \\ 17 \end{array}$$

Base: -16 addition

$$\begin{array}{r} 1 1 1 1 1 \\ 1 A F B C 9 \\ 5 9 F A 9 F \\ \hline \end{array}$$

$$(74F668)_{16}$$

→ $38 + 43 = 80$ base 9,9

$$\begin{array}{r} 8 \\ + 3 \\ \hline (11) \end{array} \quad \begin{array}{l} \text{only when} \\ \text{base} = 11 \end{array}$$

→ $(38)_x + (43)_x \neq (80)_x$

$$3x + 8 + 4x + 3 = 8x + 0$$

$$\boxed{x = 11}$$

→ $4 + 4 = 14$ base = 9,9

$$4x + 4 = x + 4$$

$$x = 4$$

not possible

$$4 + 4 = 8$$

always possible for
 $\text{base} \geq 9$

(2) Subtraction

$$\begin{array}{r} 1215 \\ - 589 \\ \hline 1076 \end{array}$$

$$\begin{array}{r} 0.99910 \\ - 10000 \\ \hline 0.9999 \end{array}$$

$$\begin{array}{r} 1011110 \\ - 1000000 \\ \hline 99999 \end{array}$$

Base :- 8

$$\begin{array}{r} 1310 \\ - 757 \\ \hline (457)_8 \end{array}$$

$$\begin{array}{r} 1A308 \\ - 8DCFF \\ \hline (11D306)_{16} \end{array}$$

Base :- 2

$$\begin{array}{r} 100112 \\ - 111111 \\ \hline (0111001)_2 \end{array}$$

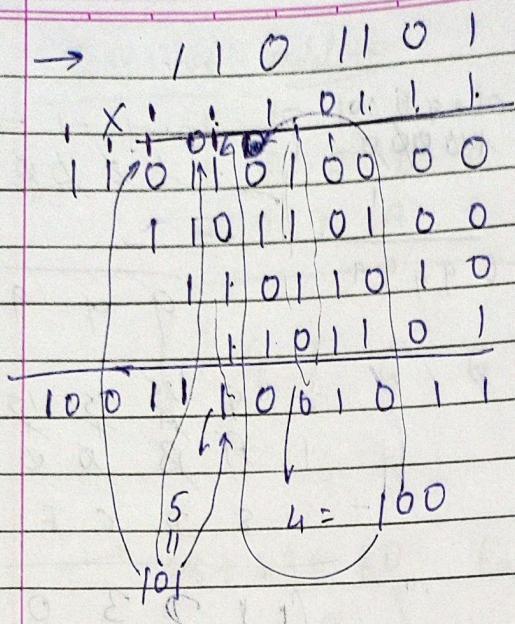
(3) multiplication

Base 8

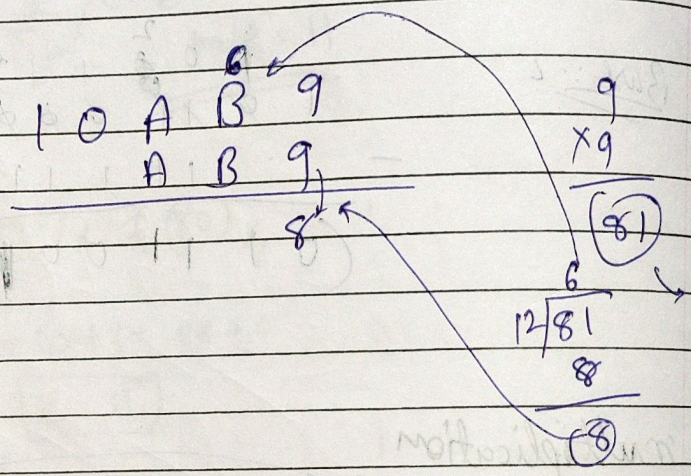
$$\begin{array}{r} 47 \\ \times 74 \\ \hline 4210 \\ 2304 \\ \hline (4444)_8 \end{array}$$

$$\begin{array}{r} 7 \\ \times 4 \\ \hline 28 \end{array} \rightarrow 4$$

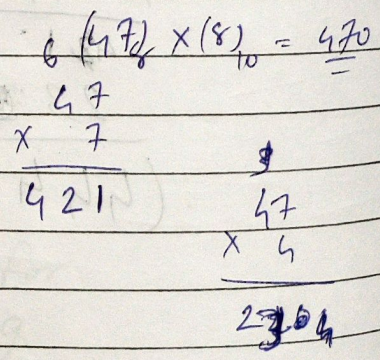
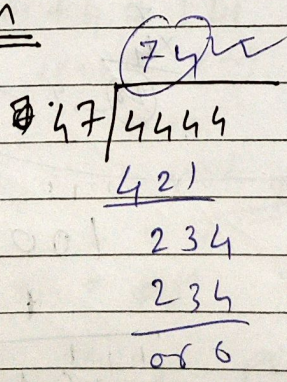
$$\begin{array}{r} 100100 \\ + 1111 \\ \hline 10010000 \\ + 10010000 \\ + 10010000 \\ + 100100 \\ \hline (100001100)_2 \end{array}$$



Base - 12



④ Division



$(11101)_2 \div (1000)_2 \Rightarrow \underline{\underline{11.101}}$

$$\begin{array}{r}
 1011001001 \\
 \hline
 1000 \overline{) 101101} \\
 \underline{1000} \\
 001110 \\
 \underline{1000} \\
 1101 \\
 \underline{1000} \\
 101 \\
 \underline{100} \\
 0010 \\
 \underline{0000} \\
 01000 \\
 \underline{1000} \\
 1000 \\
 \underline{1000} \\
 000
 \end{array}$$

$$\begin{array}{r}
 1011.101 \\
 \hline
 1000 \overline{) 101110} \\
 \underline{1000} \\
 1110 \\
 \underline{1000} \\
 1101 \\
 \underline{1000} \\
 1010 \\
 \underline{1000} \\
 1000 \\
 \underline{1000} \\
 0
 \end{array}$$

else zero
 uper

2
 nie do

Compliments for Base r :

r 's
 $r^n - (N)_r$
 no. of digits
 before point

$(r-1)$'s
 $r^n - (N)_r - r^{-m}$
 number of
 digits after
 point

Base $\rightarrow 10$ (9's complement = 10's complement $\leftarrow$ last
 digit meso 1 minus

$$\begin{array}{r}
 (123.36)_{10} \\
 \swarrow \text{10's} \\
 \searrow \text{9's} \\
 10^3 - 123.36 = 876.64
 \end{array}$$

$$\begin{aligned}
 & 876.66 - 10^{-2} \\
 & = 876.63
 \end{aligned}$$

Subtract the number from 10
add one at last digit $\rightarrow$ 10's comp.

* $(1236.432)_{10}$

all 9's - 10

9's comp =

9999.999

- 1236.432

8763.567

10's comp

8763.568

Base: -8

$(1236.467)_{10}$

8's

9's

7777.777

- 1236.467

6541.310

+1 at last

6541.311

Base: -16

$(1ABC.BDF)_{16}$

15's

16's

FFFF.FFF

- 1ABC.BDF

E543.420

add 1

E543.421

Q find 7's complement of 5555.555

2 possibility of (5555.555)

base can be $\rightarrow$ 8 or 7.

7's

$$\begin{array}{r} 7777.777 \\ - 5555.555 \\ \hline 2222.222 \end{array}$$

8's

$$\begin{array}{r} 6666.666 \\ - 5555.555 \\ \hline 1111.111 \end{array}$$

Base:-2

$$101111.1001$$

$\rightarrow$
7's comp.

$$010000.0110$$

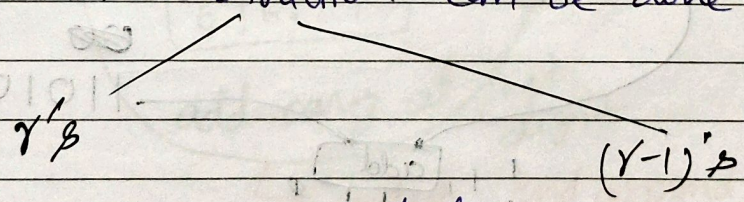
$\rightarrow$
2's comp.

$\Rightarrow$ Subtraction using complements

$$010000.0111$$

$\Rightarrow$ Why compliments?

$\rightarrow$ Subtraction can be done by addition.



Base 10:-

1236 $\rightarrow$ minuee

0065 $\rightarrow$ subtrahend

A - (B)

$\rightarrow$ 10's comp.

Take to 10's comp.

8 add

$$\begin{array}{r} 10000 \\ 1236 \\ - 0065 \\ \hline 9935 \end{array}$$

$$\begin{array}{r} 1236 \\ + 9935 \\ \hline \end{array}$$

$$\begin{array}{r} 11171 \end{array}$$

Ans

$$\begin{array}{r} 65 \\ - 1236 \\ \hline \end{array}$$

$$\begin{array}{r} 65 \\ + 8764 \\ \hline 8829 \end{array}$$

$$\begin{array}{r} 10000 \\ - 1236 \\ \hline 8764 \end{array}$$

if carry not generated

→ ans is (-)

& 10's comp of that is Ans

$$-(10000 - 8829) = -1171$$

Base:-2

$$\begin{array}{r} 101101.11 \\ 1011.1 \end{array}$$

$$\begin{array}{r} 101101.11 \\ 001011.10 \end{array} \xrightarrow{2's \text{ comp.}}$$

$$110100.10$$

$$\begin{array}{r} 101101.11 \\ + 110100.10 \\ \hline 100010.01 \end{array}$$

carry →

Ans

$$\begin{array}{r} 01011.10 \\ 101101.11 \end{array} \xrightarrow{2's \text{ comp.}}$$

$$010010.01$$

no carry here.

$$\begin{array}{r} 001011.10 \\ + 010010.01 \\ \hline 011101.11 \end{array}$$

$$\xrightarrow{2's \text{ comp.}}$$

$$100010.01$$

⇒ Subtraction using (r-1)'s complement.

Base: 10

$$\begin{array}{r} 639.42 \\ - 23.15 \\ \hline \end{array}$$

↓

Step → ①

$$\begin{array}{r} 639.42 \\ - 023.15 \\ \hline \end{array} \xrightarrow{9's \text{ comp}} 976.84$$

add

$$\begin{array}{r} 639.42 \\ + 976.84 \\ \hline \end{array}$$

$$\begin{array}{r} 639.42 \\ + 999.99 \\ - 023.15 \\ \hline 1616.26 \\ - 1000.00 \\ \hline 616.26 \\ + 0.01 \\ \hline 616.27 \end{array}$$

carry

$$\begin{array}{r} 1616.26 \\ + 1 \text{ at last digit} \\ \hline 1616.27 \end{array}$$

add carry at last digit

↳ called end around carry

$$\begin{array}{r} 023.15 \\ - 639.42 \\ \hline \end{array} \xrightarrow{9's \text{ comp}} 360.57$$

no carry

$$\begin{array}{r} 023.15 \\ + 360.57 \\ \hline 383.72 \end{array}$$

$$\begin{array}{r} 023.15 \\ + 999.57 \\ - 639.42 \\ \hline 383.72 \\ - 999.99 \\ \hline \end{array} \xrightarrow{\text{add}} -616.27$$

Ans -ve

& 9's comp is ans

Base:-2

$$\begin{array}{r} 101.11 \\ - 10.1 \\ \hline \end{array}$$

$$\begin{array}{r} 101.11 \\ - 010.10 \\ \hline 101.01 \end{array} \xrightarrow{\text{1's comp}}$$

$$\begin{array}{r} 101.11 \\ + 101.01 \\ \hline 1011.00 \end{array}$$

carry ← 1011.00

add here

$$\boxed{011.01}$$

$$\begin{array}{r} 10.1 \\ - 101.11 \\ \hline \end{array}$$

$$\begin{array}{r} 010.10 \\ - 101.11 \\ \hline \end{array} \xrightarrow{\text{1's comp}}$$

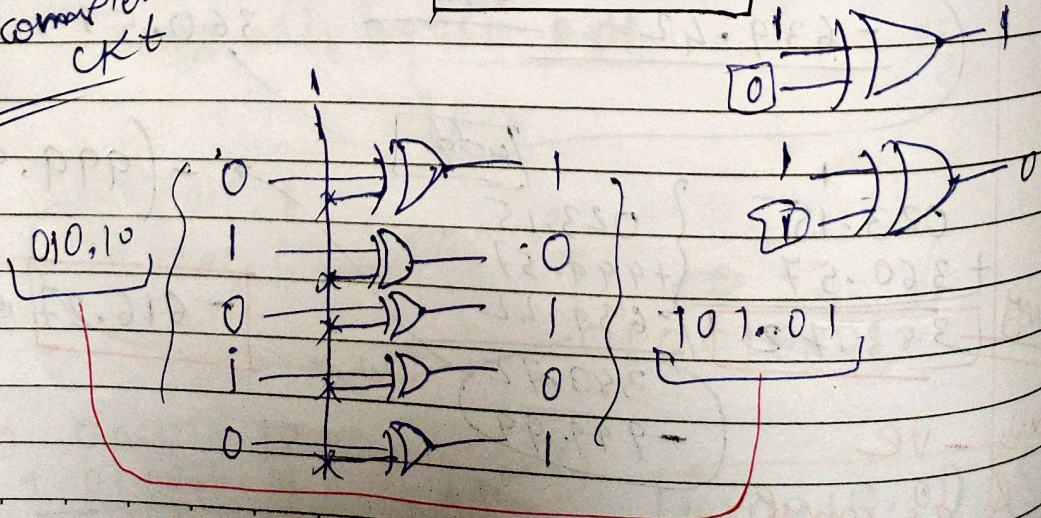
$$\begin{array}{r} 010.10 \\ + 010.00 \\ \hline 100.10 \end{array}$$

no carry.

$$\text{Ans} = -(011.01)$$

$$\boxed{= -(011.01)}$$

1's complement
ckt



Base:-8

$$\begin{array}{r}
 1267.13 \\
 - 0326.54 \quad \xrightarrow{7's \text{ comp.}} \quad 7451.23 \\
 \hline
 \text{add} \\
 \hline
 1267.13 \\
 + 7451.23 \\
 \hline
 10740.36 \\
 \xrightarrow{\text{Carry}} \quad \text{Ans} = \boxed{740.37}
 \end{array}$$

Codes

① B.C.D $\rightarrow$ weighted code

$\rightarrow$ Binary Coded decimal

$\rightarrow$ for each decimal digit

$\rightarrow$ 4 bit binary

8 bit $\rightarrow$ 1 byte

4 bit = Nibble

$\rightarrow ()_{10} \rightarrow \text{BCD}$

$$\rightarrow (123.4)_{10} \xrightarrow{\text{BCD}} 0001\ 0010\ 0011.0100$$

1 = 0001

2 = 0010

3 = 0011

4 = 0100

$(100100011.01)_{\text{BCD}}$

$\rightarrow (496.123)_{10}$

$\xrightarrow{\text{BCD}} 0100\ 1001\ 0110.0001\ 0010\ 0011$

~~00000000~~
 $(496.123)_{10} \rightarrow (10010010110.000100100011)_{\text{BCD}}$

→ $(936.42)_{10}$

↳ $(100100110110.01000010)_{BCD}$

→ BCD → $()_{10}$

1100100010101.100100101

$(39157.928)_{10}$

$(\underbrace{1011}_{11}.\underbrace{1100}_{12}.\underbrace{1010}_{10})_{BCD} \rightarrow ()_{10}$

decimal cannot support

Invalid BCD → Total 6 invalid.

→ BCD addition

↳ If carry generated or result is invalid BCD then add decimal 6 to the result.

①
$$\begin{array}{r} 0010 \\ + 0110 \\ \hline 1000 \end{array}$$
 no carry

②
$$\begin{array}{r} 1000 \\ + 0111 \\ \hline 1111 \end{array}$$
 no carry

invalid BCD

∴ add 6

$$\begin{array}{r} 1111 \\ + 0110 \rightarrow 6 \\ \hline 10101 \end{array}$$

$$\begin{array}{r} 10101 \\ \hline 15 \end{array}$$

③
$$\begin{array}{r} 1000 \\ + 1000 \\ \hline 10000 \end{array}$$
 carry → add 6

$$\begin{array}{r} 10000 \\ \hline 10110 \\ \hline 16 \end{array}$$

(4)

$$\begin{array}{r}
 1000 \\
 0111 \\
 \hline
 1111 \\
 0110 \\
 \hline
 10100 \\
 (16)
 \end{array}
 \quad
 \begin{array}{r}
 0110 \\
 1000 \\
 \hline
 1110 \\
 0110 \\
 \hline
 0100 \\
 (4)
 \end{array}
 \rightarrow
 \begin{array}{r}
 86 \\
 78 \\
 \hline
 164
 \end{array}$$

Auxiliary carry

invalid $\rightarrow$ 0101 $\rightarrow$ 1000 $\rightarrow$ 58

invalid $\rightarrow$ 1000 $\rightarrow$ 1000 $\rightarrow$ 48

invalid $\rightarrow$ 1000 $\rightarrow$ 0000 $\rightarrow$ (106)

0110 0110

10000 0110

(106)₁₀

1011 1100 1010 $\rightarrow$ invalid BCD

+ 0110 0110 0110

10010.0011.0000

(123)₁₀

Q $\rightarrow$ 1111 1011 1110

+ 0110 0110 0110

101100010.0100

(162.4)₁₀

0100 0101 $\rightarrow$ 45

0101 0101 $\rightarrow$ 55

invalid $\rightarrow$ 1001 1010 $\rightarrow$ (100)

0110

invalid $\rightarrow$ 1000 0000

0110

10000 0000

(106)₁₀

→ BCD Subtraction

⇒ Subtraction using 9's complement

- i) take 9's comp of subtrahend and add to min
 - ii) if carry is generated or we get invalid BCD add "6" to the result.
 - iii) ~~The~~ final carry generated → add to the least place
- case 1
- ~~Since~~ → positive

→ If result is valid BCD then result is positive and 9's comp is Ans

⇒ Subtraction using 10's complement

- i) take 10's comp and add
- ii) if carry is generated on invalid BCD add 6 to ~~the~~ it.
- iii) if final carry is generated then remaining is Ans

→ if valid BCD then negative of (10's comp) of that number is Ans

② Excess-3 code (X-3 code)

$$(BCD + 3) = X-3 \text{ code}$$

$(123.5)_{10}$

BCD :- 0000 0010 0011 . 0101

add 3 0011 0011 0011 . 0011

0100 0101 0110 . 1000

1 2 3 . 5
 +3 +3
 4 5 6 . 8

invalid X-3

0, 1, 2, 13, 14, 15

0000 0001 0010 1101 1110 1111
 X-3

0 $\xrightarrow{+3}$ 0000 0011

1 $\xrightarrow{+3}$ 0100

2 $\xrightarrow{+3}$ 0101

3 $\xrightarrow{+3}$ 0110

4 $\xrightarrow{+3}$ 0111

5 $\xrightarrow{+3}$ 1000

6 $\xrightarrow{+3}$ 1001

7 $\xrightarrow{+3}$ 1010

8 $\xrightarrow{+3}$ 1011

9 $\xrightarrow{+3}$ 1100

self complementing
 for Base 10

q's comp

1's complement of
 coded number gives q's
 complement of number
 itself

2 $\xrightarrow{+3}$ 0101
 q's comp
 1's comp

7 $\xrightarrow{+3}$ 1010

③ Self complementing codes

weighted code

non-weighted codes

→ X-3 code

self complementing

X-3 code is self complementing for base

→ is comp. of coded number

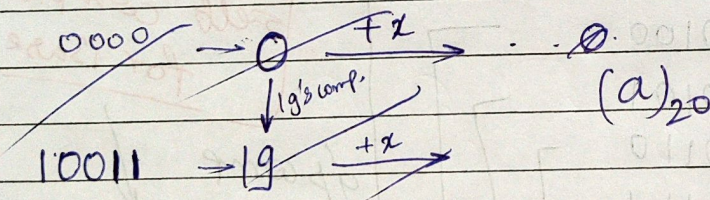
is g's comp of number it self

→ for Base 10

1's complement of coded number gives
 $(r-1)$'s complement of number it self

Ques which even code is self complementing for base 20?

Ex even → 5



$$\begin{aligned} &\text{for base } \rightarrow 20 \\ &(a)_{20} \xrightarrow{\times} (a+20)_2 \\ &\quad \downarrow \text{1's comp.} \\ &(19-a)_{20} \rightarrow \frac{(2^d-1)-(a+20)_2}{2} \end{aligned}$$

$$(19-a + k)_2 = (2^d-1)-(a+20)$$

for base → r

$$((r-1)-a) + k = (2^d-1) - (a+k)$$

$$k = 2^d - a$$

add all the weights $\rightarrow 9 \rightarrow$ self com for Base 10
 $\rightarrow 9 \rightarrow$ not

only for odd even

M	T	F	S	S	31
Page No.:					YOUVA
Date:					

for base $\rightarrow 20$

$$d = 5$$

$$K = \frac{2^d - 1}{2}$$

$$(20-1) - d + K = 2^d - 1 - d - K$$

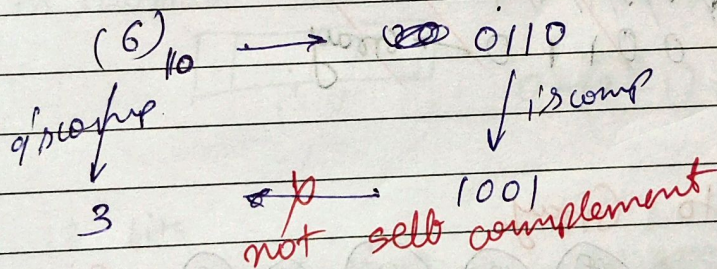
$$19 + K = 2^5 - 1 - K$$

$$2K = 31 - 19$$

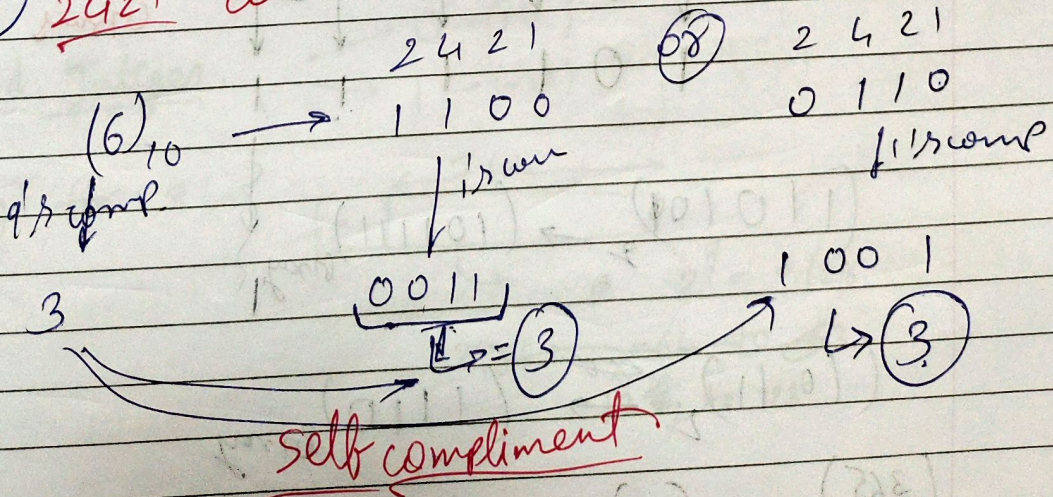
$$K = \frac{31-19}{2} \Rightarrow \boxed{K=6}$$

Weighted self complementing codes

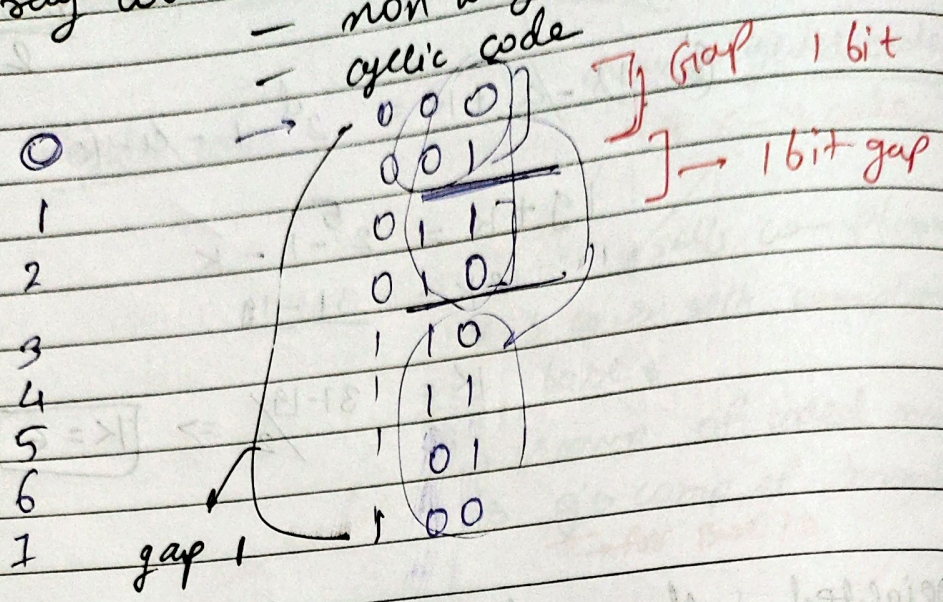
① 8421 is S.C. for Base 10??



② 2421 code



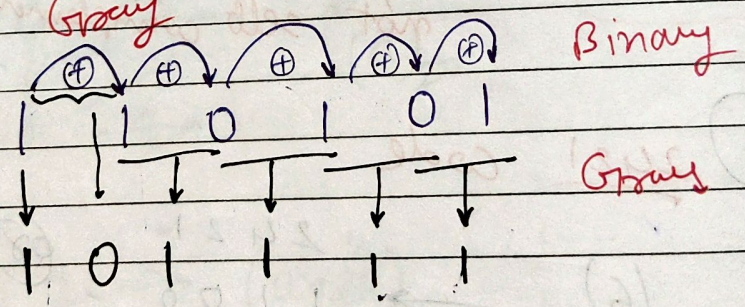
④ Gray Code :- ~~Reflec~~ Reflection Code.
 - non weighted code
 - cyclic code



(Binary to gray) $\rightarrow$ XOR

$$\begin{array}{ccccccc} 1 & 1 & 0 & 1 & 0 & 1 & \rightarrow \text{Binary} \\ \oplus & \oplus & & \oplus & & \oplus & \\ 1 & 0 & 0 & 1 & 1 & 0 & \rightarrow \text{Gray} \end{array}$$

Binary to Gray



$(110101)_2 \rightarrow (101111)_{\text{Gray}}$

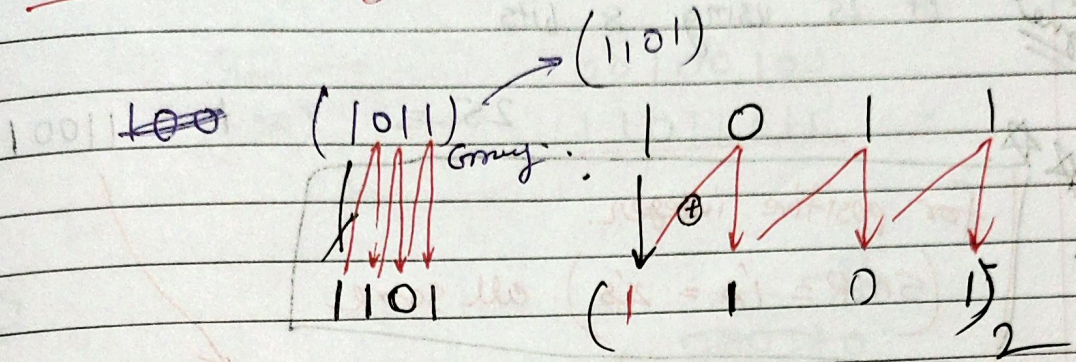
$(1011)_2 \rightarrow (1110)_{\text{Binary}}$

$(365)_{10} \rightarrow ()_{\text{Gray}}$

$$(011\ 110\ 101)_2 \rightarrow (01000\ 1111)_{\text{Gray}}$$

Binary to Gray $\rightarrow$
 Gray to Binary $\rightarrow$

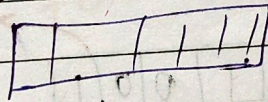
Gray to Binary



① Representation of Numbers in Computer

① Unsigned Integer

Que > If 8 bits are needed to store unsigned integer in computer then find the range.

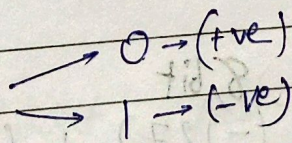


$$0 - (2^8 - 1) \Rightarrow (0 - 255)$$

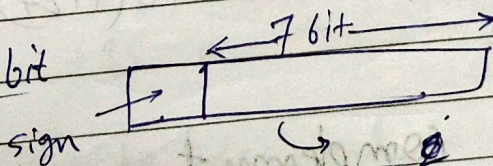
for 12 bits

$$0 - (2^{12} - 1) \Rightarrow (0 - 4095)$$

② Signed Integer



range for 8 bit



$$2^7 = 128$$

$$\text{range} = -128 \text{ to } 127$$

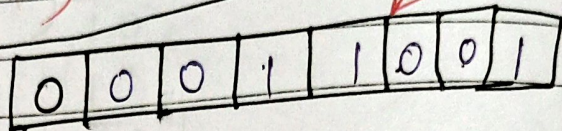
for positive is ~~is~~ : irrespective of the just convert into b

Positive signed

Give SMR, i's and 2's complement of 25 using 8 bits

$$25 = 011001$$

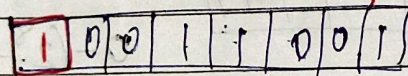
for positive integer.
(SMR = i's = 2's) all same



negative signed number

→ Here SMR, i's comp, 2's comp representation for (-ve) number will be different.

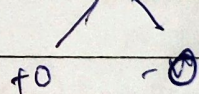
① -25 in SMR 8 bit.



→ SMR

SMR disadvantage

→ Two representation for zero



Range for 8 bit

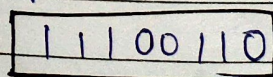
→ (-127) to (+127)

② -25 in i's complement.

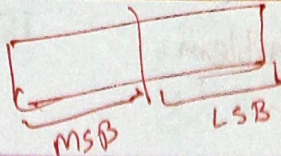
$$+25 \rightarrow 00011001$$

i's comp

(-25) in i's comp



16 bit



-36 using 1's comp

36 → 00100100
 (-36) → 11011011

+25

00011001

-25

1's comp 00011001 → 11100110

16 bit ↓

00000000 00011001

1's comp 16 bit

11111111 11100110

MSB → all zero added

+0 = -0 problem

① Find 1's complement of 25 in 8 bits

25 → 00011001
 1's comp 11100110

② Find 1's complement represent of 25 in 8 bit

① → +25 → 00011001

1's represent to decimal

01000100

negative

11110101

1's comp

ans = -11

∴ +ve

68

00001010

11

No +0, -0 problem here

③ -25 in 2's complement representation,

$$+25 = 00011001$$

↓ 2's complement

$$-25 \Rightarrow 11100111$$

⇒ Smallest ve number using 2's complement

c's comp

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

smallest

$$10000000 = -128$$

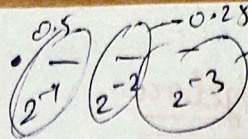
2's comp largest

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

$$= 127$$

	SMR	1's	2's
1111111	-127	-0	-1
10000000	-0	-127	-128
0111111	+127	+127	+127
00000000	0	0	0

$$0.5 \rightarrow 0.1$$

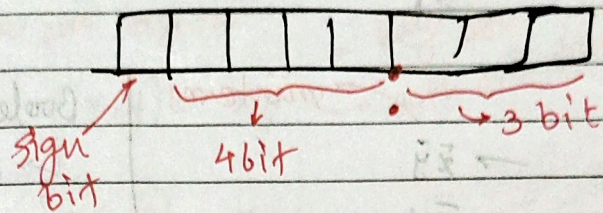


$$0.125$$

M T W T F S S						
Page No.:				YOUVA		
Date:						

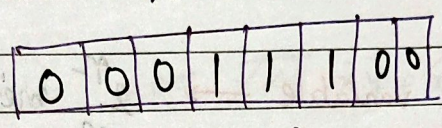
Fixed Number

8 bit



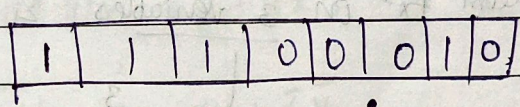
① $3.5 \rightarrow$

$$11.1$$

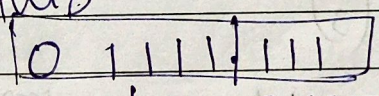


② $-12.25 \rightarrow$

$$1100.01$$

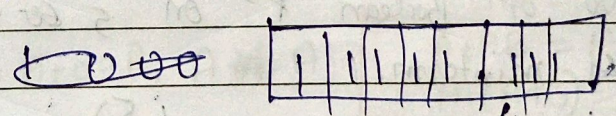


largest for this



$$\rightarrow 15.875$$

Smallest negative Number



$$\rightarrow -15.875$$

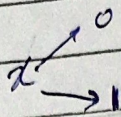
range -15.875 to 15.875

$\rightarrow$ not all represented in b/w.

Floating representation

$\rightarrow$ refer COA NOTES

② Boolean Algebra



x	y	minterm	Boolean function
0	0	$\bar{x}\bar{y}$	→ different combination of minterm.
0	1	$\bar{x}y$	
1	0	$x\bar{y}$	
1	1	xy	

n boolean variable $\rightarrow 2^n$ minterms
 $\rightarrow 2^n$ Boolean fⁿ

Que) No. of boolean fⁿ on 3 variables & having exactly 3 minterm

$\rightarrow 2^3 \rightarrow$ Total minterm

$$\binom{8}{3}$$

atmost 3 minterms :- $\binom{8}{0} + \binom{8}{1} + \binom{8}{2} + \binom{8}{3}$

Que) No. of Boolean fⁿ on 5 variables & having exactly 4 minterm.

$$\binom{25}{4}$$

2 variable $\xrightarrow{2^2}$ 16 $f(x)^n$

only 7 are useful.

AND :- $f(x, y) = xy$
 both are true then AND is true

OR :- $f(x, y) = x + y \rightarrow \bar{x}y + \bar{y}x + xy$
 any 1 true $\rightarrow$ OR is true

NOT :- $f(x) = \bar{x}$

NAND :- $f(x, y) = \overline{xy} = \bar{x}\bar{y} + \bar{x}y + x\bar{y}$

NOR :- $f(x, y) = \overline{x+y} = \bar{x}\bar{y}$

XOR :- $f(x, y) = \bar{x}y + \bar{y}x$

XNOR :- $f(x, y) = xy + \bar{x}\bar{y}$

Boolean laws

1) Idempotent law: $A + A = A$
 $A \cdot A = A$

2) Identity law: $A + 0 = A$, $A \cdot 1 = A$

3) Domination law: $A + 1 = 1$, $A \cdot 0 = 0$

4) Commutative law: $A + B = B + A$, $A \cdot B = B \cdot A$

5) Distributive law: $A \cdot (B + C) = (A \cdot B) + A \cdot C$
 $A + BC = (A + B)(A + C)$

6) Demorgan's law: $\overline{(A \cdot B)} = \bar{A} + \bar{B}$
 $\overline{(A + B)} = \bar{A} \cdot \bar{B}$

7) Absorption law

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

$$A + AB = A$$

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

8) Unit property

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

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

9) zero property

$$A \cdot 0 = 0$$

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

$$\cancel{x+y} \cdot x(x+y) = x$$

Practice

① $(A+C)(\bar{A}+B)$

$$A \cdot \bar{A} + AB + C\bar{A} + CB$$

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

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

$$\underline{AB + C\bar{A} + \bar{A}BC + ABC}$$

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

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

② $(A+B) \cdot (\bar{A}+C) \cdot (B+C)$

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

$$\underbrace{(A+B)(\bar{A}+C)}_x \cdot \underbrace{(A+B+C)(\bar{A}+B+C)}_{x+y}$$

$$\cancel{AB} + (A+B)C \rightarrow x$$

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

③

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

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

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

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

$$= A \oplus B$$

④

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

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

$$A+B+A$$

$$= A+B$$

⑥

$$(x+y+xy)(x+z)$$

$$(x+y)(x+z)$$

$$x+yz$$

⑤

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

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

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

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

$$z$$

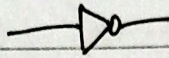
Basic gates — AND
— OR
— NOT

Derived gates — NAND
— NOR
— XOR
— XNOR

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

① Logic Gate :-

1. NOT :-

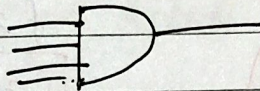


x	$\bar{x}$
0	1
1	0

$$f = \bar{A}$$

2. AND :-

$$f = x \cdot y \cdot z \dots$$

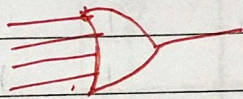


x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Both the input are
one $\rightarrow$ output. 1
Otherwise zero

$$f = x \cdot y$$

3. OR :-



$$f = x_1 + x_2 + \dots + x_n$$

x	y	OR
0	0	0
0	1	1
1	0	1
1	1	1

- if any of input is '1'
output is '1'

$$f = \bar{x}y + x\bar{y} + xy$$

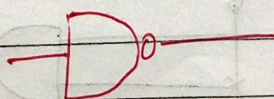
$$f = x + y$$

n variable

$\rightarrow 2^n - 1$ present min term

④ NAND gate

AND + NOT



O/P of NAND is '0' if
all input are '1'

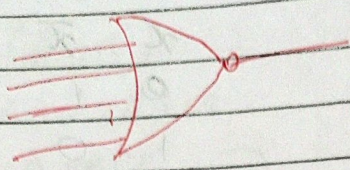
x	y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

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

A.

NAND & ~~OR~~ NOR are universal gate

⑤ NOR.



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

Output of NOR is 1 if all input are 'zero'.

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

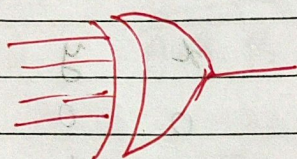
$$f = \overline{x+y}$$

comparator

$\frac{n-1}{2}$ min term

⑥ XOR

$$x \oplus y$$



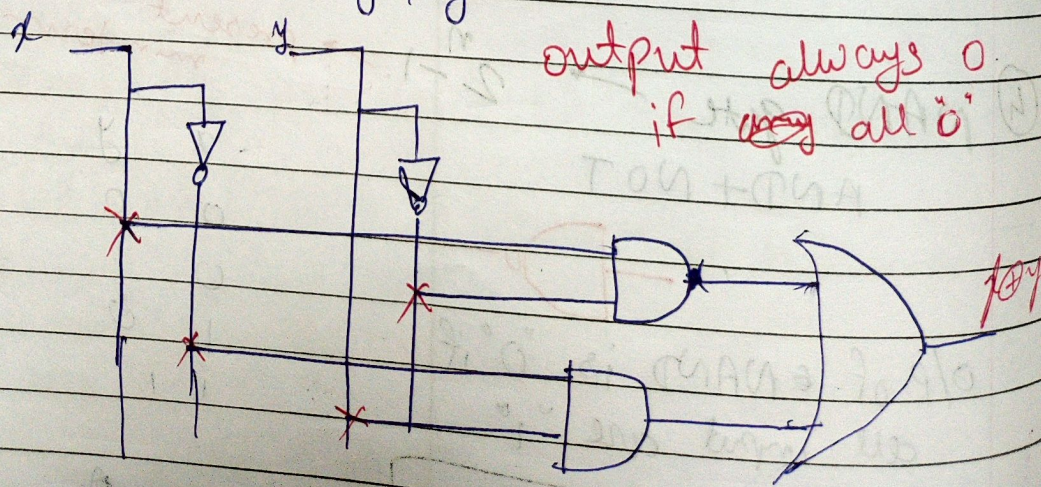
x	y	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

O/p of XOR gate is 0 if even NO. of input are 1

or

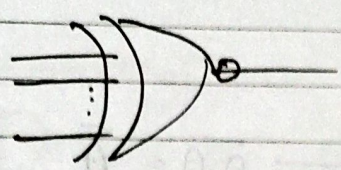
O/p of XOR gate is 1 if odd NO. of input are 1

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



output always 0 if any all 0

⑦ XNOR $\rightarrow$ also comparator $x \odot y$



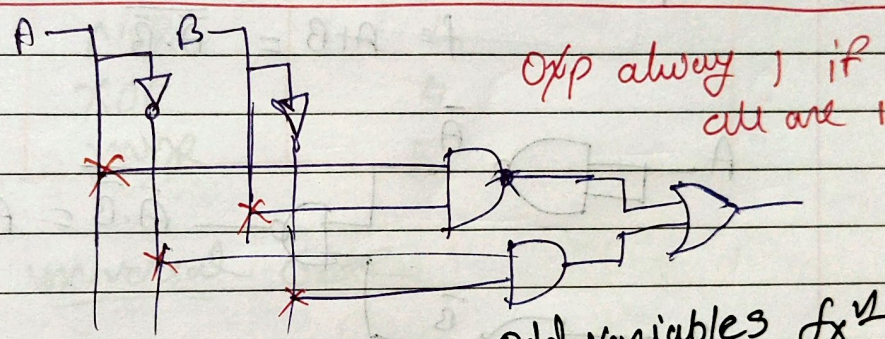
x	y	$x \odot y$
0	0	1
0	1	0
1	0	0
1	1	1

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

★★

★ o/p of XNOR gate is 1 if even no. of inputs are 0

★ o/p of XNOR gate is 0 if odd no. of input are 0



O/p always 1 if all are 1

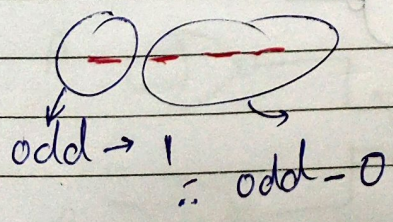
Odd number of variable

even :- 0
 odd :- 1

for odd variables $f_{x \odot y}$

XOR & XNOR
 Both same

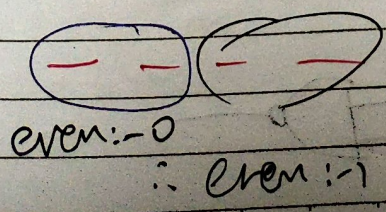
even variable



$$\begin{aligned} \text{XOR} &= 1 \\ \text{XNOR} &= 0 \end{aligned}$$

for even variable

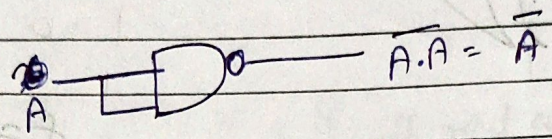
XOR = XNOR



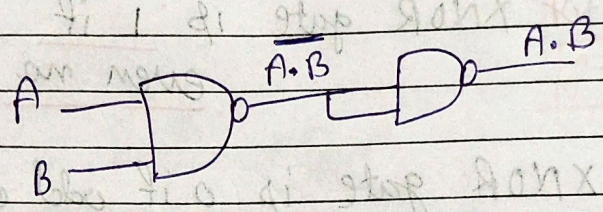
$$\begin{aligned} \text{XOR} &= 0 \\ \text{XNOR} &= 1 \end{aligned}$$

① NAND AS universal gate

① NOT

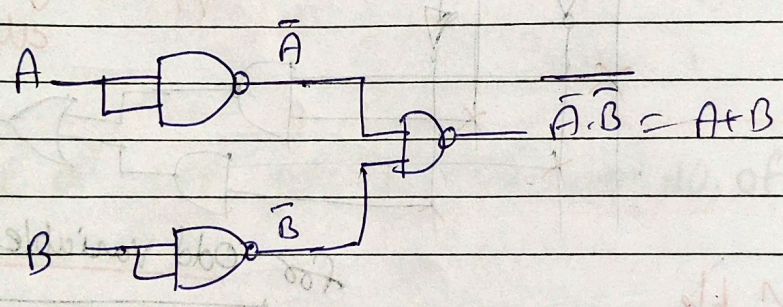


② AND

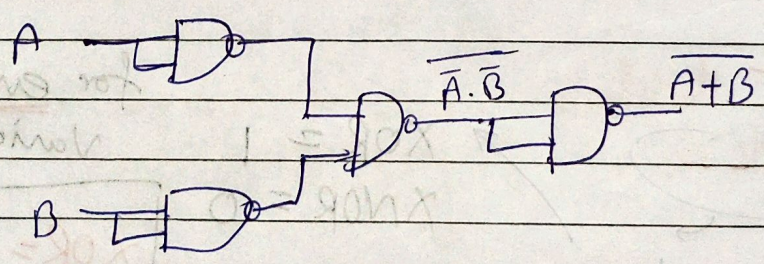


③ OR

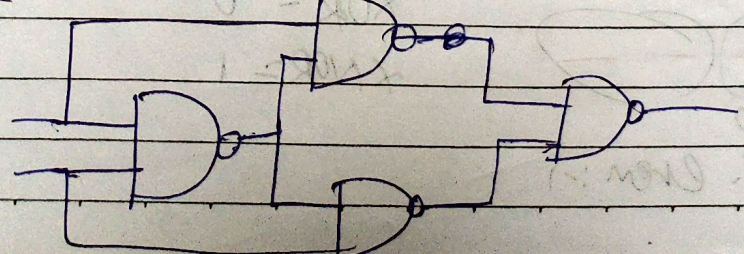
$$f = A + B = \overline{\bar{A} \cdot \bar{B}}$$



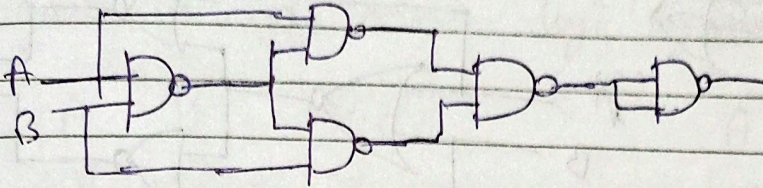
④ NOR



⑤ XOR



⑥ XNOR



~~NOT~~
~~AND~~
~~OR~~
~~NOR~~

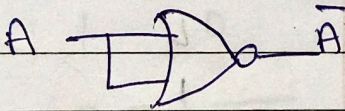
NOT
 AND
 NAND
 OR
 NOR
 XOR
 XNOR

no. of NAND GATE

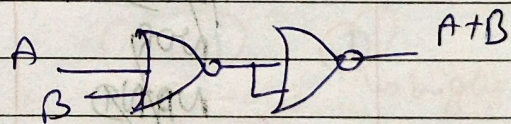
1
 2
 1
 3
 4
 4
 5

⑦ NOR As universal Gate

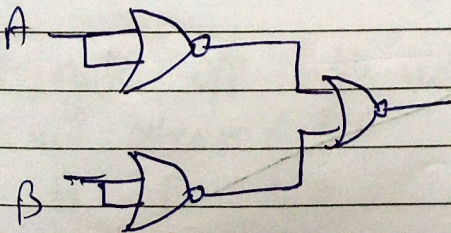
① NOT



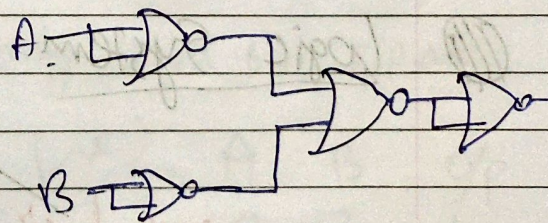
② OR



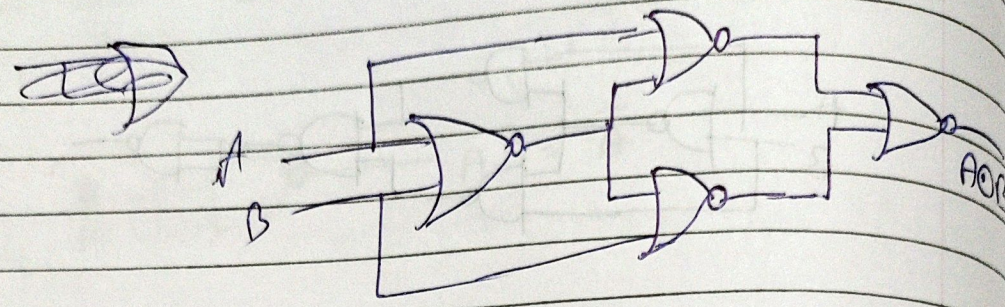
③ AND



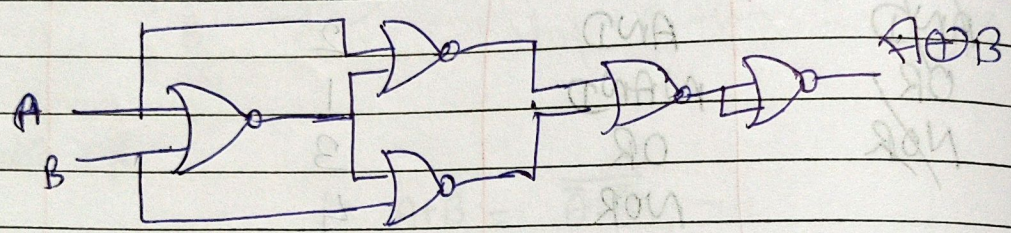
④ NAND



(5) XNOR



(6) XOR



		no. of XNOR required
NOR	NOT	1
OR	AND	3
AND	OR	2
NOR	NAND	4
NAND	NOR	1
	XOR	5
	XNOR	4

Logic System

+ve logic system
 high $\rightarrow$ 1
 low $\rightarrow$ 0

-ve logic system
 high $\rightarrow$ 0
 low $\rightarrow$ 1

$AND \leftrightarrow OR$
 $NAND \leftrightarrow NOR$
 $XOR \leftrightarrow XNOR$

Ques If a gate o/p high voltage only when all I/p's are high & low voltage otherwise. What is gate in +ve & -ve logic system

(+ve)	A	B	O/P	(-ve)	A	B	O/P
0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	1
1	0	0	0	1	0	1	1
1	1	1	1	1	1	1	1

AND +ve $\rightarrow xy$
 OR -ve $\rightarrow A+B$

Ques If gate o/p high voltage only when all I/p's are low & high otherwise. What is gate in +ve & -ve logic system.

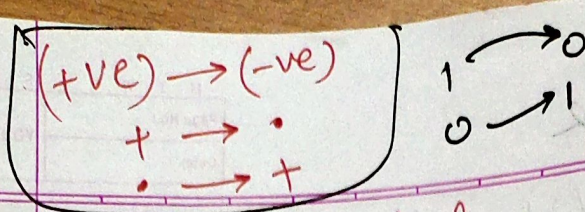
(+ve)	A	B	O/P	(-ve)	A	B	O/P
0	0	0	1	0	0	0	1
0	1	0	0	0	1	0	1
1	0	0	0	1	0	1	1
1	1	1	0	1	1	1	0

$f = \overline{x \cdot y} = \overline{x+y}$
 NOR
 $f = \overline{x \cdot y}$
 NAND

Ques If gate o/p high voltage only when both I/p are different otherwise low.

(+ve)	A	B	O/P	(-ve)	A	B	O/P
0	0	0	0	0	0	0	1
0	1	1	1	0	1	0	0
1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	1

XOR
 XNOR



→ duality principal.

$$\underbrace{(A \bar{B} + \bar{A} C)}_{(+ve)} \rightarrow \underbrace{(A + \bar{B}) \cdot (\bar{A} + C)}_{(-ve)}$$

Que) find dual of expression $\bar{A}B + 1 + \bar{C}D$

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

Que) find dual of expression $\overline{AB} + \overline{C+D}$

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

→ If expression is ~~tan~~ tautology dual also ~~tan~~ tautology.

② 2 ways to represent Boolean fxⁿ

~~Disjunctive~~

SOP

sum of product

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

~~conjunctive~~

POS

product of sum

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

① SOP

→ As specified directly

$$f(A, B, C) = \underline{ABC} + \underline{\bar{A}\bar{B}\bar{C}} + \underline{\bar{A}BC} + \underline{A\bar{B}\bar{C}}$$

All are present
 → no one redundant

$$f(A, B, C) = BC + \bar{B}\bar{C} = \text{S.O.P}$$

common
 economical SOP
 S-SOP
 principle of disjunctive normal form

SOP → SSOP

→ simply add missing variable in each term

$$f(A, B, C) = AB + BC$$

C missing
A missing

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

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

$$f(A, B, C) = \boxed{ABC + AB\bar{C} + \bar{A}BC} \rightarrow \text{S.SOP}$$

$$f(A, B, C) = \sum m(7, 6, 3)$$

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

Binary socho.

$$7 = 111 = xyz$$

$$6 = 110 = xy\bar{z}$$

$$3 = 011 = \bar{x}yz$$

Que) $f(A, B, C) = \bar{A}B + \bar{B}C$ → add missing

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

$$= \boxed{\sum m(3, 2, 5, 1)}$$

Ques) $f(A, B, C, D) = AB + CD$

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

$$= \Sigma m(15, 14, 13, 12, 11, 7, 3)$$

Short cut

$f(A, B, C, D) = AB + CD$

AB → 11 -- --

CD → -- -- 11

$= \Sigma m(12, 13, 14, 15, 3, 7, 11, 15)$

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

↳ not in S-SOP

A B C D

$\bar{A}B \rightarrow 01 --$

$\bar{B}CD \rightarrow -- 011$

$\bar{C}D \rightarrow -- -- 01$

$= \Sigma m(4, 5, 6, 7, 3, 11, 1, 9, 13)$

(2) POS:-

↳ using Max term

↳ complement of min term

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

min term

on

$$f(x, y) = (A+B) \cdot (\bar{A}+\bar{B})$$

$$= \pi m(0,3)$$

① $f(A, B, C) = (A+B) \cdot (B+C)$

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

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

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

$$= \pi M \left(0, \frac{1}{2}, \frac{1}{2} \right)$$

① sop to pos

① $f(A, B) = \bar{A}B + A\bar{B}$

10 $\Sigma m(1, 2)$

$$T_{lm}(0, 3)$$

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

⑦

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

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

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

② $f(A, B, C) = AB + \bar{B}C$

method 1

$$f(A, B, C) = \sum m(6, 7, 5, 1)$$

↓ missing

$$\prod M(0, 2, 3, 4)$$

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

method 2 = $(A+C) \cdot (A+\bar{B}+\bar{C}) \cdot (\bar{A}+B+C)$

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

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

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

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

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

method 2:

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

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

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

$$\begin{aligned} \textcircled{3} \quad f(A, B, C) &= AB + BC \\ &= B(A + C) \\ &= (AB + B)(A + C) \\ &= B(A + C)(A + C) \end{aligned}$$

$$\begin{aligned} \textcircled{4} \quad f(A, B, C) &= AB + CD \\ &= (AB + C)(AB + D) \\ &= (A + C)(B + C)(A + D)(B + D) \end{aligned}$$

POS to SOP

$$\textcircled{1} \quad F(A, B) = \overline{A}\overline{B}(\overline{A} + B) \cdot (A + \overline{B})$$

$$\begin{aligned} f &= \prod m(2, 1) \\ &= \sum m(0, 3) \\ &= (\overline{x}\overline{y}) + (xy) \end{aligned}$$

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

$$\boxed{AB + \overline{A}\overline{B}}$$

$$\textcircled{2} \quad f(A, B, C) = (A + B) \cdot (\overline{B} + C)$$

$$f = \prod m(1, 0, 6, 2)$$

$$f = \sum m(3, 4, 5, 7)$$

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

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

$$\sum m(3, 7, 4, 5)$$

$$\overline{A}\overline{B} + \overline{B}\overline{C} + \overline{A}C + BC$$

$$\boxed{\overline{A}\overline{B} + \overline{A}C + BC}$$

$$\sum m(4, 5, 3, 7)$$

Both same

if Dual & complement are same $\rightarrow$ orthogonal fx

function

①

neutral

adder 1 I/P
adder 0 I/P

- no. of minterm = no. of maxterm

not neutral.

$\rightarrow$ no. of minterm $\neq$ no. of maxterm

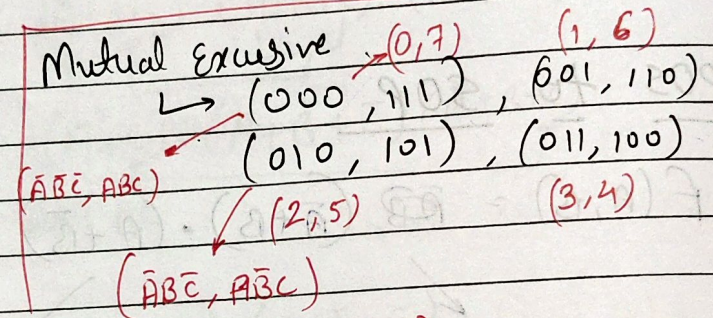
Ex.

XOR, XNOR

AND, NAND
OR, NOR

for n variable
no. of p.f.x

$$= \binom{2^n}{2^{n-1}}$$



② Self dual (natural but not mutual exclusive)

non self dual

$f(x)$ and its dual are same

$f(x)$ and its dual not same

$\rightarrow$ all self dual are neutral

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

$\downarrow$ dual

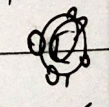
$$(\bar{A} + \bar{B}) \cdot (\bar{A} + B)$$

$\bar{A}$

not mutually exclusive

mutually exclusive terms

- (0, 3)
- (1, 2)



only possible when simplified

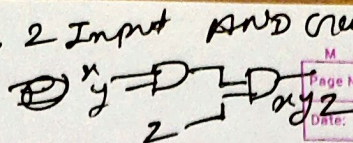
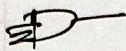
$f(x)$ = only one variable

Ex. $f(A, B) = \bar{A}$

Orthogonal no. of orthogonal

$$\binom{2^{n-1}}{2^{n-2}}$$

3/5/p and Gate = 2, 2 Input AND Gate

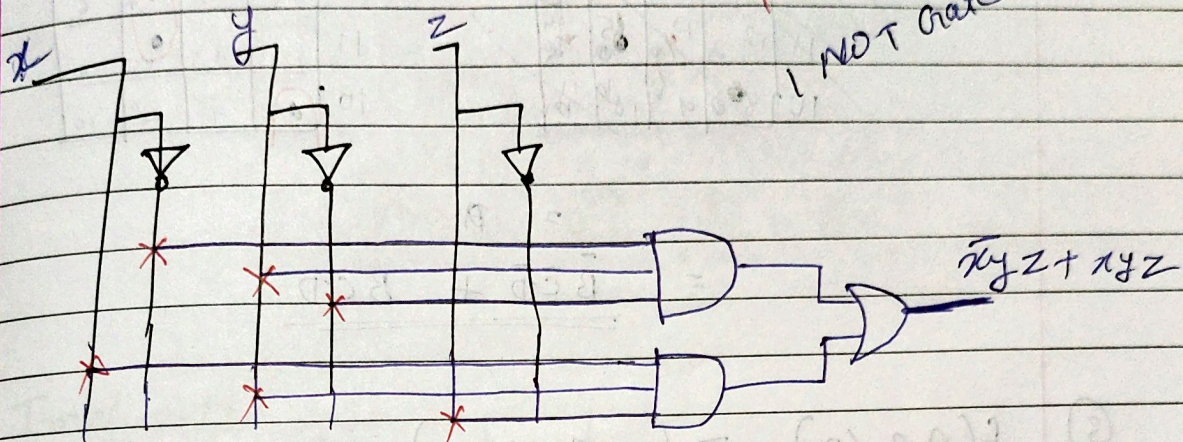


M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

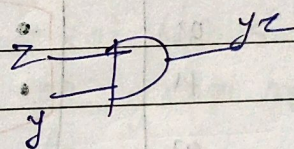
K-map

$$f(x, y, z) = \bar{x}yz + xyz$$

2 AND³ Gate
1 OR Gate
1 NOT Gate



$$\begin{aligned} f(x, y, z) &= \bar{x}yz + xyz \\ &= yz(\bar{x} + x) \\ &= yz \end{aligned}$$



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

we have to use K-map for minimize the fxⁿ

no. of cells in Kmap

= number of variable
2

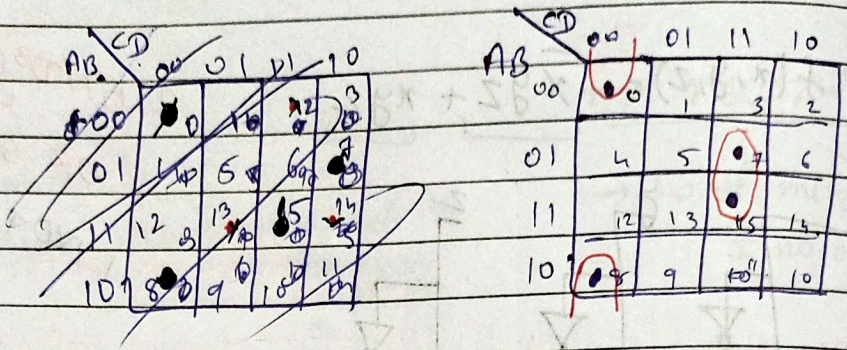
$$f(A, B, C, D) = \sum m(0, 1, 13, 15)$$

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

size of rectangle
1 2 4 8

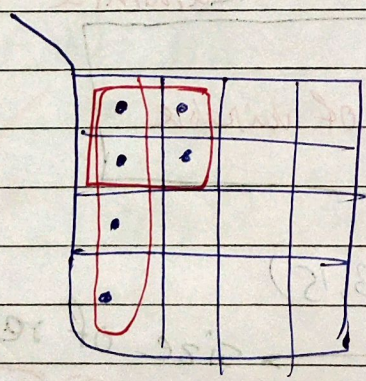
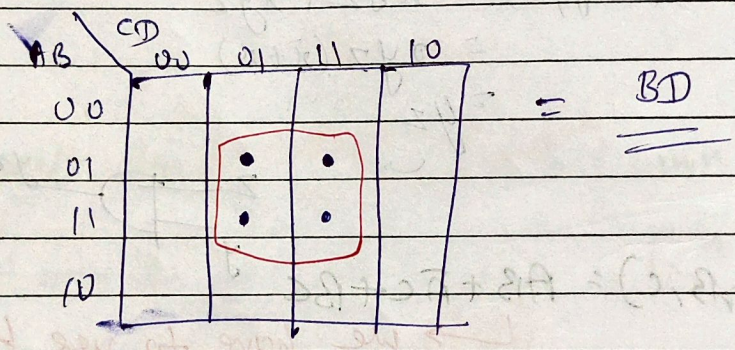
$$f = \bar{A}\bar{B}\bar{C} + ABD$$

② $f(A, B, C, D) = \sum m(0, 7, 8, 15)$

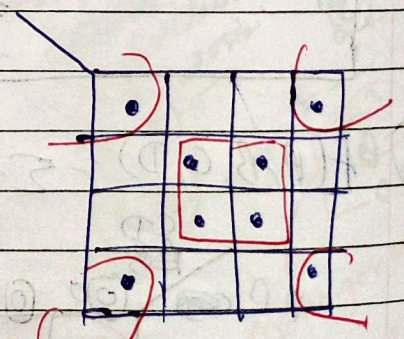


$$= \bar{B} \bar{C} \bar{D} + B C D$$

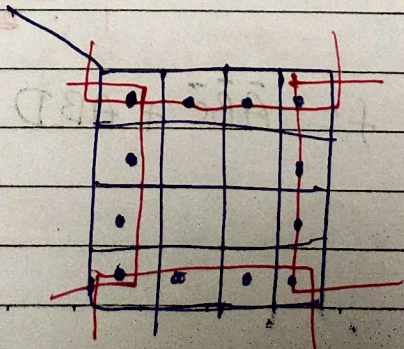
③ $f(A, B, C, D) = \sum m(5, 7, 13, 15)$



reuse



all corner one cell



Q.7) $f(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 5, 10, 11)$

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

→ not a cell because we use all 1 which make minimum cell.

Ans. = $\bar{A}\bar{C} + BC$

③ Implicants

$f_1 \rightarrow f_2$
 $f_1 \subseteq f_2$

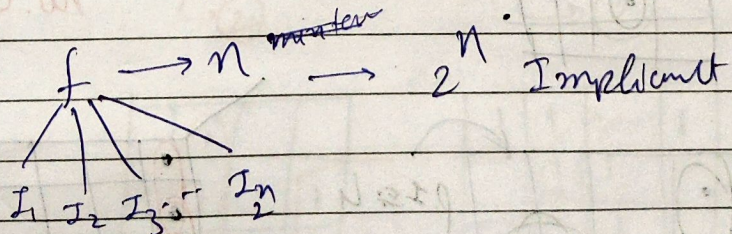
AB \ CD	00	01	11	10
00		1	1	
01		1	1	
11			1	
10	1			1

→ not an implicant.

→ Koi bhi single one or group of one pickad do.

③ Prime Implicant

→ A prime implicant is an implicant that can't be covered by more reduced implicants



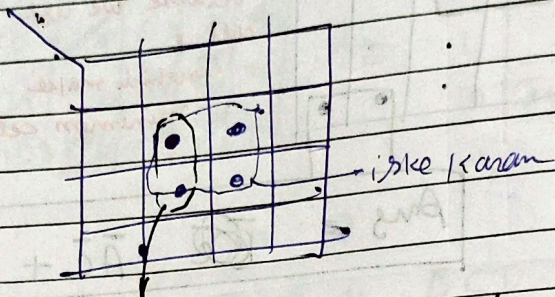
$(I_K) \rightarrow PI$

→ If and only if there is no any

Implicant cover this I_K and its literal count is less than I_K .

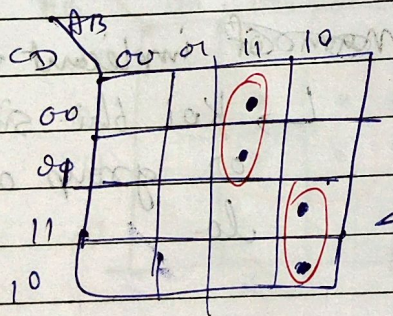
Visualize p.i

→ Any single 1 or group of 1's that can't be included in bigger rectangle in K-map

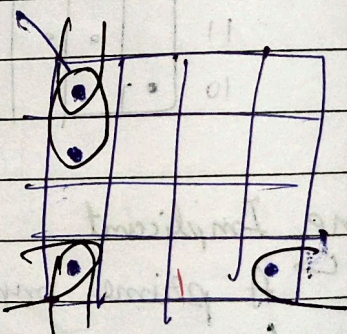


not PI
because isse bad
rectangle me yeh aa rahi
hai

Q. no. of PI is given K-map as

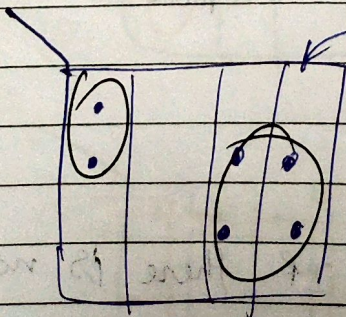


2-PI

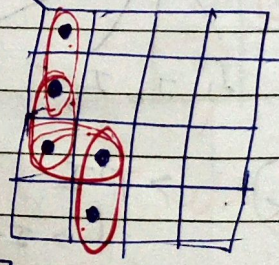


no. of PI = 2

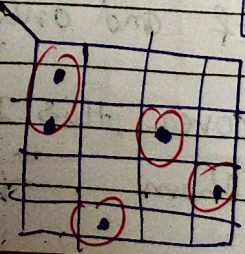
no. of PI = 3



PI = 4



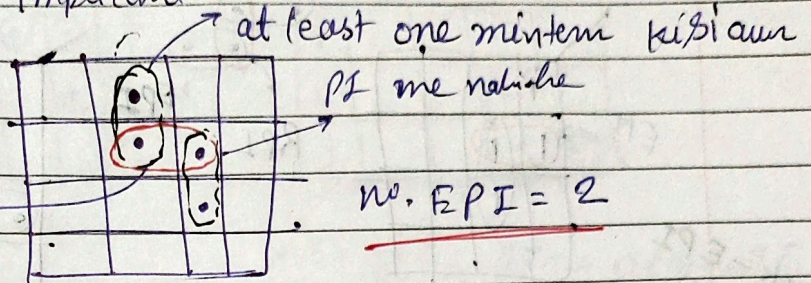
no. of PI = 4



PI = 4

⑧ Essential Prime Implicant :-

↳ Those prime implicant which contain atleast 1 minterm which is not included in any other prime implicant.



no. EPI = 2

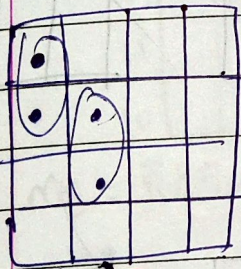
~~exhibi esl~~

minterm nahiche fo

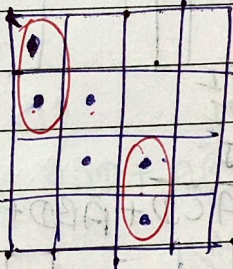
PI = 3

break PI me nah ho.

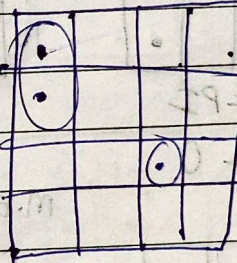
Q7 no. of EPI



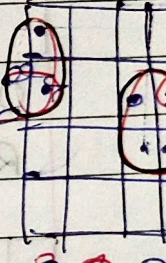
↳ (2)



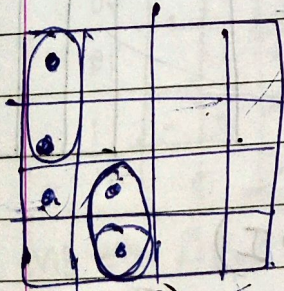
2



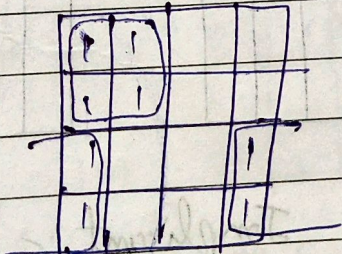
2



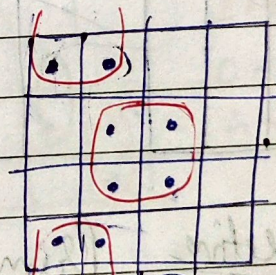
2



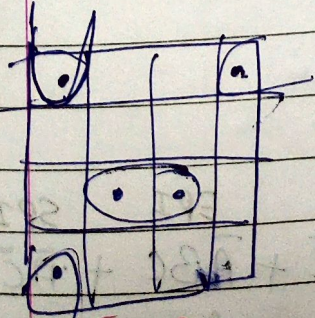
↳ (2)



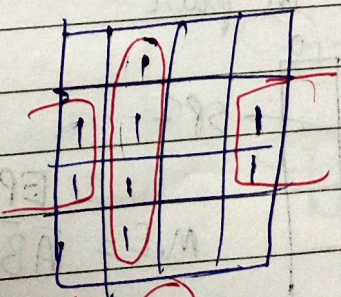
↳ (2)



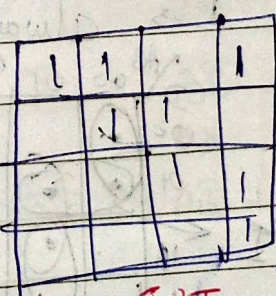
↳ (2)



↳ 3



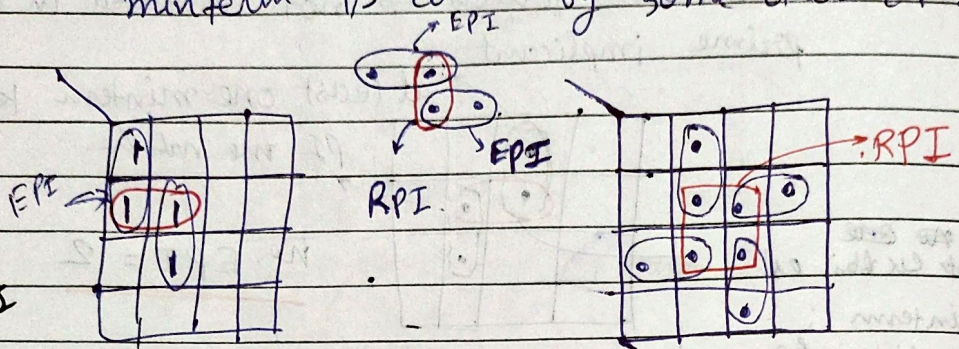
↳ (2)



↳ 0 EPI

Redundant Prime Implicant :: (RPI)

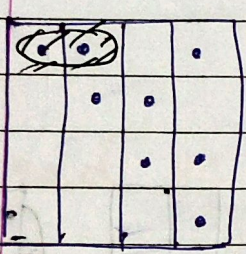
→ A prime implicant in which each of its minterm is covered by some other E.P.I



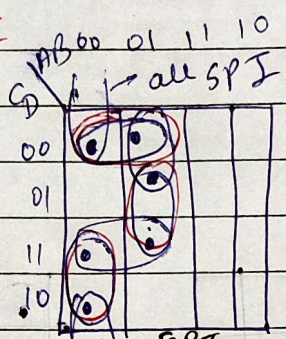
Black - EPI
 Red - RPI

1 RPI

4 - EPI



0 - EPI
 RPI - 0

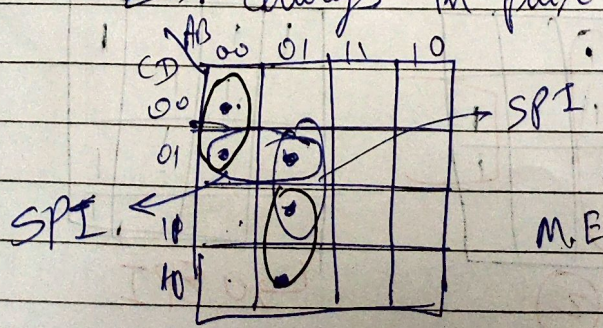


0 - EPI
 0 - RPI
 6 - SPI

$$M.E = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D}$$

Selective Prime Implicant :: (SPI)

→ which is neither EPI nor RPI
 → always in pair



2 EPI

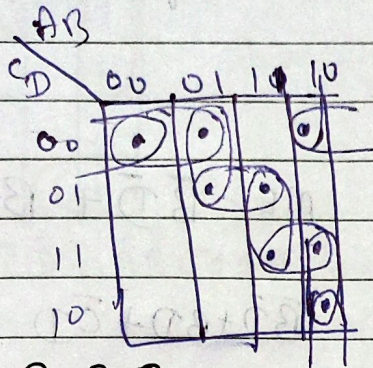
EPI EPI SPI

$$M.E = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

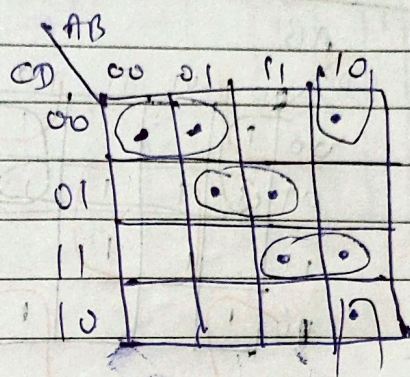
or

$$M.E = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

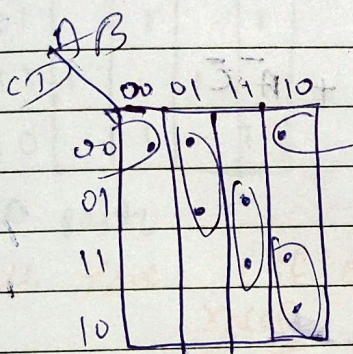
Q. No. SPI



0-ESP EPI
 0-RPI
 8-OSPI

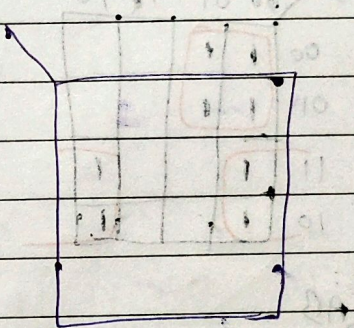


$$M.E = \bar{A}\bar{C}\bar{D} + \bar{B}\bar{C}\bar{D} + A\bar{C}\bar{D} + A\bar{B}\bar{D}$$

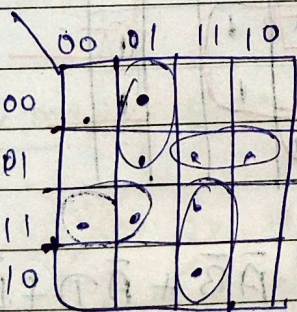


$$M.E = \bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{D} + A\bar{B}\bar{C}$$

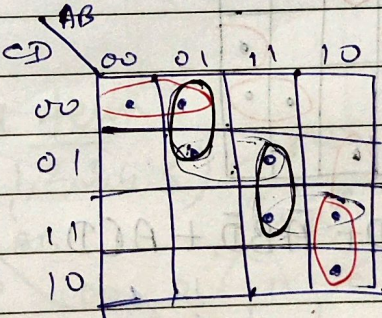
or



Ques



$$M.E = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}\bar{D} + \bar{A}\bar{C}\bar{D} + A\bar{B}\bar{C}$$



2-ESP EPI
 0-RPI
 4-SPI

or

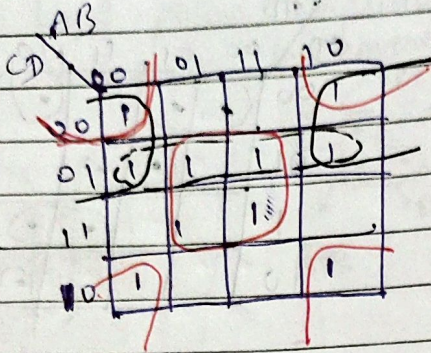
$$M.E = \bar{A}\bar{C}\bar{D} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{D}$$

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

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

Question: minimized Expression 19

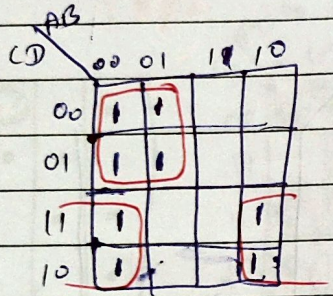
Red-EP I
 Black-SP I



$$ME = \bar{B}\bar{D} + BD + \bar{B}\bar{C}$$

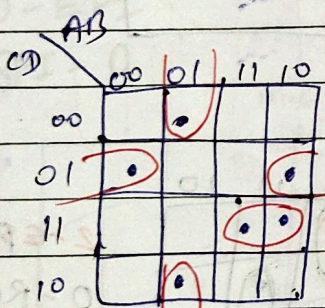
$$\bar{B}\bar{D} + BD + \bar{C}D$$

②

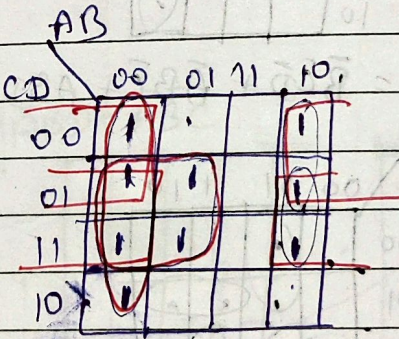


$$ME = \bar{B}\bar{C} + A\bar{C}$$

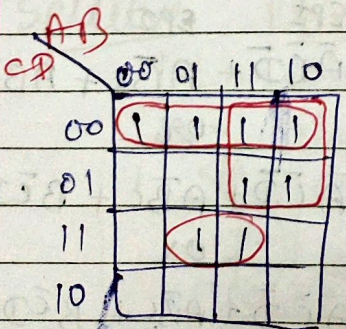
③



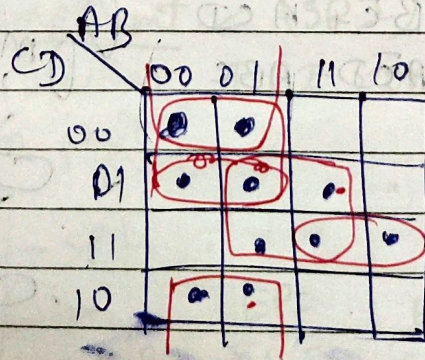
$$ME = \bar{B}\bar{C}D + \bar{A}B\bar{D} + ACD$$



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



$$ME = \bar{C}\bar{D} + A\bar{C} + BCD$$



$$ME = \bar{A}\bar{D} + BD + \bar{A}\bar{C}D + ACD + \bar{A}\bar{C} + ACD$$

If no cell
XOR, XNOR
1 if odd 1

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

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

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

all have odd 1

$$= A \oplus B \oplus C \oplus D$$

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

no pair

all have 1 if even zero
XNOR

$$A \odot B \odot C \odot D$$



K-map using max term

$$f(A, B, C, D) = \prod M(1, 2, 3, 4, 5) \rightarrow \text{POS}$$

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

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

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

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

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

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

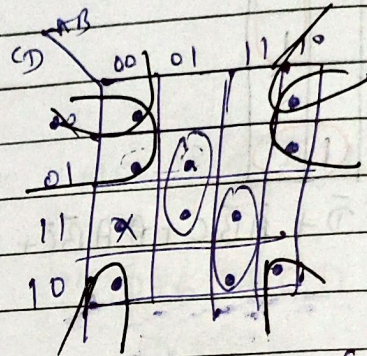
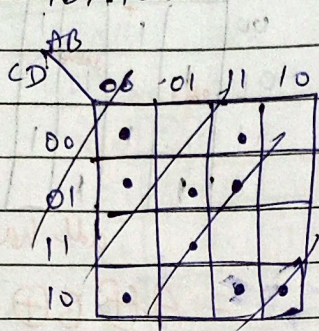
$$\cdot (\bar{A} + \bar{C} + \bar{D})$$

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

cell 1 $\rightarrow$ 4 variable
 2 $\rightarrow$ 3 variable
 4 $\rightarrow$ 2 variable

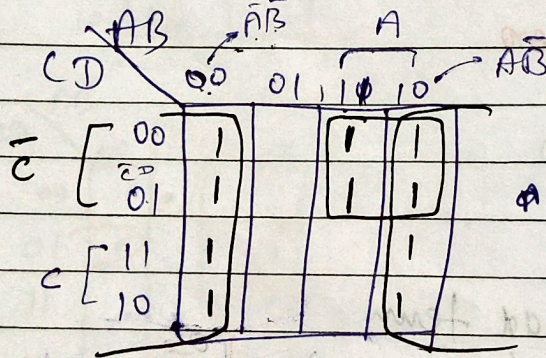
8 $\rightarrow$ 1 variable
 16 $\rightarrow$ 0 variable

② $f(A, B, C, D) = \Pi m(0, 1, 2, 5, 7, 8, 9, 10, 14, 15)$



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

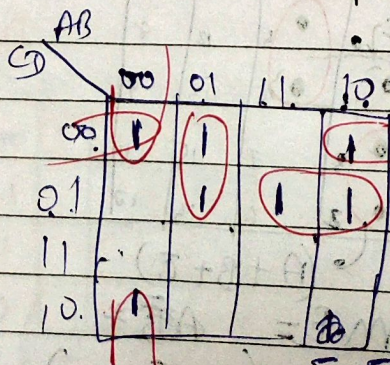
Q) $f(A, B, C, D) = \underbrace{A\bar{B}}_{2 \text{ cell.}} + \underbrace{\bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}}_{2 \therefore 4 \text{ cell.}} + A\bar{C}D + A\bar{C}\bar{D}$



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

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

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



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

① Don't care 99

Bool $f(x^n)$

completely
~~ten~~ $\hookrightarrow$ present
 $\hookrightarrow$ absent

Incomplete

~~ten~~ $\hookrightarrow$ present
 $\hookrightarrow$ absent
 $\hookrightarrow$ Don't care
 $\hookrightarrow$ 'X'
 $\hookrightarrow$ '0' or '1'

Q

① $f(A, B, C, D) = \sum m(0, 1, 2, 3) + d(4)$

$\overline{A}\overline{B}$

we consider here '0'

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

② $f(A, B, C, D) = \sum m(0, 1, 2) + d(3)$

$\overline{A}\overline{B}$

Consider

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

		1	
X	X	1	X
		1	
		1	

Ans = $AB + BC$

③

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

$= BD + BC + A$

If completely specified $f(x^n) \rightarrow \text{SOP} = \text{POS}$
 If incomplete $f(x^n) \rightarrow \text{SOP may, may not be eqn POS}$

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

Ans = $BD + \bar{B}D$

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

minimum L.C 99

$= \bar{B}\bar{D} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$
 (2) (3) (3) = 8

Consider '0'

CD \ AB	00	01	11	10
00	X	0		
01	0		X	0
11		X	X	0
10	X	0	0	X

$12 + 2 + 2 + 3 = \dots$

Que

Find minimum expression for a $f(x^n)$ involving BCD digit - f is 'on' w.r.t odd BCD digit

$f(A, B, C, D) = \sum m(1, 3, 5, 7, 9) + d(10, 11, 14, 15)$

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

Ans = 7

5 - Variable K-map.

cells = $2^5 = 32$

BC

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

BC

DE	00	01	11	10
00	16	20	24	
01	17	21	25	
11	19	23	27	
10	18	22	26	

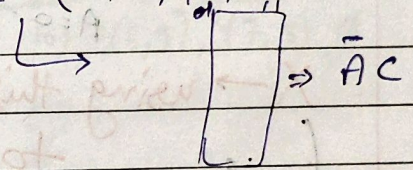
A=0

A=1

① $\sum m(4, 5, 6, 7)$

= $\bar{A}BC$

② $\sum m(4, 5, 6, 7, 12, 13, 14, 15)$

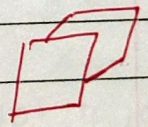


③ $f(A, B, C, D, E) = \sum m(4, 5, 6, 7, 20, 21, 22, 23)$

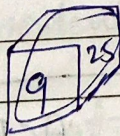
= $\bar{A}\bar{B}C + A\bar{B}C$

= $\bar{B}C$

Imagine yeh ek
 duere ke uper niche
 hai:



④ $\sum m(9, 25)$



$\Rightarrow B\bar{C}\bar{D}E$

Que The Boolean Expression
 $\bar{A}BE + BCDE + B\bar{C}DE + \bar{B}CDE$
 can be simplified to

$BE + \bar{B}DE$

if the don't care minterms are

		BC				BC			
		00	01	11	10	00	01	11	10
A	0	00				00			
	0	01		1	1	01		X	1
	0	11		1	1	11		1	X
	0	10	1	X		10	1	X	
		A=0				A=1			

X → using this we can reduce this
 to $BE + \bar{B}DE$

d(6, 22, 29, 27)

$\bar{B}CDE + ABCDE + ABCDE + ABCDE$

6 Variable - K map

B	0	0	1
	1	1	1
C	0	1	1
	1	1	1

$\bar{B}CF + ACF$

Que which minterm

Questions

① Which minterms must be added to make follow f_x self dual??

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

self dual only when it have literal count '1'

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

$$\Sigma m(5, 6, 7, 11, 13, 15)$$

for self for self dual $(0, 15), (1, 14), (2, 13), (3, 12), (4, 11), (5, 10), (6, 9), (7, 8)$

1, 3 include or

1, 12 include or

14, 3 include or

14, 12 include

② Consider the following boolean function of four variables:

$$f(A, B, C, D) = \Sigma m(0, 2, 5, 7, 8, 13, 15)$$

The function is:

A) Independent of one variable

B) Independent of two variable

C) " " three variable

dependent of all variable

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

③ $f_1(A, B, C, D) = \sum m(1, 2, 5, 7, 9, 13, 15)$

$f_2(A, B, C, D) = \sum m(2, 5, 7, 10, 11, 14)$

what are the minterms in

a) $\bar{f}_1 = \sum m(0, 3, 4, 6, 8, 10, 11, 12, 14)$

b) $\bar{f}_2 = \sum m(0, 1, 3, 4, 6, 8, 9, 12, 13, 15)$

c) $f_1 \cdot f_2 = \sum m(2, 5, 7) \Rightarrow f_1 \cdot f_2 = \text{all } f_1, f_2$

d) $f_1 + f_2 = \sum m(1, 2, 5, 7, 9, 10, 11, 13, 14, 15)$

e) $f_1 \oplus f_2 = \sum m(1, 9, 10, 11, 13, 14, 15)$

d) $f_1 \odot f_2 = \sum m(0, 2, 3, 4, 5, 7, 6, 8, 12)$

④ How many functions does f_1, f_2 & $f_1 + f_2$ represents?

0 $x \rightarrow 0$ $f_1(a, b, c) = \sum m(0, 2, 4) + d(3, 5, 7) \rightarrow 2^3 = 8 \text{ } f_1 x^n$

1 $x \rightarrow x$ $f_2(a, b, c) = \sum m(2, 3) + d(1, 6, 7) \rightarrow 2^3 = 8 \text{ } f_2 x^n$

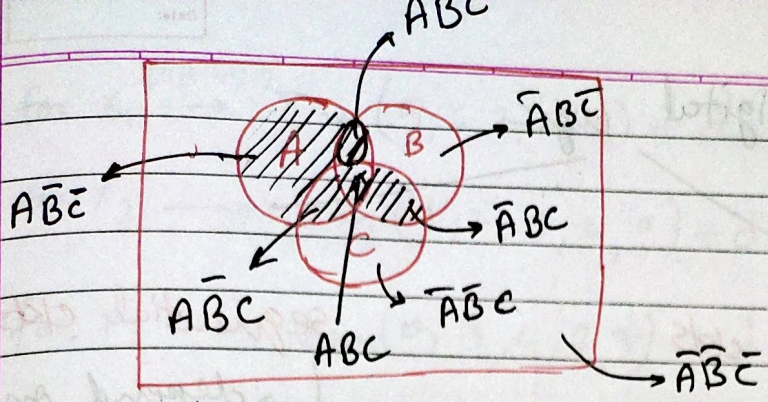
for f_2 f_1 $f_1 + f_2$ $\rightarrow x$

$f_1 \cdot f_2 = \sum m(2) + d(3, 7) \rightarrow 4 \text{ } f_1 f_2 x^n$

$f_1 + f_2 = \sum m(0, 2, 3, 4) + d(1, 6, 7, 5)$

$\rightarrow 2^4 = 16 \text{ } f_1 + f_2 x^n$

(5)



Ans:- $A + \bar{A}BC$
 $\boxed{= A + BC}$

Variable Entered Map [VEM] :-

→ Kmap with variable in cell.

$C = 0$

$\min \text{ term} = 1$
 $0, 0 \ C = 1$

A	B	C	$f(A,B,C)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$f(A,B,C) = \sum m(1, 2, 3, 5, 6)$

Another method

$f(A,B,C) = \sum m(1, 2, 3, 5, 6)$
 ↳ Kmap Banake solve

step 1) put all c & $\bar{c} = 0$ → write f_x^n

step 2) put all $\bar{c} = 1$ & $c = 0$ & ~~1~~ 1 ko don't care
 ↳ $(f_x^n) \bar{c}$

step 3) put all $c = 1$ & $\bar{c} = 0$ & 1 ko don't care
 ↳ $(f_x^n) c$



ckt in Digital Logic.

Combinational ckt's

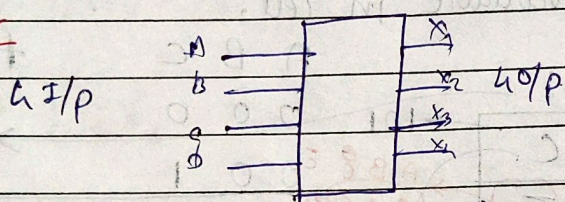
↳ depend on input only

sequential ckt's

↳ depend on present input & past output

Q → Design BCD to excen-3 No. converter

step 1:-



step 2:-

A	B	C	D	X ₁	X ₂	X ₃	X ₄
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

for $X_1 \xrightarrow{P(A,B,C,D)} \Sigma m(5,6,7,8,9) + d(10,11,12,13,14,15)$

for $X_2 \xrightarrow{\quad} \Sigma m(1,2,3,4,9) + d(10,11,12,13,14,15)$

for $X_3 \xrightarrow{\quad} \Sigma m(0,3,4,7,8) + d(10,11,12,13,14,15)$

for $X_4 \xrightarrow{\quad} \Sigma m(0,2,4,6,8) + d(10,11,12,13,14,15)$

K map $\rightarrow$ for X_1

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

$X_1 = A + BD + BC$

for X_2

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

$X_2 = \bar{A}\bar{B}C + \bar{B}\bar{C}D + B\bar{C}\bar{D}$
 $X_2 = \bar{B}D + \bar{B}C + B\bar{C}\bar{D}$

X_3

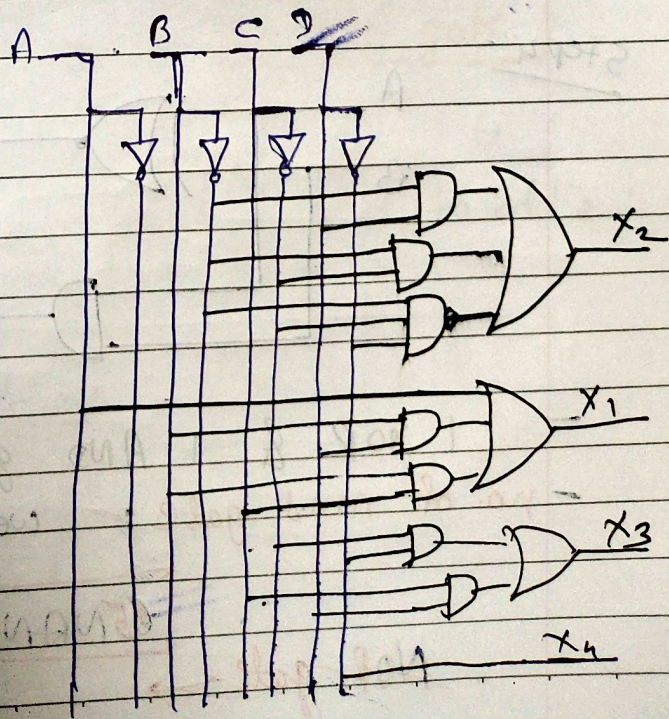
AB \ CD	00	01	11	10
00	1	1	X	1
01			X	
11	1	1	X	X
10			X	X

$X_3 = \bar{C}\bar{D} + CD$

$X_4 \Rightarrow$

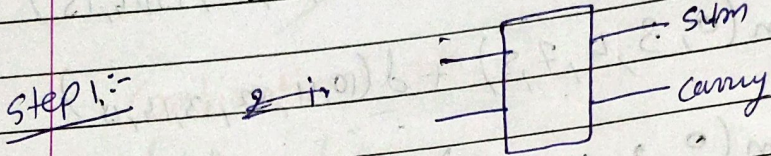
AB \ CD	00	01	11	10
00	1	1	X	1
01			X	
11			X	X
10	1	1	X	X

$X_4 = \bar{D}$



Half - Adder

↳ used to add 2, 1-bit number.



Step 2:-

A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

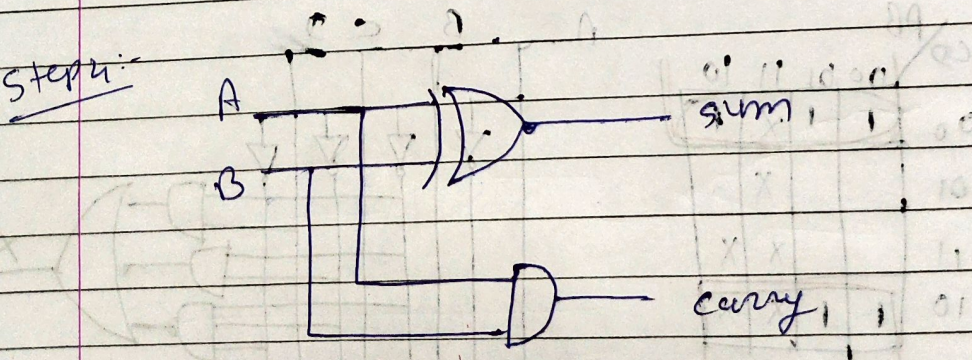
Step 3:-

for sum:- $\Sigma(1, 2)$

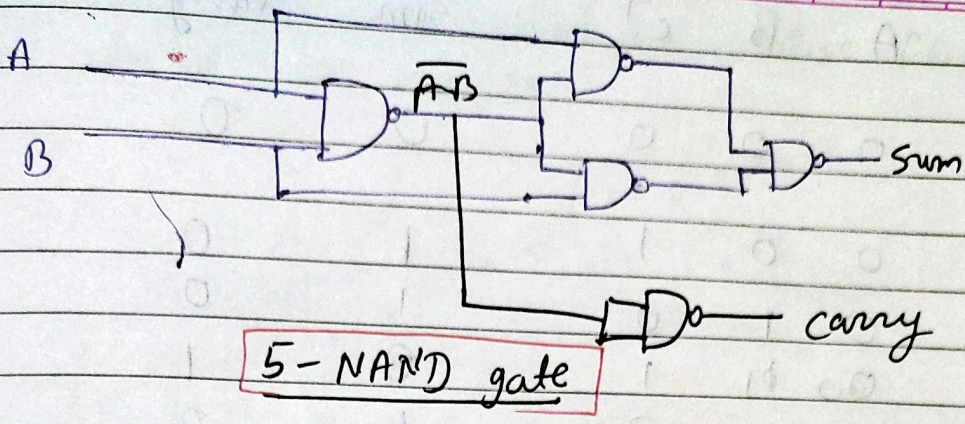
$Sum = \bar{A}B + A\bar{B}$

$A \oplus B$

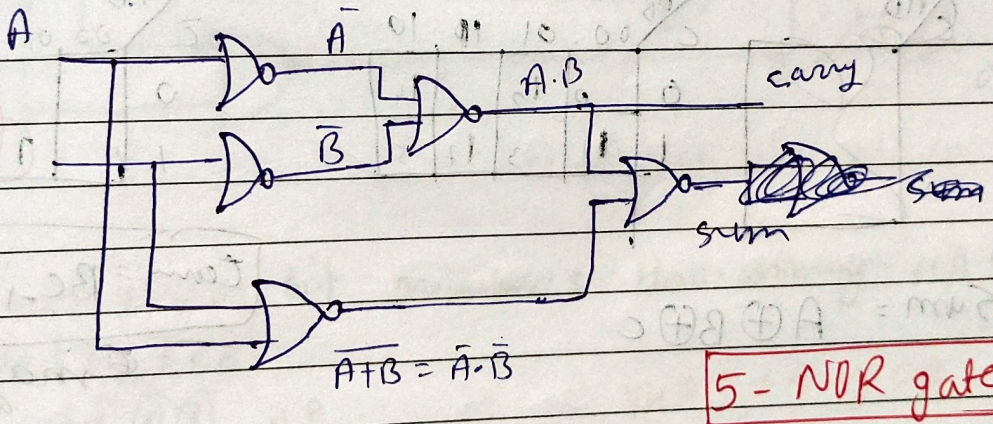
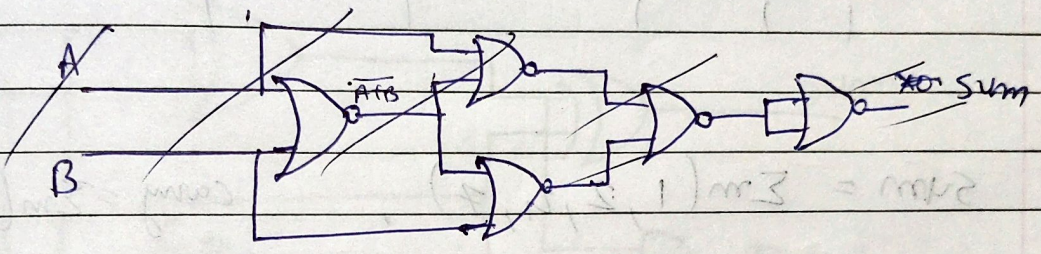
for carry:- $A.B$



- 1 XOR & 1 AND gate
- no. of nand gate :- we know for XOR = 4 NAND
for AND = 1 NAND
- ∴ **5 NAND gate**
- NOR gate →



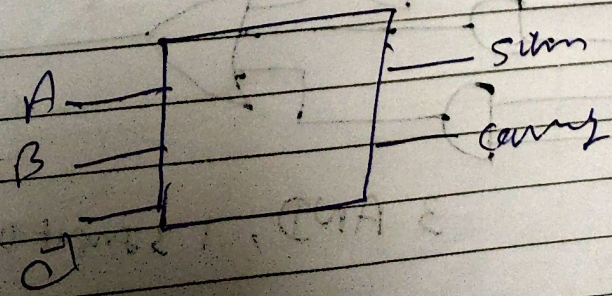
no. of nor



Full adder

Full adder is a c.c. which is used to add 2; 1 bit number & 1 previous carry.

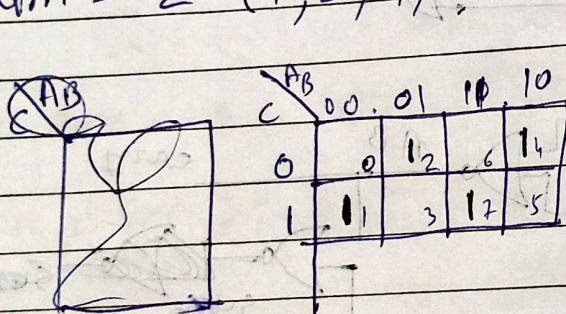
step 1:-



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

$$\text{Sum} = \sum m(1, 2, 4, 7)$$

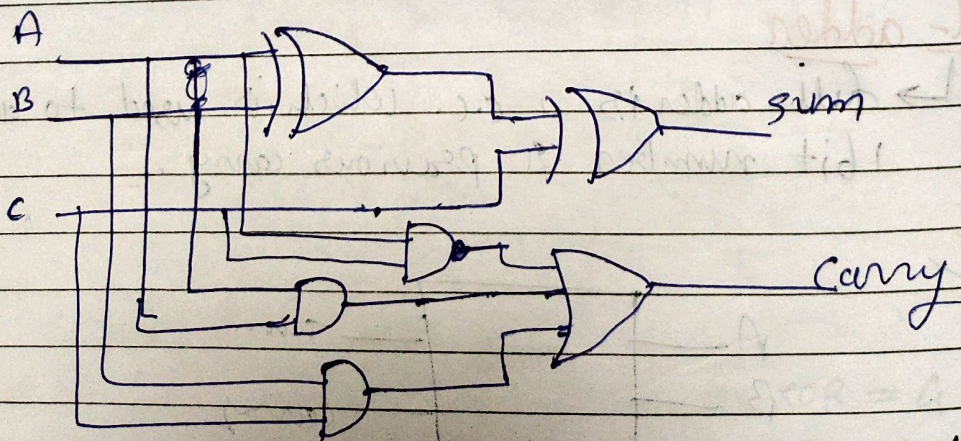
$$\text{Carry} = \sum m(3, 5, 6, 7)$$



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

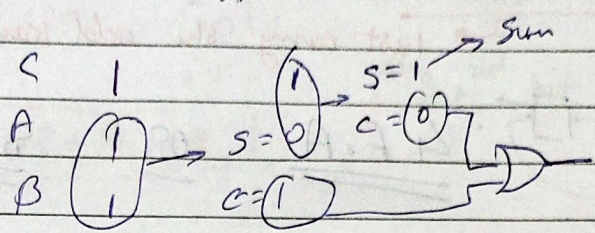
$$\text{Carry} = BC + AC + AB$$

Majority fn

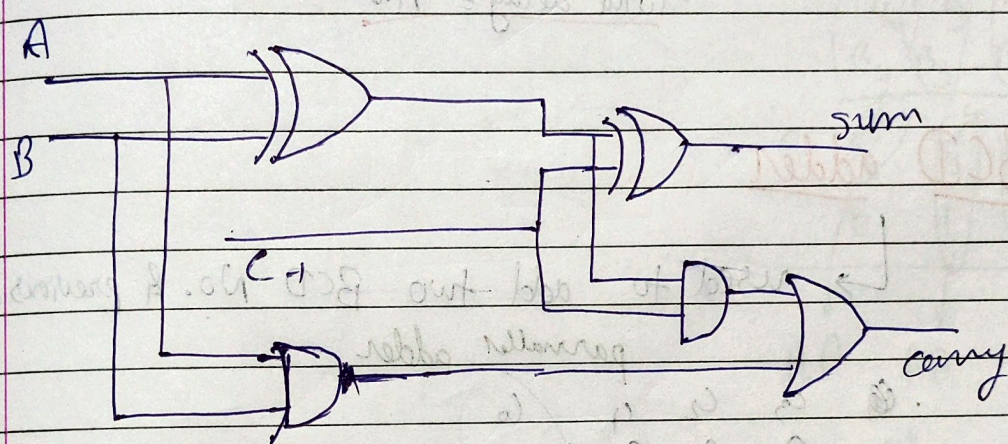


3 AND, 1 3 input OR, 2 XOR

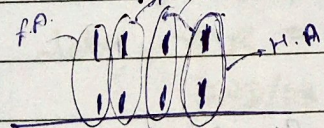
→ How many HA are needed to realise FA?
 what else do we need?



2 H.A & 1 OR gate.



→ To add 2, 4 bit numbers How many HA & FA are needed?



$$1 \text{ HA} + (n-1) \text{ FA}$$

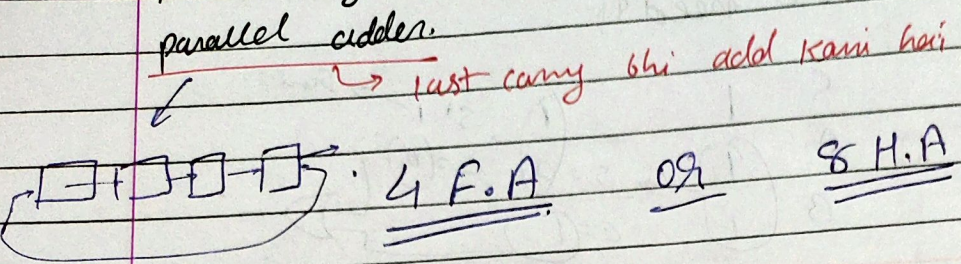
$$1 \text{ HA} + 3 \text{ F.A}$$

$$7 \text{ H.A}$$

$$1 \text{ HA} + 2(n-1) \text{ FA}$$

$$= 2n-1 \text{ H.A.}$$

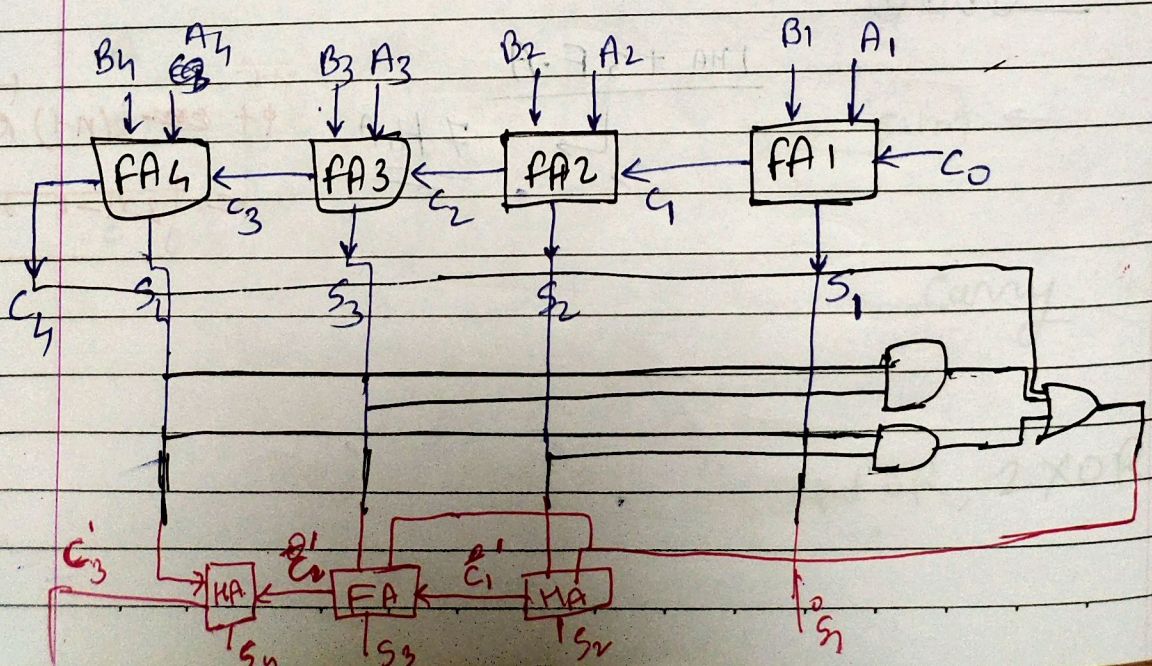
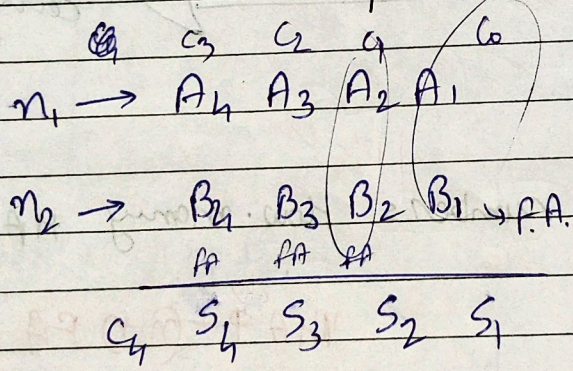
→ How many HA & F.A are needed to design 4 bit parallel adder.



→ 'n' bit parallel adder = n F.A
 delay of each F.A = d
Total delay = n.d

BCD adder

→ used to add two BCD No. & previous carry parallel adder



for invalid BCD check

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

10010
 1011
 11...

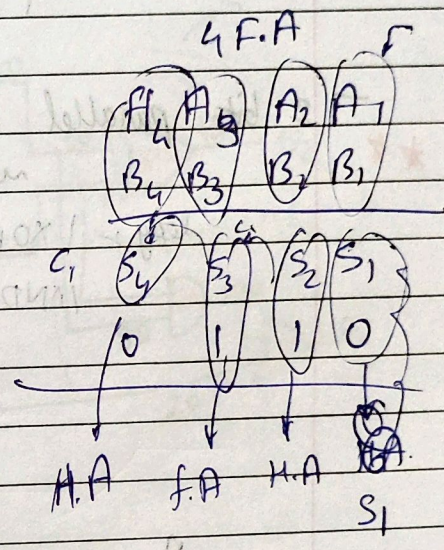
1st and 2nd bit is 1

or
 1st & 2nd bit is = 1

$AB + AC + C$

or C_n is one

→ 5 F.A + 2 H.A

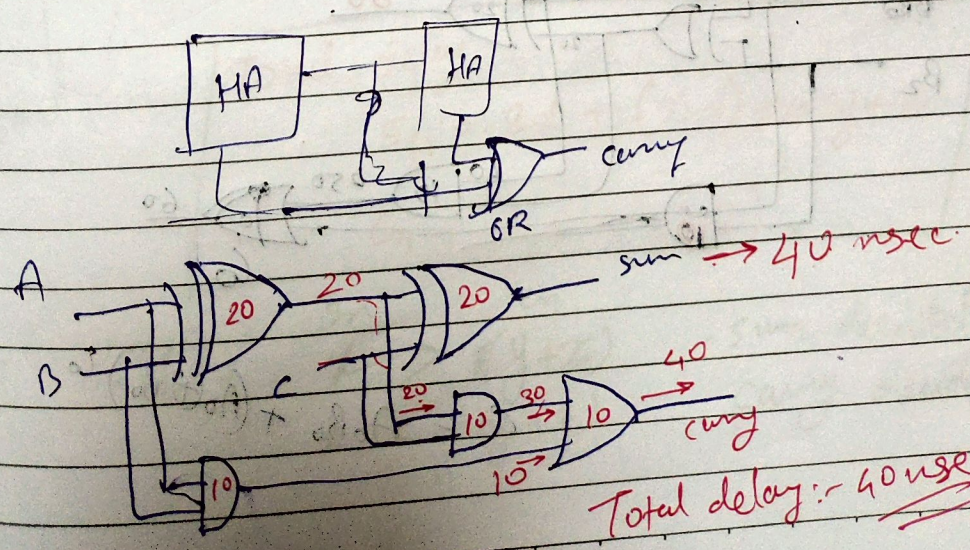


Look ahead Carry Adder

→ Consider an Implementation of FA using 2 H.A.

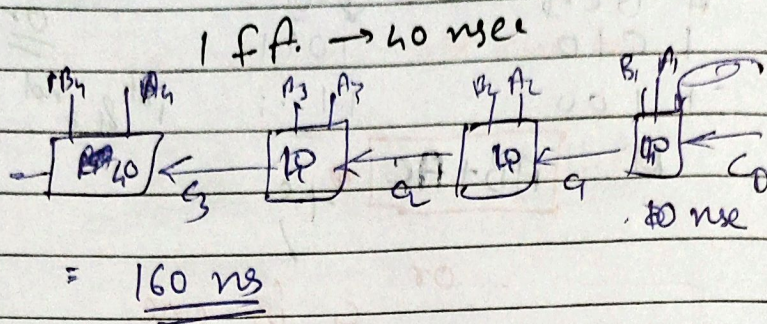
Delay of XOR = 20 nsec

AND/OR = 10 nsec



Total delay :- 40 nsec

→ 2 bit parallel adder. (ripple carry) or ripple



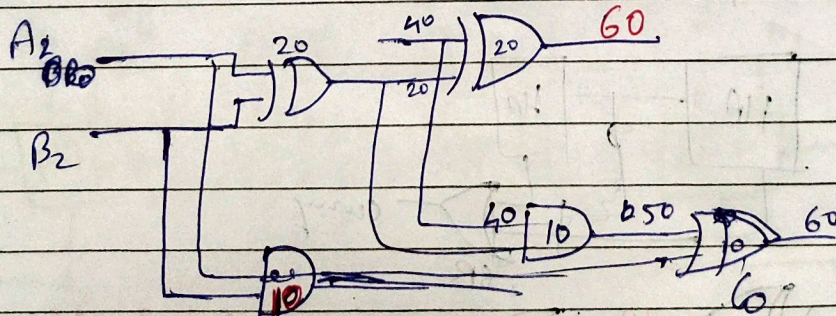
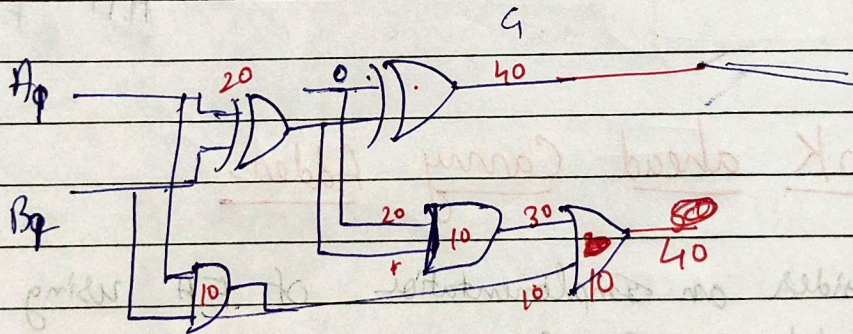
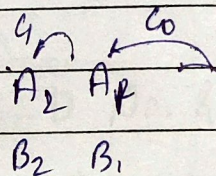
→ 2 bit parallel adder

using H.A.

delay: 1 XOR → 20 nsec

1 AND/OR → 10 nsec

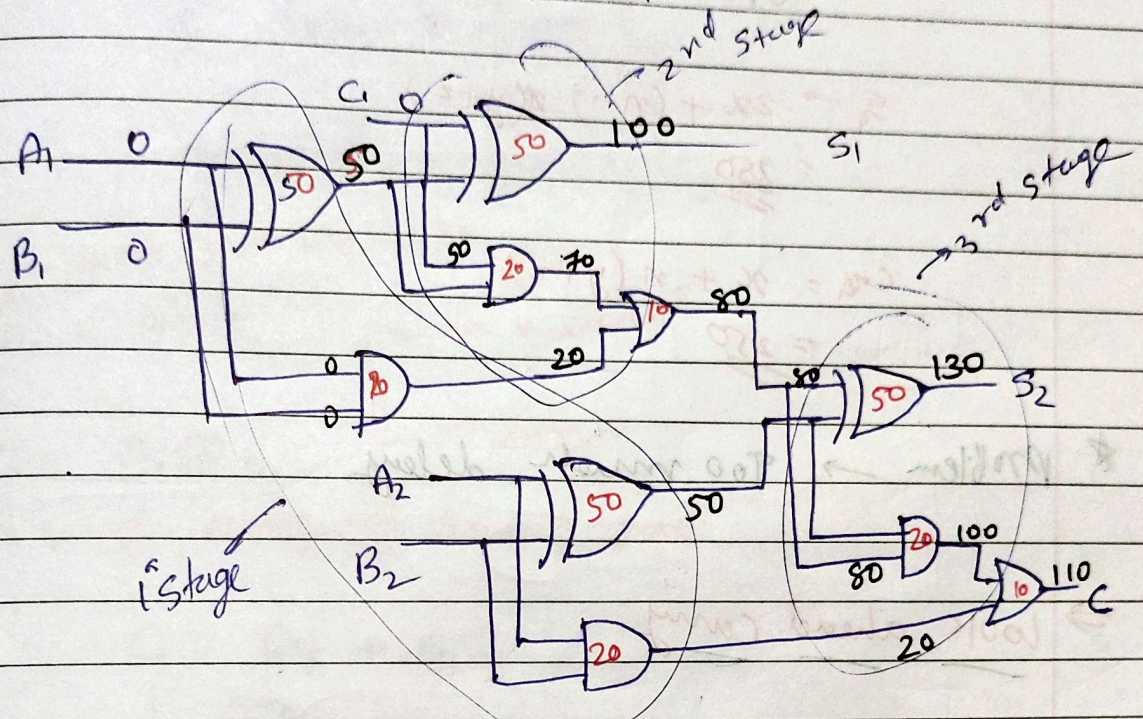
Ans = 60 nsec



$$C_0 = A_0 B_0 + (A_0 \oplus B_0) C_0$$

→ 2 bit parallel adder

delay XOR → 50
 AND → 20
 OR → 10

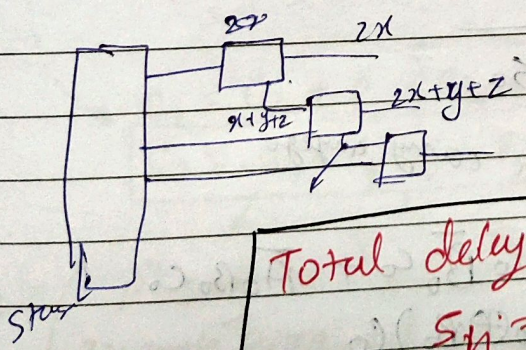


Total delay = 130 ns

→ n bit number

↳ (n+1) stage

XOR → x
 AND → y
 OR → z



Total delay

$$S_n = 2x + (n-1)x + y + z$$

$$C_n = x + n(y+z)$$

diff if $x > (y+z)$
 $x < (y+z)$

- sum decided
 carry decided factor

Look ahead carry $\Rightarrow A_0 \cdot B_0 + (A_0 \oplus B_0) C_0$

Example

XOR $\rightarrow 50 \text{ nsec}$
AND $\rightarrow 30 \text{ nsec}$
OR $\rightarrow 20 \text{ nsec}$

4 bit num

$n=4$

$$S_i \rightarrow 2x + (n-1)(y+z)$$

$$= 250$$

$$C_{out} = x + n(y+z)$$

$$= 250$$

* Problem $\rightarrow$ Too much delay

$\Rightarrow$ look ahead carry

$\rightarrow$ 4 bit Look ahead carry adder

$$n_1 \rightarrow A_3 A_2 A_1 A_0$$

$$n_2 \rightarrow B_3 B_2 B_1 B_0$$

$$C_4 \quad S_3 \quad S_2 \quad S_1 \quad S_0$$

Both 1 Toh carry aayega

$$C_1 \Rightarrow A_0 \cdot B_0 + A_0 \bar{B}_0 C_0 + \bar{A}_0 B_0 C_0$$

$$C_1 = A_0 \cdot B_0 + (A_0 \oplus B_0) C_0$$

Both A_0 & B_0 (1) $\rightarrow C_1$ is 1

if A_0 or B_0 any one is 1 & C_0 is 1 then carry 1

$$C_1 = A_0 B_0 + (A_0 \oplus B_0) C_0$$

✖ ✖ ✖ ✖ ✖

$$C_1 = A_0 B_0 + (A_0 \oplus B_0) C_0$$

$$C_2 = A_1 B_1 + (A_1 \oplus B_1) C_1$$

$$C_3 = A_2 B_2 + (A_2 \oplus B_2) C_2$$

$$C_4 = A_3 B_3 + (A_3 \oplus B_3) C_3$$

→ carry generated at i^{th} stage

$$G_i = A_i \cdot B_i \quad \text{--- generate}$$

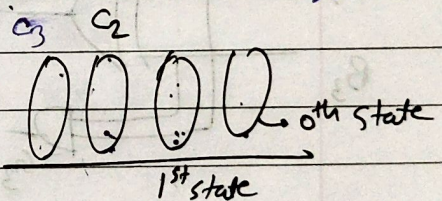
→ carry propagated from i^{th} stage

$$P_i = A_i \oplus B_i \quad \text{--- for}$$

$$C_1 = G_0 + P_0 C_0$$

→ carry generate here (or) carry propagate here

$$\begin{aligned} C_2 &= G_1 + P_1 C_1 \\ C_3 &= G_2 + P_2 C_2 \\ C_4 &= G_3 + P_3 C_3 \end{aligned}$$



$$C_2 = G_1 + P_1 \{ G_0 + P_0 C_0 \}$$

$$C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_2 = \left(\begin{array}{l} 1^{st} \text{ stage ne} \\ \text{generate kiya} \end{array} \right) \text{ or } \left(\begin{array}{l} 0^{th} \text{ stage} \\ \text{pe generate} \\ \text{huan tha and } P_1 \\ \text{ne propogate kiya} \end{array} \right) \text{ or } \left(\begin{array}{l} \text{initially carry} \\ \text{thar 0th stage} \\ \text{and 1st stage ne} \\ \text{Forward kily} \end{array} \right)$$

$$C_3 = \left(\begin{array}{l} 2^{nd} \text{ stage pe} \\ \text{generate} \end{array} \right) \text{ or } \left(\begin{array}{l} 1^{st} \text{ stage pe} \\ \text{generate huan} \\ \text{and } 2^{nd} \text{ stage} \\ \text{ne prop} \end{array} \right) + \left(\begin{array}{l} 0^{th} \text{ stage pe} \\ \text{generate huan} \\ \text{1st 2nd ne forward} \end{array} \right) + \left(\begin{array}{l} \text{initially} \\ \text{thar 0th stage} \\ \text{ne forward} \end{array} \right)$$

$$C_3 = G_2 + G_1 P_2 + G_0 P_1 P_2 + C_0 P_1 P_2$$

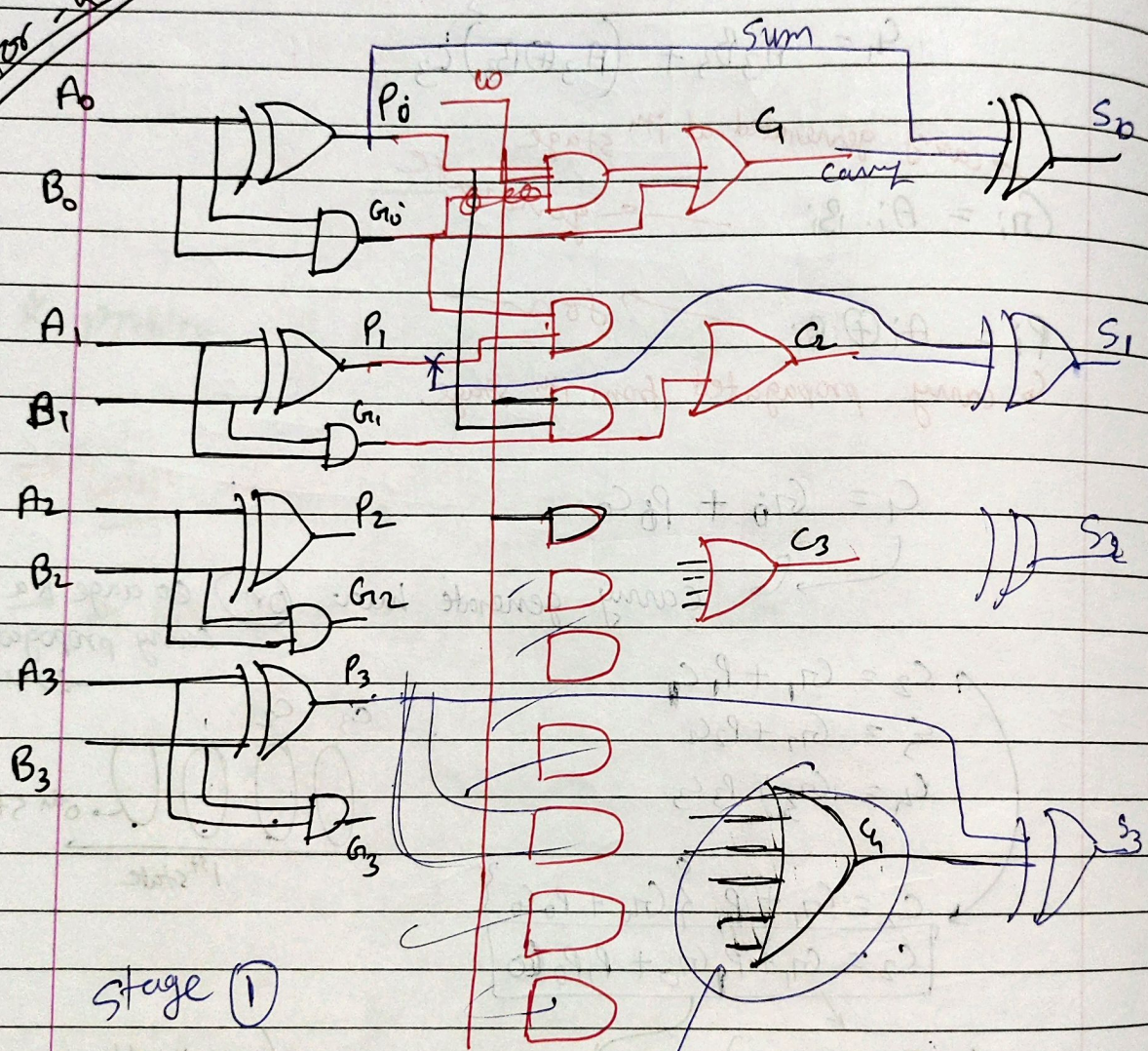
$$C_4 = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + C_0 P_1 P_2 P_3$$

$c_3 \ c_2 \ c_1 \ c_0$
 $A_3 \ A_2 \ A_1 \ A_0$
 $B_3 \ B_2 \ B_1 \ B_0$

$G_0 = A_0 \oplus B_0$
 $G_i = A_i \cdot B_i$
 $P_i = A_i \oplus B_i$

$C_1 = G_0 + C_0 P_0$
 $C_2 = G_1 + G_0 P_1 + C_0 P_0 P_1$
 $C_3 = G_2 + G_1 P_2 + G_0 P_1 P_2 + C_0 P_0 P_1 P_2$
 $C_4 = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + C_0 P_0 P_1 P_2 P_3$

for 4 Bit



Stage ①

Stage ②

Stage ③

Stage ④

fan in no. of Input

Max^m fan in for And gate is 5
 max^m fan in for OR gate is 5

for ~~an~~ n-bits

HA, HS

5 NAND

$C_{n-1} \dots C_2 C_1 C_0$
 $A_{n-1} \dots A_2 A_1 A_0$
 $B_{n-1} \dots B_2 B_1 B_0$

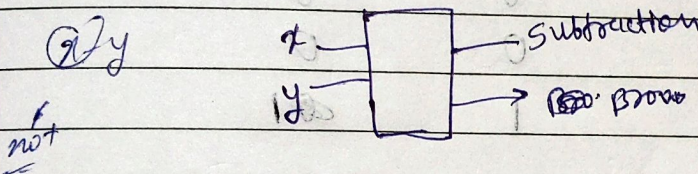
C_n
 $S_{n-1} \dots S_2 S_1 S_0$

all input OR and AND combine them max. amount for AND = $n+1$
OR = $n+1$

no. of stages = 4

Half Subtractor

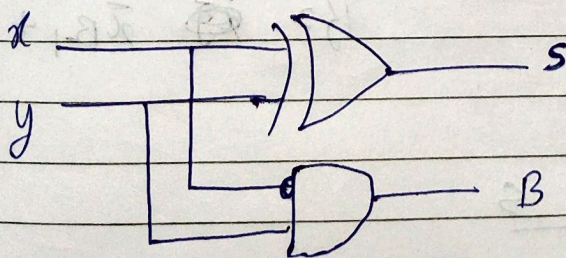
used to subtract two one bit num.



A	X	Y	S	B
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	1	0	0

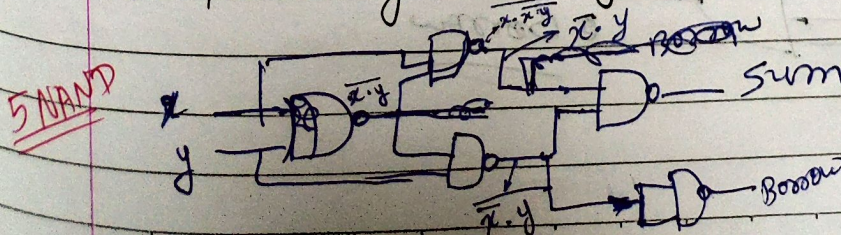
$$S = X \oplus Y$$

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



1 XOR
1 NOT
1 AND

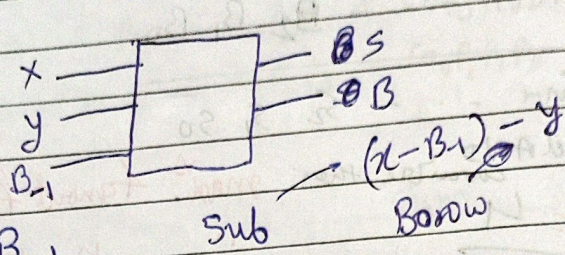
→ min how many NAND gate are needed to realise H.S. 89



$$\frac{X \cdot \bar{X} \cdot \bar{Y}}{X(\bar{X} + \bar{Y})}$$

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

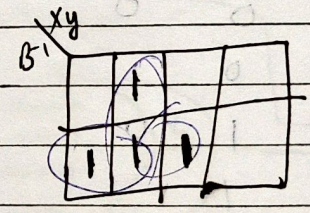
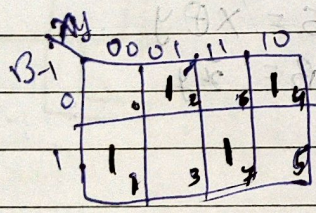
Full Subtractor:



X	y	B ₋₁	Sub	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

$$S = \sum m(1, 2, 4, 7)$$

$$B = \sum m(1, 2, 3, 7)$$

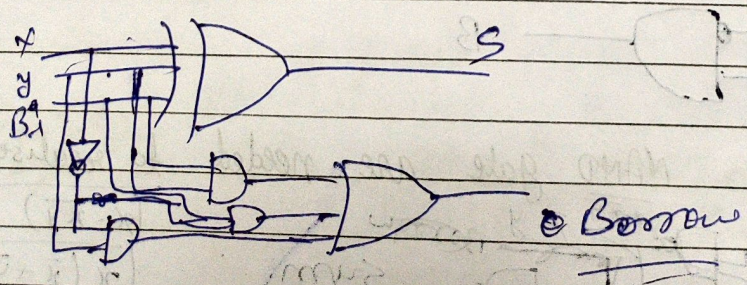


$$S = X \oplus y \oplus B_{-1}$$

$$\Downarrow$$

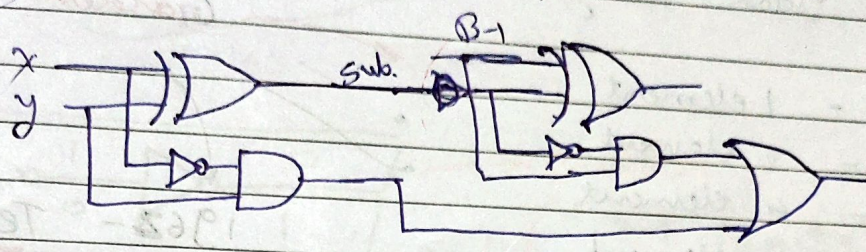
$$X \oplus y \oplus B_{-1}$$

$$B = \bar{x}B_{-1} + yB_{-1} + \bar{y}x$$



→ How many H.S. are needed else do ~~we~~ we need?? to realize f.s. & what

$(x \ y \ B)$



2 H.S. & one OR

CJ Sir

Digital Logic

no full form
JK Flip flop → made up on IC.

Jack Kilby

Gordon E Moore

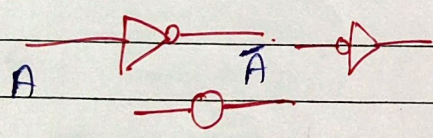
- 1958 - 1 element
- 1960 - 2 element
- 1962 - 4 element
- 1964 - 8 element
- 1968 - 16 element

1962 - Term binary
→ Intel

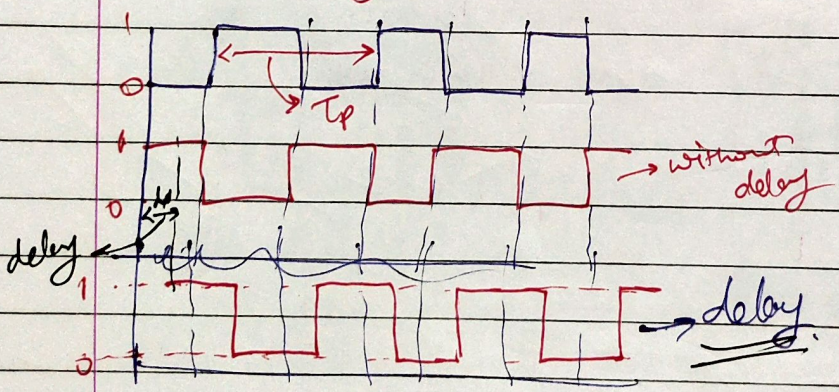
1965 :- statement

Number of element fabricated on a wafer will be doubled in every 18 to 24 months
~~single chip~~

① NOT Gate



inverter,
complement, logic,
negation



$$f = \frac{1}{T_p}$$

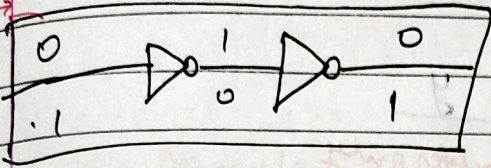
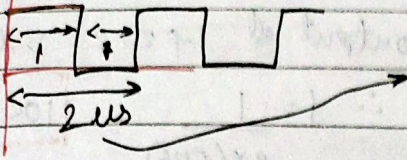
moore's law

→ number of element fabricated on a wafer will be doubled in every 18 to 24 months.

frequency = $\frac{1}{\text{Time period}}$

→ If Delay = 1 μs

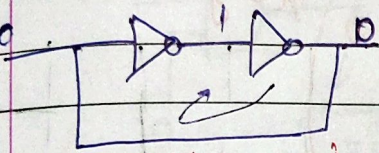
$f = \frac{1}{2 \mu s}$



Buffer.

input = output

used to provide delay.

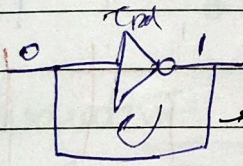


stick in loop

constant

$f = 0$

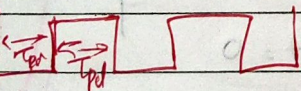
- Basic memory
- Bistable - multivibrator
- DC generator



change every time

0 → 1 → 0 → 0 → 1

if 3 NOT



$f = \frac{1}{2T_{pd}}$

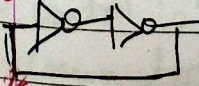
$f = \frac{1}{6T_{pd}}$

- ~~un~~stable multivibrator
- square wave generator
- clock generator
- free running ckt
- Ring oscillator

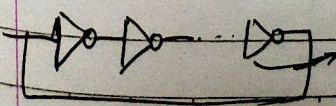
if delay = T_p

$f = \frac{1}{2T_p}$

even (not)



Basic memory, Bistable multivibrator
DC generator ($F = 0 \text{ Hz}$)

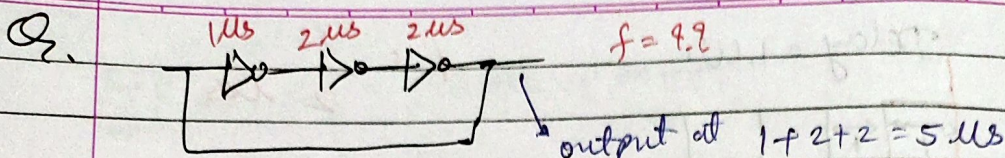


Astable multivibrator

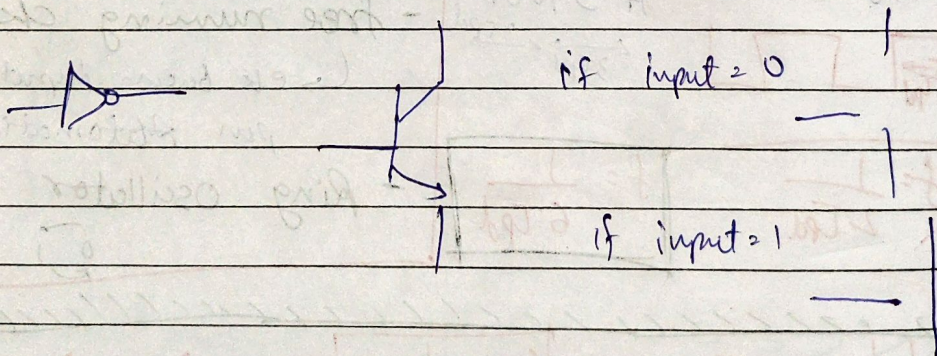
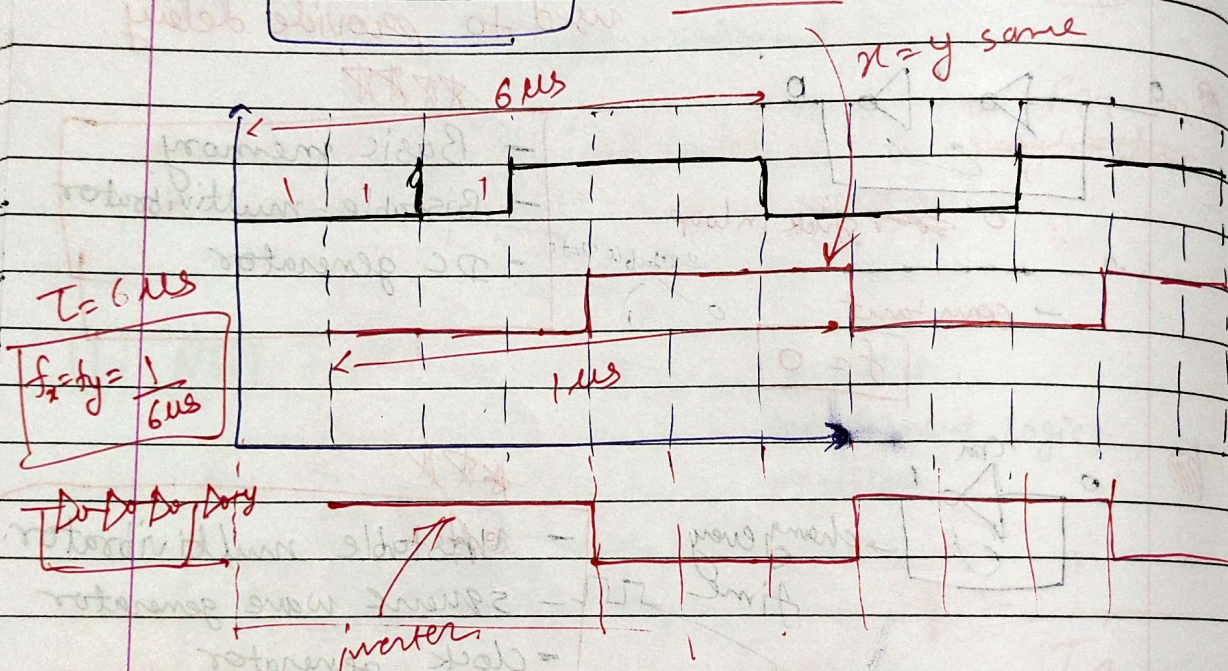
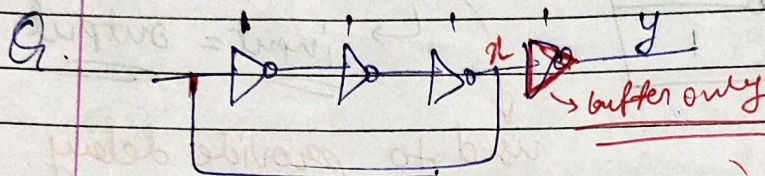
$F = \frac{1}{2nT_{pd}}$

square wave generator

frequency always in loop & equal at every node



$$\therefore f = \frac{1}{2 \times (5\mu s)} = \boxed{100 \text{ KHz}}$$

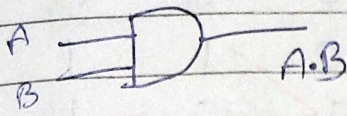


Associativity = $(A+B)+C = A+(B+C)$

Commutative = $A \cdot B = B \cdot A$

Any one input is zero output is 0

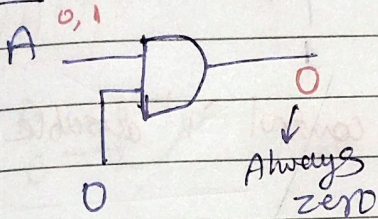
② AND



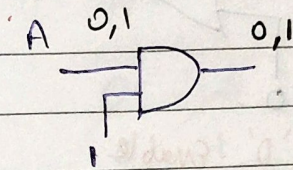
$$A \cdot 0 = 0$$

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

→ Enable / Disable

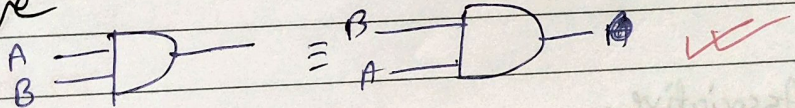


control '0' Disable

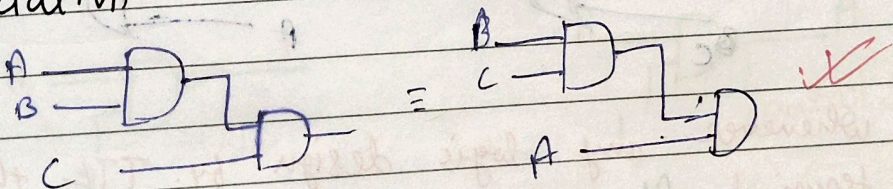


control '1' Enable

→ Commutative



→ Associative



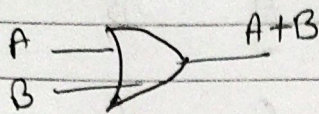
- $A \cdot 0 = 0$ (any one input zero o/p zero)
- control '0' disable, control '1' enable
- commutative, Associative follow

{ Any one input high }
output high

TTL $\rightarrow 1$
DTL $\rightarrow 0$

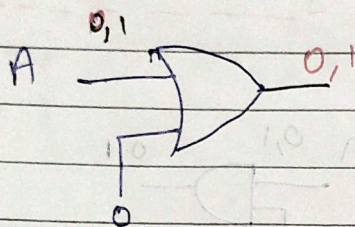
Page No.:	M T W T F S S
Date:	YOUVA

③ OR Gate

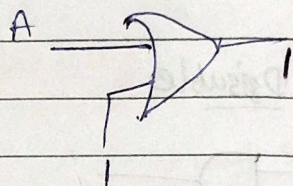


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

→ Enable/disable

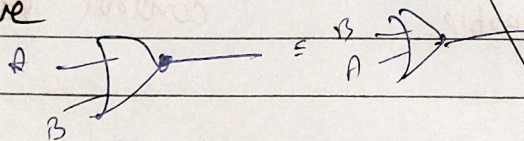


control '0' Enable

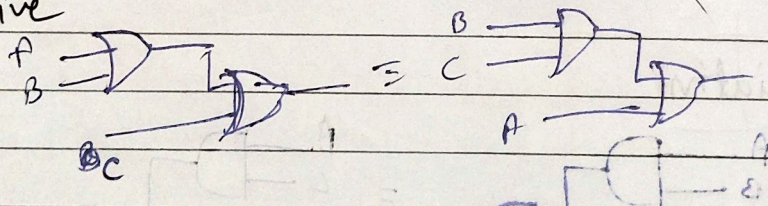


control '1' disable

- Commutative

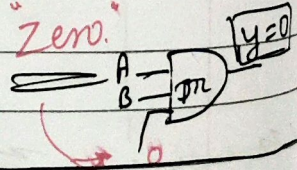
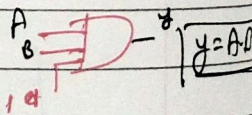


- Associative



TTL → whenever any logic design by TTL the floating terminal Always acts as high.

ECL → floating terminal :- always zero.

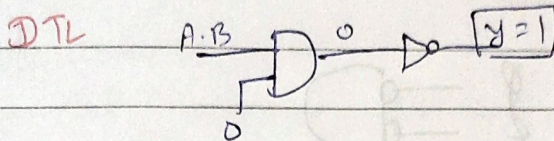
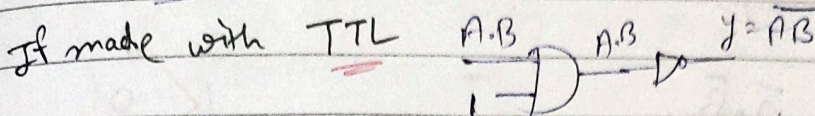
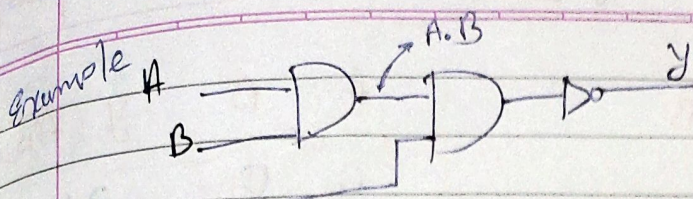


→ any one high → output high.

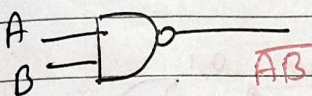
- control '1' disable, control '0' Enable

- Commutative as well as associative

NAND $\rightarrow$ Any one zero $\rightarrow$ o/p one



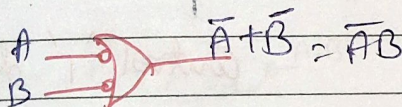
NAND



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

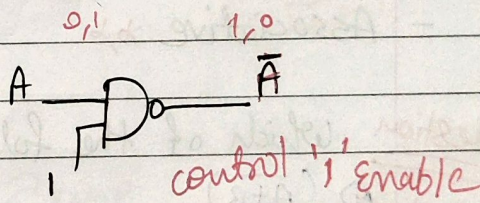
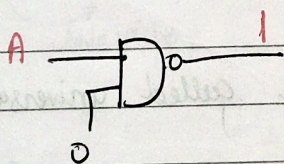
A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

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



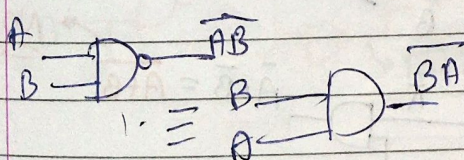
Bubbled or

enable / disable

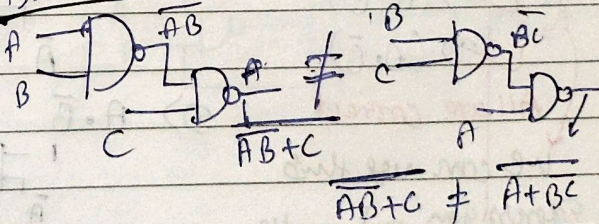


control '0' disable

- Commutative



Associative X



- Any input low / o/p high

- Bubbled or

- control '0' disable

- control '1' enable

- commutative

Associative X

MUX, Decoder, etc.

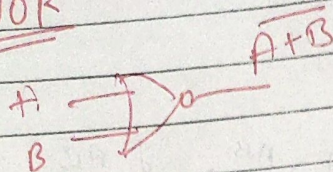
NAND, NOR are universal gate logic

Page No.:

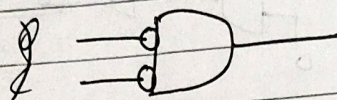
Date:

YOUVA

④ NOR



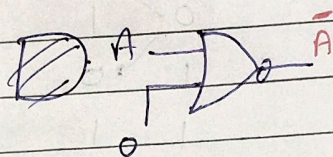
$$\overline{A+B} = \bar{A} \cdot \bar{B}$$



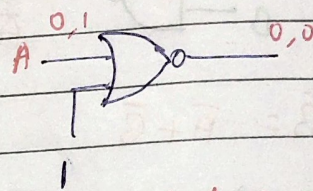
Bubbled-AND = NOR

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

- Enable / Disable



Control '0' Enable



Control '1' Disable

- Commutative ✓

- Associative ✗

Question Which of the following is/are called universal gates?

A) $\overline{A+B}$

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

C) $A+B$

D) $A \cdot B$

E) $\overline{A+B}$

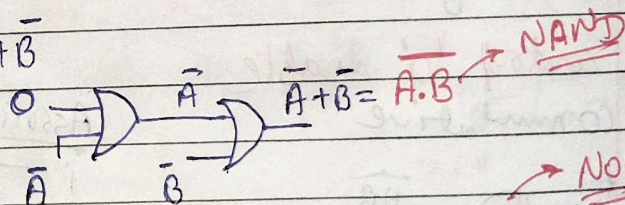
F) $\overline{A \cdot B}$

All are correct

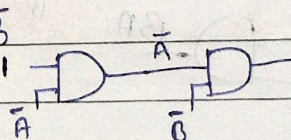
We can use this

expression and make

NAND/NOR then it will be universal



D) $A \cdot B$



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

- NOR { only any one input is high, output low }

- control '0' Enable, control '1' Disable

- Commutative ✓, Associative ✗

① $\overline{A \cdot B}$

② $\overline{A + B}$

3) $A + B$

4) $\overline{A} \cdot \overline{B}$

5) $\overline{A} + \overline{B}$

6) $A + \overline{B}$

7) $\overline{A} \cdot B$

8) $A \cdot \overline{B}$

9) MUX

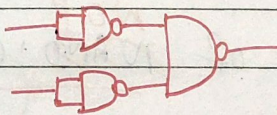
10) Decoder + OR logic

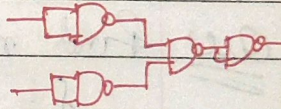
all are
universal
logic

NAND As universal gate

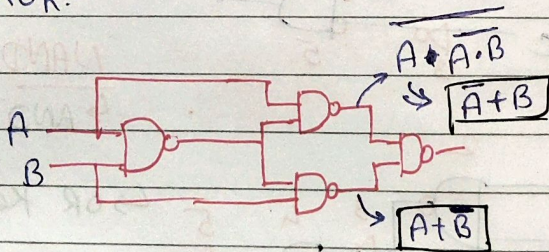
① NOT :- $A \rightarrow \overline{A}$

② AND :- $A \cdot B$

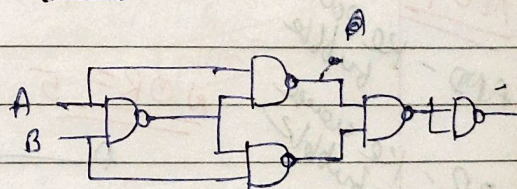
③ OR :- 
 2 bubble piche

④ NOR :- 

⑤ XOR :-



⑥ XNOR



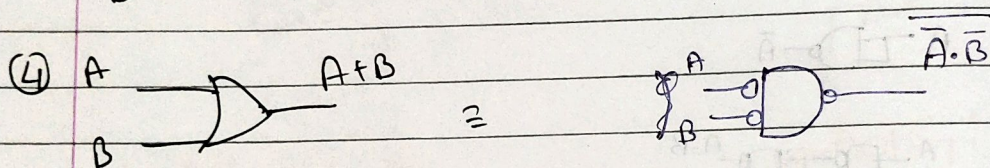
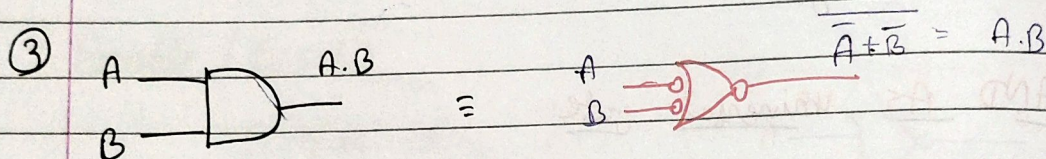
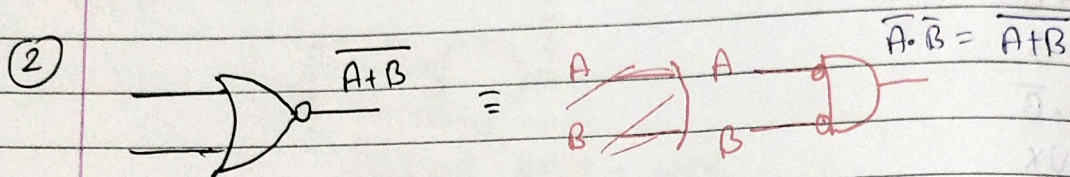
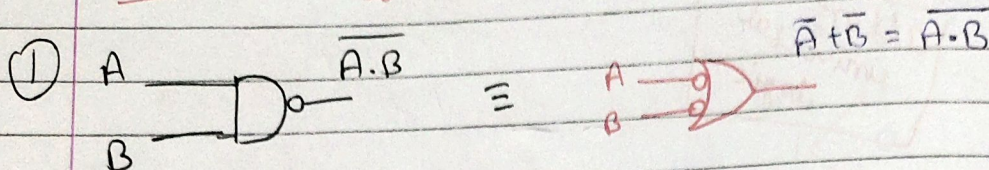
NAND $\rightarrow$ NOT - 1
 AND - 2
 OR - 3
 NOR - 4
 XOR - 4
 XNOR - 5

NOR $\rightarrow$ NOT - 1
 OR - 2
 AND - 3
 NAND - 4
 XNOR - 4
 XOR - 5

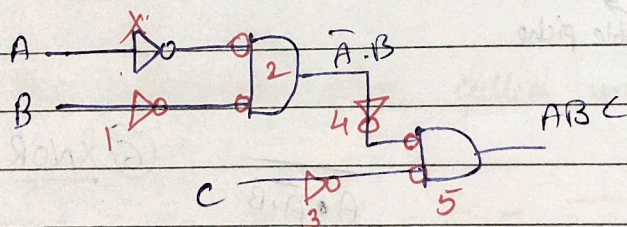


Minimization

Alternative symbol

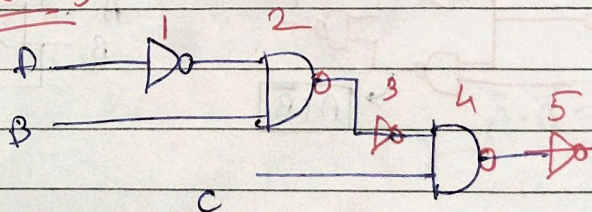


Ques Find minimum number of NOR NAND Gate
 $f = \overline{A}BC$



NOR
 AND - ke piche bubble
 OR - ke aage bubble

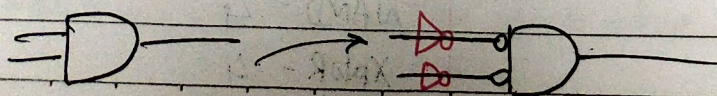
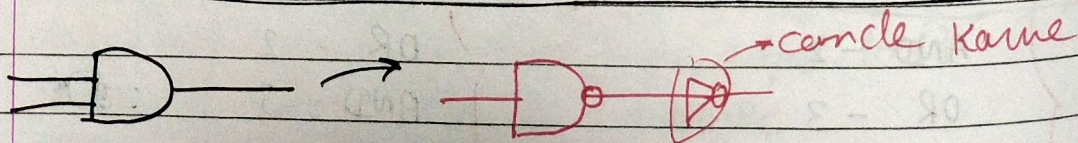
NOR = 5



NAND

AND ke aage bubble
 OR ke piche bubble

NAND = 5



cancel karte

Type :- 1

$\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D} \cdot E \dots$ all the variable use with $\cdot$ only

$n =$ Total variable

$K =$ complement variable ($\bar{A}, \bar{B}, \dots$)

minimum	NAND	NOR
	$(2n-2)+K$	$(3n-3)-3$

$f = \bar{A}BC$

$n=3, K=1$

NAND = $2 \times 2 + 1 = 5$

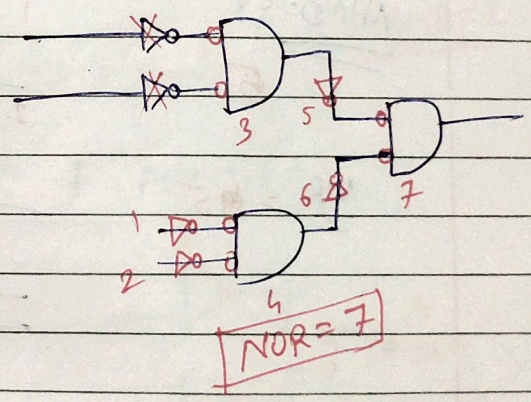
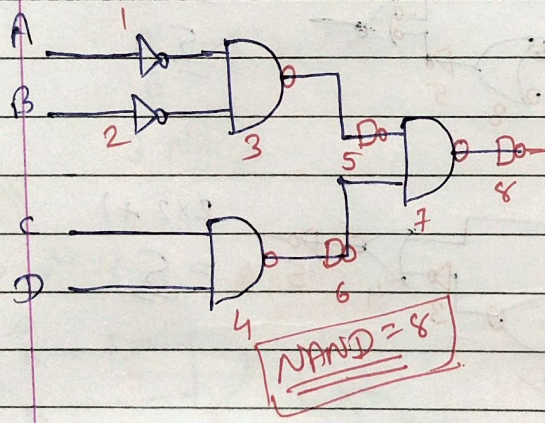
NOR = $3 \times 2 - 1 = 5$

$f = \bar{A} \cdot \bar{B} \cdot C \cdot D$

$n=4, K=2$

NAND = $2 \times 3 + 2 = 8$

NOR = $3 \times 3 - 2 = 7$



Type :- 1

$f = A \cdot B \cdot C \cdot \bar{D} \dots$

NAND = $(2n-2)+K$

NOR = $(3n-3)-K$

Type: 22

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

$$f = A(1 + A\bar{B}C + BCD)$$

$$f = A$$

0 NAND
0 NOR

Type: 2

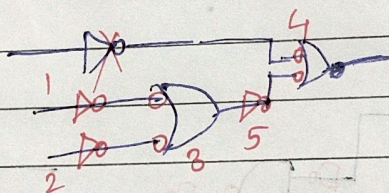
$$y = A + \bar{B} + C + \bar{D} + \dots$$

NAND	NOR
$(3n-3) - K$	$(2n-2) + K$

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

$$\text{NAND} = 9.9$$

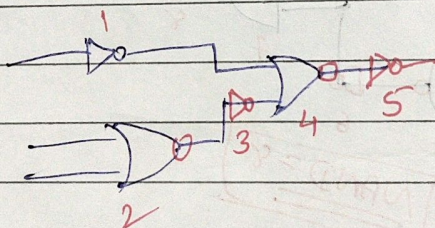
5



$$3 \times 2 - 1$$

$$= 5$$

$$\text{NOR} = 9.5$$



$$2 \times 2 + 1$$

$$= 5$$

$$f = A + B + \bar{C} + D + \bar{E} + \dots$$

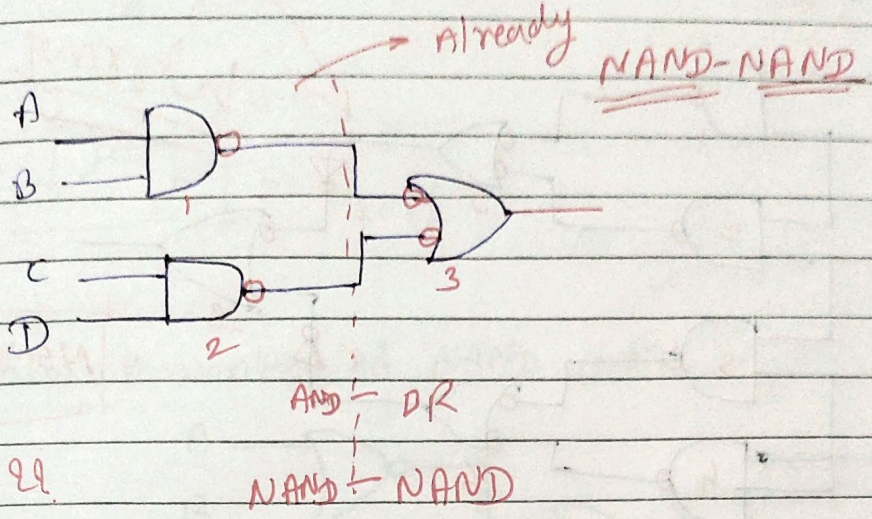
$$\text{NAND} = (3n-3) - K$$

$$\text{NOR} = (2n-2) + K$$

for min
 $\boxed{\text{NAND} \rightarrow \text{SOP}}$
 $\boxed{\text{NOR} \rightarrow \text{POS}}$

Type 3

$Y = AB + CD \rightarrow (\text{SOP})$

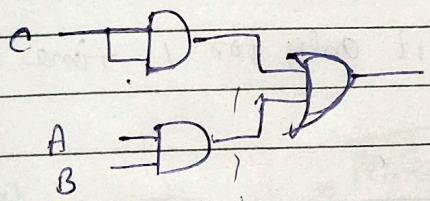


NAND = 3

3

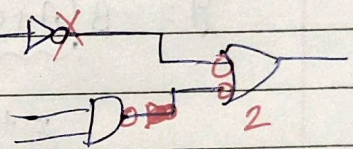
$\rightarrow AB + C$

$= AB + C \cdot C$



3 NAND

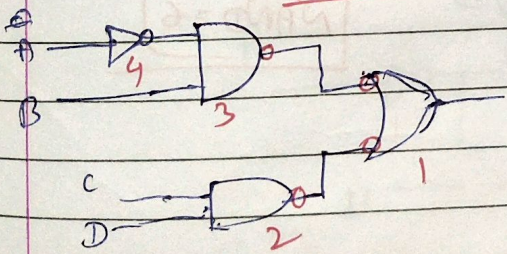
$\rightarrow \bar{A} + BC$



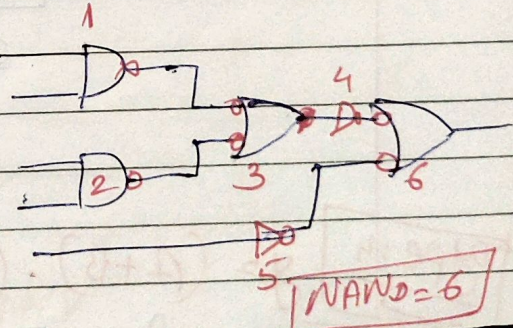
NAND = 2

$\rightarrow \bar{A}B + CD$

NAND = 4



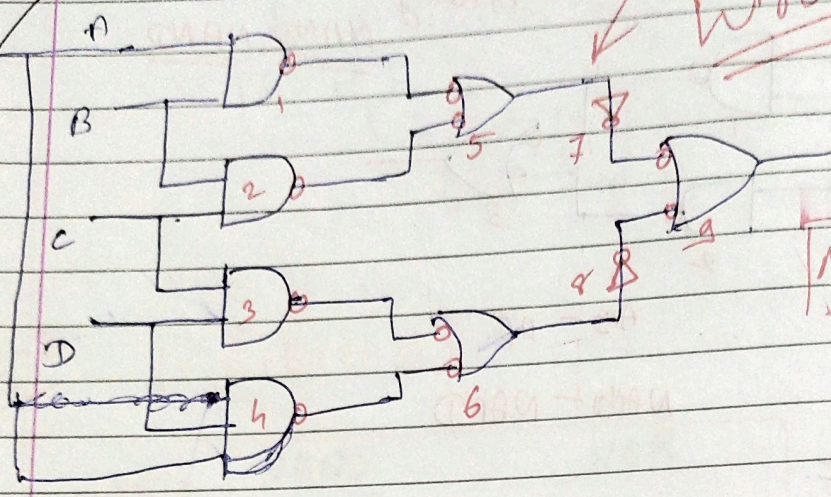
$\rightarrow f = AB + CD + E$



NAND = 6

Whenever in the problem minimum number of NAND Gates are asked then write the function in SOP form & implement it by using AOI which is AND-OR implementation and it is exactly equal to NAND-NAND Implementation

* $Q = y = AB + BC + CD + DA$

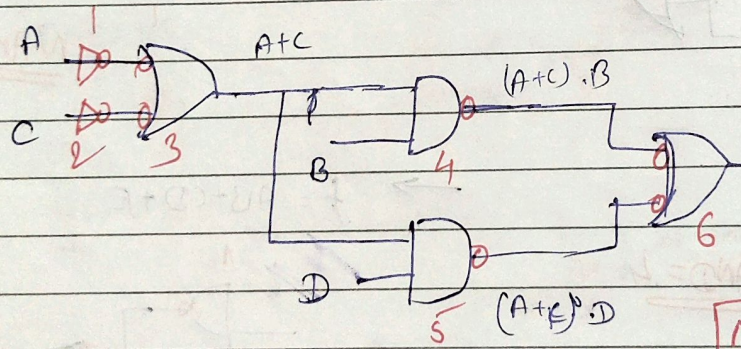


NAND = 9

→ simplify for

$y = B(A+C) + D(A+C)$

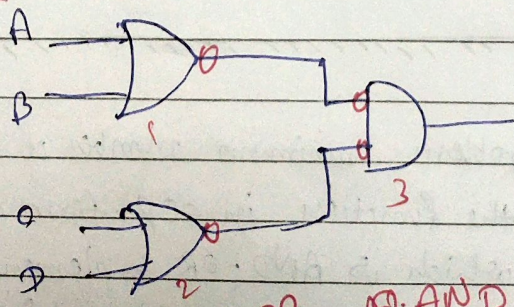
2-time → make it only for 1 time



NAND = 6

type: 4

$y = (A+B) \cdot (C+D)$



3 NOR

OR - AND
 NOR - NOR

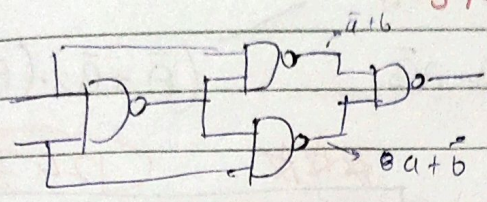
Type-5

$$A\bar{B} + B\bar{A} \rightarrow \text{XOR}$$

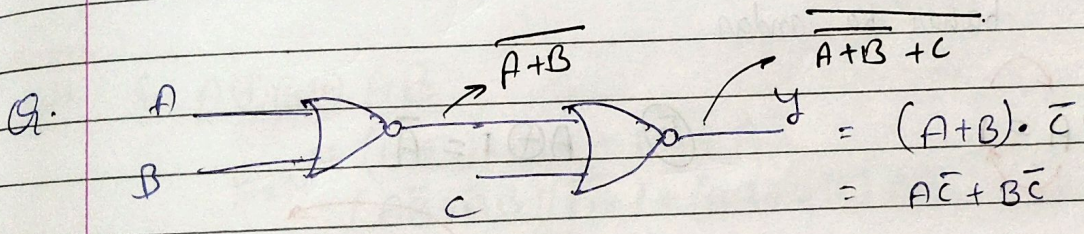
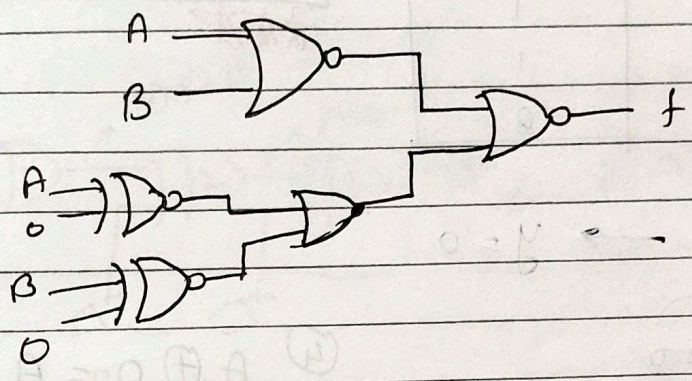
↓
5 NAND

$$A\bar{B} + AB \rightarrow \text{NOR}$$

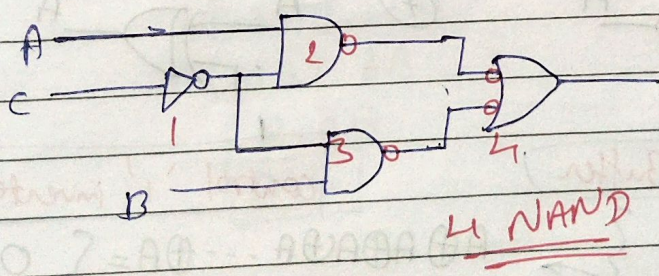
↓
5 NOR



que minimum number of NAND gate.



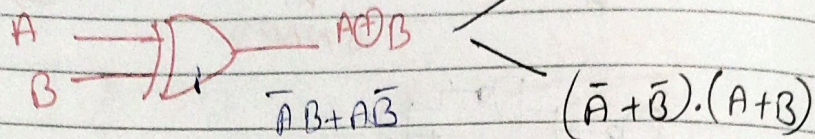
minimum 2-input NAND



- whenever in the problem minimum number of NOR gate are asked rewrite the function in POS form and implement it by AOI. which is OR-AND implementation and exactly equal to NOR-NOR Implementation

Buffer :- Just for Delay purpose
input & output are same

⑤ XOR gate



A	B	$y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

*** (247) ***
 $A \oplus B \oplus C = \sum m(1, 2, 4, 7)$
 $A \odot B \odot C = \sum m(1, 2, 4, 7)$

① $A = B \rightarrow y = 0$

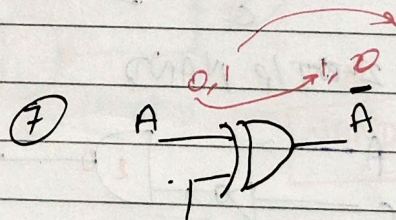
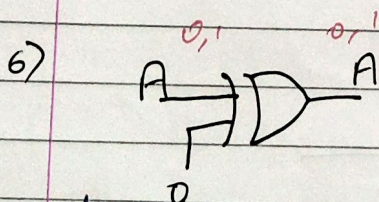
② $A \oplus A = 0$

④ $A \oplus 0 = A$

andar Ka bahar,
bahar Ka andar

③ $A \oplus \bar{A} = 1$

⑤ $A \oplus 1 = \bar{A}$



control '0' Buffer

control '1' inverter

$A \oplus A = 0$

$A \oplus A + A = A$

$A \oplus A \oplus A \oplus A = 0$

$A \oplus A \oplus A \oplus A \dots \oplus A = \begin{cases} 0 & \text{even} \\ A & \text{odd} \end{cases}$

* when odd number of input are 1 $\rightarrow$ output '1'

* control '0' Buffer

* control '1' inverter

$A \oplus A = 0$
 $A \oplus \bar{A} = 1$

$A \oplus A \oplus A \oplus A \dots \oplus A = \begin{cases} 0 & \text{even} \\ A & \text{odd} \end{cases}$

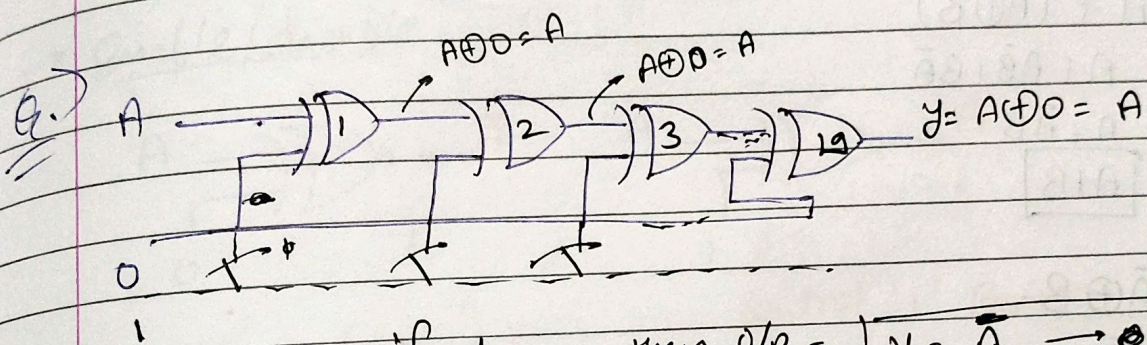
for odd input :- $XOR = XNOR$
 for even input :- $XOR = \overline{XNOR}$

→ Commutative As well As Associative

$$A \oplus A = 0$$

$$A \oplus \bar{A} = 1$$

→ odd parity detector
even parity generator



if 1 → then o/p =
 $y = \bar{A} \rightarrow \text{odd}$
 $y = 0 \rightarrow \text{even}$

Ques $f = A \oplus B \oplus \overline{AB}$

$$\begin{aligned}
 & (A \oplus B) \cdot \overline{AB} + A \oplus B \cdot AB \\
 & (A\bar{B} + \bar{A}B)(\overline{AB}) + (A\bar{B} + \bar{A}B)AB \\
 & (A\bar{B} + \bar{A}B)(\bar{A} + \bar{B}) + AB \\
 & A\bar{B} + \bar{A}B + AB \\
 & A(B + \bar{B}) + \bar{A}B \\
 & A + \bar{A}B \\
 & \boxed{A + B}
 \end{aligned}$$

$f = A \oplus B \oplus \overline{AB} \Rightarrow A \oplus B$

$f = A \oplus B \oplus AB = A \oplus B \star \star$

$f_2 = A \oplus B \oplus C \oplus ABC = ABC$

for ~~even~~ odd $A \oplus B \oplus C = A \odot B \odot C$

$$\rightarrow f = A \oplus B \oplus \overline{AB}$$

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

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

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

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

$$\overline{A}\overline{B} = \overline{AB} \quad \boxed{\overline{A+B}}$$

$$\rightarrow A + (A \oplus B)$$

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

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

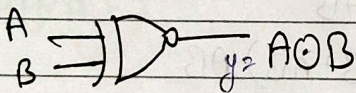
$$= \boxed{A+B}$$

$$\rightarrow \overline{A} \oplus B$$

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

$$= A \odot B$$

⑥ XNOR Gate



A	B	y
0	0	1
0	1	0
1	0	0
1	1	1

$$* A \oplus B \oplus \overline{AB} = \overline{A+B}$$

$$* A \oplus B \oplus C \oplus \overline{ABC} = \overline{A+B+C}$$

$$* \overline{A} \oplus B = A \oplus \overline{B} = \overline{A \odot B} = A \odot B$$

$$* \overline{A} \oplus \overline{B} = \overline{A} \odot \overline{B} = A \oplus B$$

XOR & XNOR follow commutative & as well as associative

Equal detector
Equivalence logic
Coincidence logic.

① $A=B \quad y=1$
 $A=\bar{B} \quad y=0$

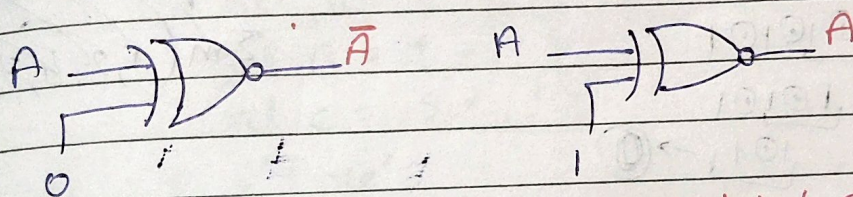
② $A \odot A = 1$

③ $A \odot 1 = A$

⑤ $A \odot \bar{A} = 0$

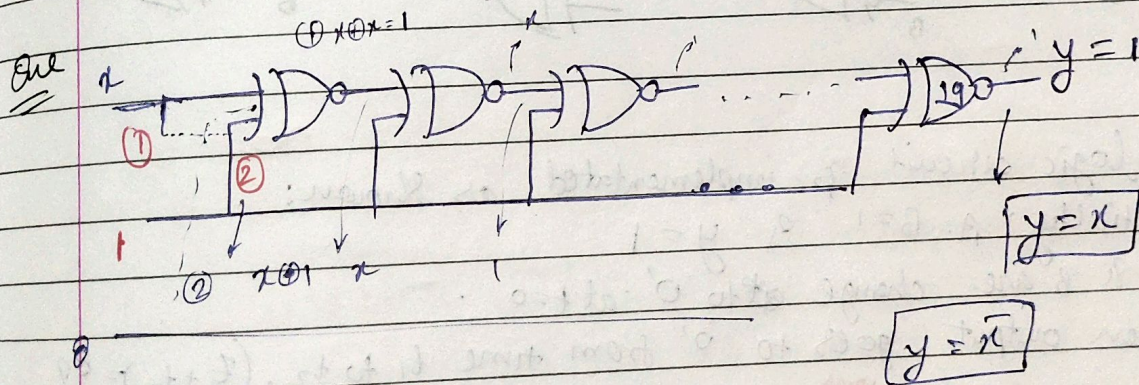
⑥ $A \odot 0 = \bar{A}$

→ Enable/disable



control '0' inverter

control '1' Buffer



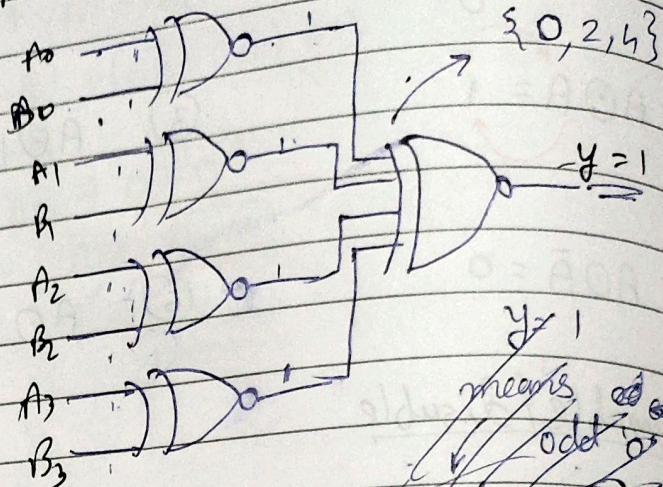
→ $A \odot A = 1$
 $A \odot A \odot A = \bar{A}$
 $A \odot A \odot A \odot A = 1$

$A \odot A \odot A \odot A \dots \odot A = \begin{cases} 1 & \text{even} \\ A & \text{odd} \end{cases}$

- ① * When odd input are "zero" → o/p high
- ② control '0' inverter
- ③ control '1' Buffer
- ④ $A \odot A \odot A \odot A \dots \odot A = \begin{cases} 1 & \text{even} \\ A & \text{odd} \end{cases}$

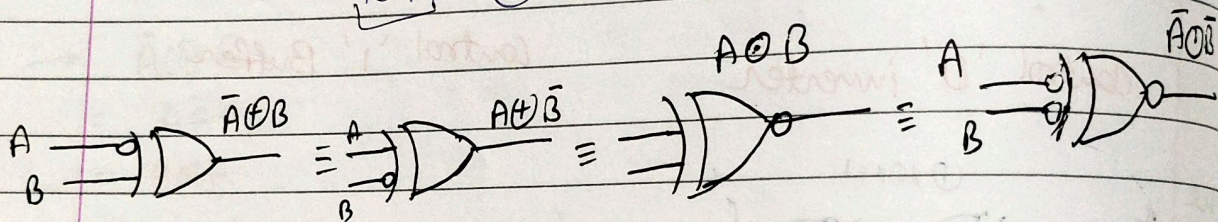
Que If the output $y=1$
 then correct input is/are

- ✓ A) 1111, 0000
- B) 1010, 0111
- ⊗ C) 0101, 0101
- ✓ D) 1100, 1110



$\Sigma m \{0, 2, 4\}$
 $y=1$
 means odd
 $\Sigma m \{1, 2, 4, 7, 8\}$

10101
 10101
 101 → 0



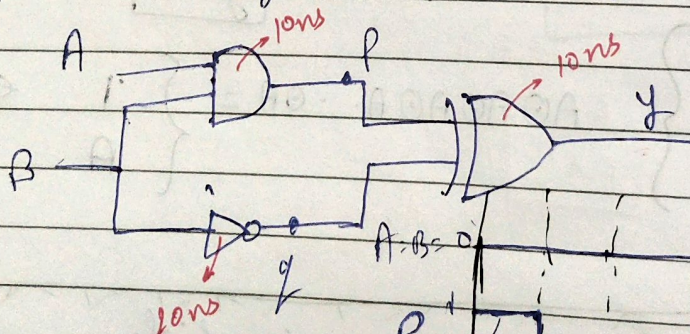
Que

A logic circuit is implemented as shown:

Initially $A=B=1$ & $y=1$

A & B are change to '0' at $t=0$

then output goes to '0' from time t_1 to t_2 . $(t_1+t_2)=98$

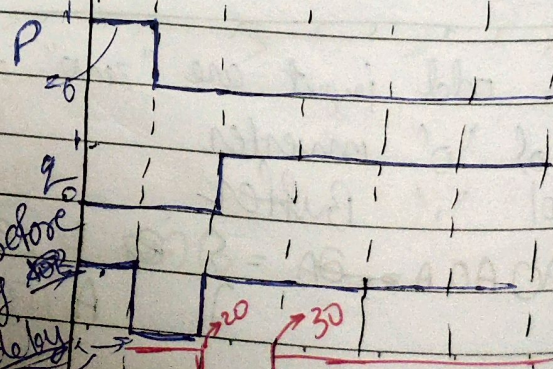


$t_1+t_2 = 30+10 = 50$

when $A=B=1$
 $P=1$
 $Q=1$

after change
 $A \& B$
 P change after
 10 ns

which is before
 y delay
 $\rightarrow$



XNOR $\xrightarrow{\text{and}}$ XOR

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

$$\rightarrow x_1 \oplus x_2 \oplus x_3 \oplus x_4 = 0$$

then

$$x_1 \oplus x_3 = x_2 \oplus x_4$$

complete function

$\rightarrow$ if all other Boolean functions can be constructed from this set and a set of input variable.

set A = $\{+, \times, '\}$ (OR, AND, complement)

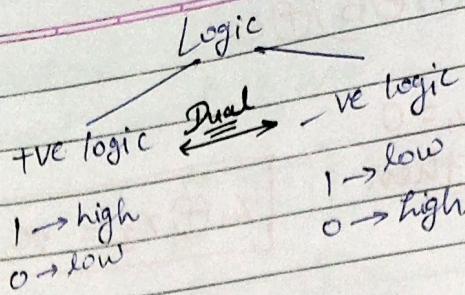
set B = $\{+, '\}$

set C = $\{\times, '\}$

ex. $x' + yz'$

☆

Dual



And

A	B	y
1	1	1
0	0	0
0	1	0
1	0	0
1	1	1

Dual

1' ko 0'
0' ko 1

A	B	y
1	1	1
1	0	1
0	1	1
0	0	0

$A+B \xrightarrow{\text{dual}} A \cdot B$

$1 \xrightarrow{\text{dual}} 0$

$A \xrightarrow{\text{dual}} A$

$\bar{A} \xrightarrow{\text{dual}} A$

$\text{AND} \longleftrightarrow \text{OR}$

$\bullet \xrightarrow{\text{dual}} +$

$\text{NAND} \xrightarrow{\text{dual}} \text{NOR}$

$\text{XOR} \longleftrightarrow \text{XNOR}$

$\text{Buffer} \longleftrightarrow \text{Buffer}$

$\text{inverter} \longleftrightarrow \text{inverter}$

Natural $\therefore$ # min term = # max

mutual Exclusive $\therefore$ (000, 111) (1, 1)
(001, 110) (1, 1)
(010, 101) (2, 2)

Self Dual $\therefore$ Natural

mutual exclusive $\rightarrow$ & not
 $f = \bar{A}\bar{B}C + A\bar{B}C + \bar{A}BC$

$\Rightarrow$ f is a function & f^D is Dual of f
then $\rightarrow$ f and f^D may be same
 $\rightarrow$ f and f^D may be complement
 $\rightarrow$ f and f^D may be different function

n Variable $f(x^n) \rightarrow 2^n \rightarrow$ Total distinct $f(x^n) = 2^n$

min term $\rightarrow 2^n$

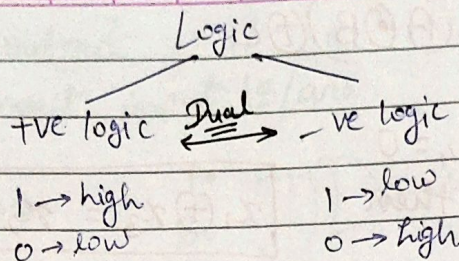
max term $\rightarrow 2^n$

$\frac{2^n}{2} = 2^{n-1}$
min term present

Total self dual $f(x^n) = 2^{n-1}$

Dual

Self Dual $\rightarrow f = f^D$



And

A	B	y	$\xleftrightarrow{\text{Dual}}$	A	B	y
0	0	0	'1' ko '0'	1	1	1
0	1	0	'0' ko '1'	1	0	1
1	0	0		0	1	1
1	1	1		0	0	0

$A+B \xleftrightarrow{\text{dual}} A \cdot B$

$1 \xleftrightarrow{\text{dual}} 0$

$A \xleftrightarrow{\text{dual}} A$

$\bar{A} \xleftrightarrow{\text{dual}} A$

$\text{AND} \xleftrightarrow{\text{dual}} \text{OR}$

$\bullet \xleftrightarrow{\text{dual}} +$

$\text{NAND} \xleftrightarrow{\text{dual}} \text{NOR}$

$\text{XOR} \xleftrightarrow{\text{dual}} \text{XNOR}$

$\text{Buffer} \xleftrightarrow{\text{dual}} \text{Buffer}$

$\text{inverter} \xleftrightarrow{\text{dual}} \text{inverter}$

Natural $\rightarrow$ # min term = # max term

mutual Exclusive $\rightarrow$ (000, 111) (1, 7)
(001, 110) (1, 6)
(010, 101) (2, 5)

Self dual $\rightarrow$ Natural

mutual exclusive $\rightarrow$ not mutually sym

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

$\Rightarrow$ f is a function & f^D is Dual of f
then $\rightarrow$ f and f^D may be same
 $\rightarrow$ f and f^D may be complement
 $\rightarrow$ f and f^D may be different function

n Variable $f(x^n) \rightarrow 2^n$ Total distinct $f(x^n) = 2^n$

min term $\rightarrow 2^n$ max term $\rightarrow 2^n$

$\frac{2^n}{2} = 2^{n-1}$ min term present

Total self dual $f(x^n) = 2^{n-1}$

Minimization

- By the minimization function terms and variables will minimize, so that less number of hardware required, circuit will be cost effective, delay will be less. and complexity is also less.

Boolean Algebra

K-map

Boolean function $\rightarrow$ combination of input on which output will depend

Function

SOP

POS

Sum of product

Product of sum

min term

max term

terms complement

$$\text{min term} = \overline{\text{max term}}$$

A	B	min term	max term	function (f)
0	0	$\bar{A} \cdot \bar{B}$	$A + B$	1
0	1	$\bar{A} \cdot B$	$A + \bar{B}$	0
1	0	$A \cdot \bar{B}$	$\bar{A} + B$	0
1	1	AB	$\bar{A} + \bar{B}$	1

Standard canonical $\rightarrow$ Each term

example
 $f = \bar{A}\bar{B} + AB$
 $f = \sum m(0, 4)$

$$f = \bar{A}\bar{B} \cdot 1 + \bar{A}B \cdot 0 + A\bar{B} \cdot 0 + AB \cdot 1$$

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

$$\sum m(0, 3)$$

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

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

$$\sum m(1, 2)$$

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

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

SOP POS $\rightarrow$ Age se

$\rightarrow$ number of minterm in standard & Canonical SOP

$$f(A,B,C) = A + BC$$

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

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

$$= \underset{\textcircled{1}}{ABC} + \underset{\textcircled{2}}{AB\bar{C}} + \underset{\textcircled{3}}{A\bar{B}C} + \underset{\textcircled{4}}{A\bar{B}\bar{C}} + \underset{\textcircled{5}}{\bar{A}BC}$$

Ans = 5

$\Rightarrow$ Literals

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

$\nearrow \uparrow \uparrow \uparrow \uparrow$
 1 2 3 4 5

not count because already we counted A

5

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

$\nearrow \nearrow \uparrow \uparrow \uparrow \uparrow$
 1 2 3 4 5 6

6

Literals :-

① Distribution Theorem

$$(A+B)(A+C)$$

$$\underline{\underline{A+BC}}$$

$$A+BC$$

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

$$A+BCD = (A+B)(A+C)(A+D)$$

② Redundancy Theorem

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

redundant term

yes 2 complement to each other

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

③ Transpose Theorem

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

$$A\bar{A} + AC + \bar{A}B + BC$$

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

$$\rightarrow (\bar{A} + \bar{B})(A+B) = \bar{A}B + A\bar{B} = A \oplus B$$

$$(\bar{A} + B)(A + \bar{B}) = \bar{A}\bar{B} + BA = A \odot B$$

④ Morgan's Law

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

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

- ① Distribution Theorem :- $A+BC = (A+B)(A+C)$
- ② Redundancy Theorem :- $AB + \bar{A}C + BC = AB + \bar{A}C$
↘ (redundant term)

③ Transpose Theorem :- $(A+B)(\bar{A}+C) = \underline{\underline{AC + \bar{A}B}}$

④ Morgan's Law

key

Duty cycle.

$$\frac{T_{ON}}{T_{ON} + T_{OFF}}$$

⑤ Absorption Theorem

$$A + AB = A$$

$$A(1+B)$$

⇒ minimize of the f_x

① A B y

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

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

$$= \bar{A}[\bar{B} + B] + AB$$

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

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

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

POS :- $\bar{A} + B$

② A B y

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

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

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

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

✶ if we have n variable
no. of distinct f_x

$$2^{2^4} = 2^{16} = 2^6 \cdot 2^{10} = \underline{\underline{64 \times 1024}}$$

Hamming distance :- ~~2 bit~~ ~~successive numbers~~ ke bich jitna distance ho in Gray code :- Hamming distance = 1 bit

M T W T F S S						
Page No.:				YOUVA		
Date:						

⇒ Distingu⁺ function

for 1 variable :- $\{A, \bar{A}, 0, 1\}_4$

for 2 variable $\{A, B\}$:-

$\bar{A}\bar{B}$	$A+B$	$\bar{A}$	0
$\bar{A}B$	$A+\bar{B}$	A	1
$A\bar{B}$	$\bar{A}+B$	$\bar{B}$	$\bar{A}B+A\bar{B}$
AB	$\bar{A}+\bar{B}$	B	$AB+A\bar{B}$

↓ minterm
↓ max term

} 16

for n variable $\left\{ 2^{2^n} \right\}$ distinct function

2^n minterm

2^n max term

⊙ K-map

↳ Based on Gray Code

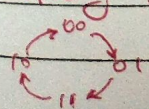
$A \backslash B$	0	1
0	00	01
1	10	11

$A \backslash B$	00	01	11	10
0				
1				

⇒ Bada se Bada group & Kamse Kam Group

⇒ no. of distinct $f_x^n = 2^{2^n}$

⇒ Gray code :- successive numbers are differ by 1 bit



↳ unity hamming distance code.

↳ cyclic code

↳ reflecting code



A	BC			
	1	0		
			0	1

Que

	$\bar{A}$	A
$\bar{B}$	1	X
B	0	X

$C = 0$

$C = 1$

	$\bar{A}$	A
$\bar{B}$	1	X
B		X

Q.

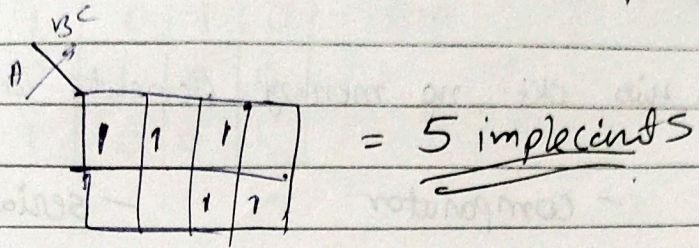
	$\bar{A}$	A
$\bar{B}$	1	X
B	1	X

$\bar{B}\bar{C}$

$$\therefore \boxed{\bar{B}\bar{C} + C = \bar{B} + C}$$

Implicants :- minterms in canonical form are called

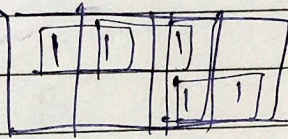
↳ no. of 1 in k-map are called implicant.



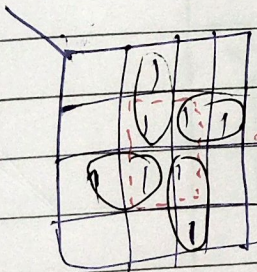
Prime Implicant

:- Total number of all possibilities of formation Group.

↳ Bada group ban gaya toh chote nhi banene.



4 PI



5 PI

Combinational ckt

→ This is the ckt in which output will depend on present input; hence it is also called static ckt.

→ In this ckt, no memory elements are present.

- Comparator

- MUX

- DEMUX

- Encoder

- Decoder

- HA

- FA

- H.S

- F.S

- serial adder

- parallel adder {Ripple carry adder

- Look ahead carry adder

- Multiplier

- 2's complement adder/subtractor

- Code-converter

- Designing of new ckt.

Designing step

① Find no. of input & outputs

② Write the truth Table.

③ Write the logical Expression.

④ ~~minimize~~ minimize the logical Expression.

⑤ Hardware implementation.

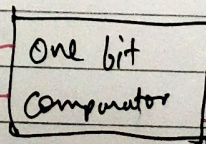
① magnitude Comparator

Q: design 1 bit comparator

step 1:-

A

B



$A > B (x)$

$A < B (y)$

$A = B (z)$

Step:-2

		A > B	A < B	A = B
A	B	X	Y	Z
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

B > A

Step:3

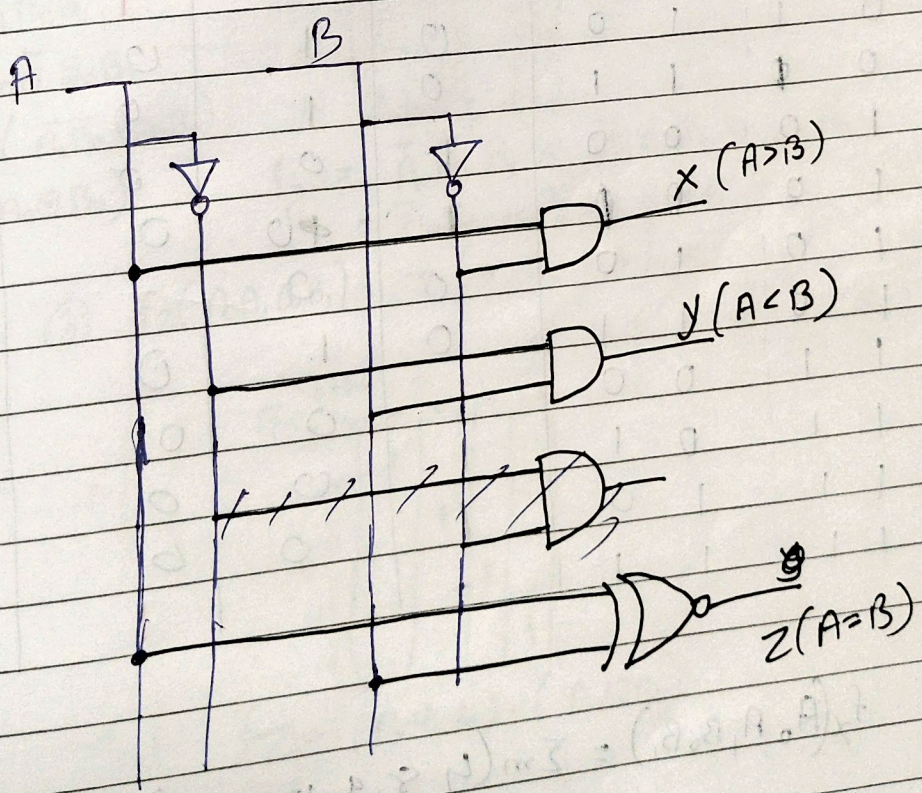
$$f_x(A,B) = A \cdot \bar{B}$$

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

$$f_z(A,B) = A \odot B$$

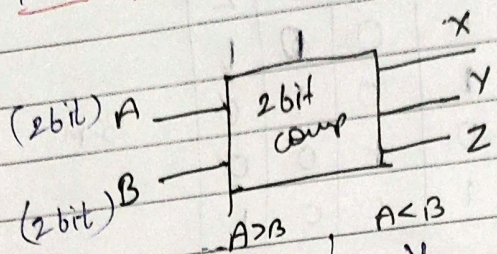
Step:4 already minimized.

Step:5



Q Design Two bit comparator

Step 1:



Step 2

				A > B X	A < B Y	A = B Z
A		B				
A ₀	A ₁	B ₀	B ₁			
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

Step 3

$$f_x(A_0, A_1, B_0, B_1) = \sum m(4, 8, 9, 12, 13, 14) \quad (A > B)$$

$$f_y(A_0, A_1, B_0, B_1) = \sum m(1, 2, 3, 6, 7, 11) \quad (A < B)$$

$$f_z(A_0, A_1, B_0, B_1) = \sum m(0, 5, 10, 15) \quad (A = B)$$

Step: 4 minimization

① $f_x(A_0A_1, B_0B_1) = \sum m(4, 8, 9, 12, 13, 14)$

semi minimized

$A_0A_1 \backslash B_0B_1$	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	1	1	1	1
10	1	1	1	0

$= A_0\bar{B}_0 + \bar{A}_0A_1\bar{B}_0\bar{B}_1$
 $+ A_0A_1B_0\bar{B}_1$

$= A_0\bar{B}_0 + A_1\bar{B}_1(\bar{A}_0\bar{B}_0 + A_0B_0)$
 $= A_0\bar{B}_0 + A_1\bar{B}_1(A_0 \odot B_0)$

② $f_y(A_0A_1, B_0B_1) = \sum m(1, 2, 3, 6, 7, 11)$

$A_0A_1 \backslash B_0B_1$	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	1	1	1	1
10	1	1	1	0

$f_y = B_0\bar{A}_0 + \bar{A}_0\bar{A}_1\bar{B}_0B_1$
 $+ A_0\bar{A}_1B_0B_1$

$= B_0\bar{A}_0 + \bar{A}_1B_1(\bar{A}_0\bar{B}_0 + A_0B_0)$
 $= B_0\bar{A}_0 + \bar{A}_1B_1(A_0 \odot B_0)$

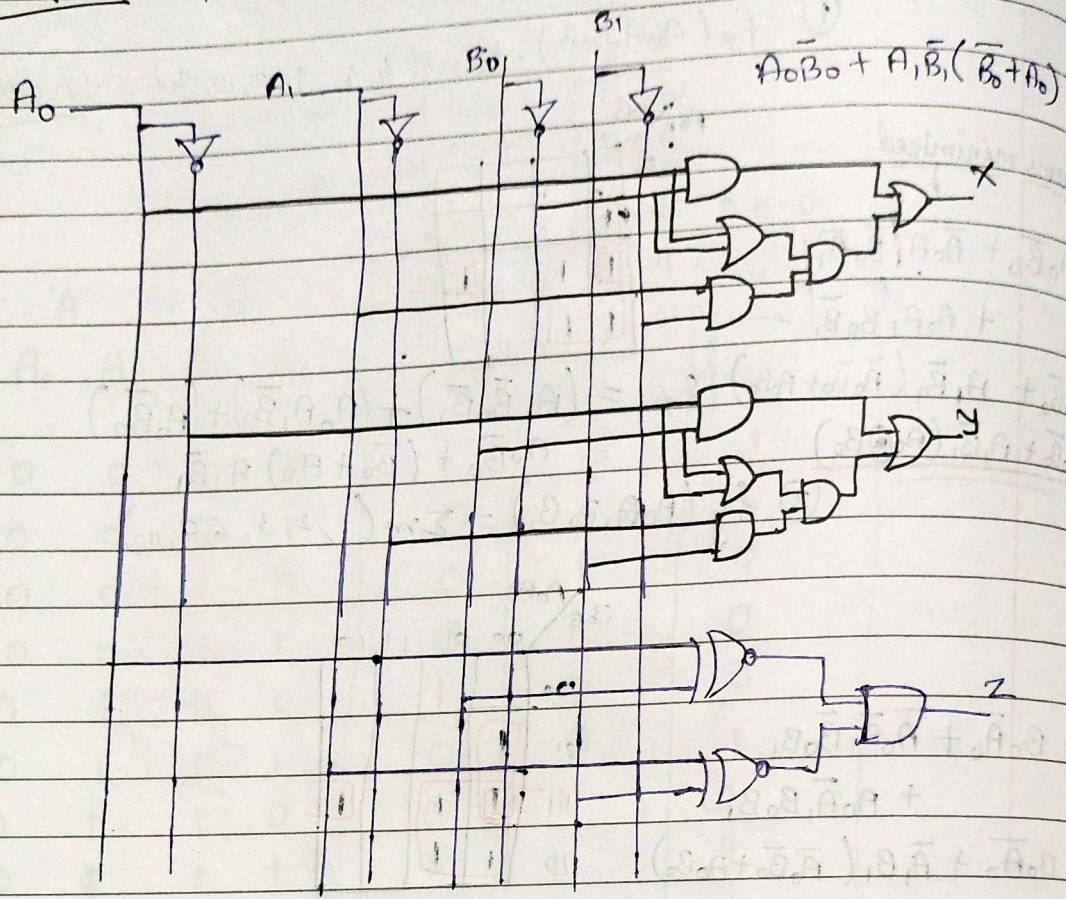
$f_y = \bar{A}_0\bar{A}_1B_1 + \bar{A}_1B_0B_1 + B_0\bar{A}_0$
 $= \bar{A}_0B_0 + \bar{A}_1B_1(B_0 + A_0)$

③ $f_z(A_0A_1, B_0B_1) = \sum (0, 5, 10, 15)$

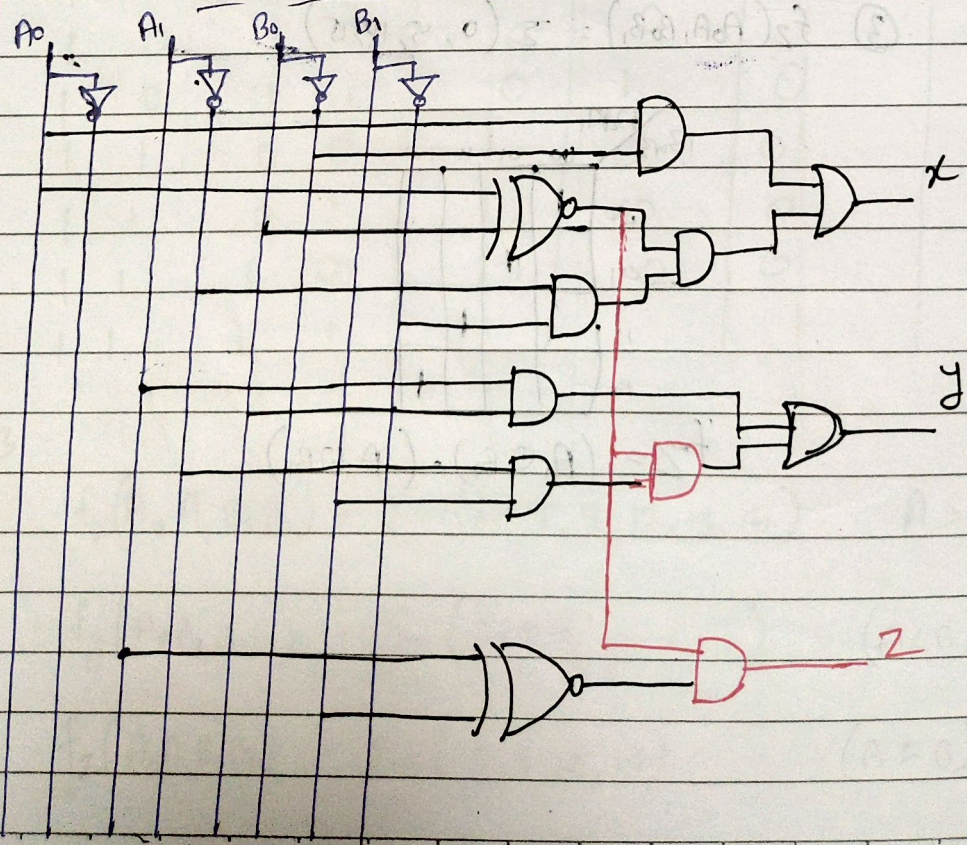
$A_0A_1 \backslash B_0B_1$	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$f_z = (A_0 \odot B_0) \cdot (A_1 \odot B_1)$

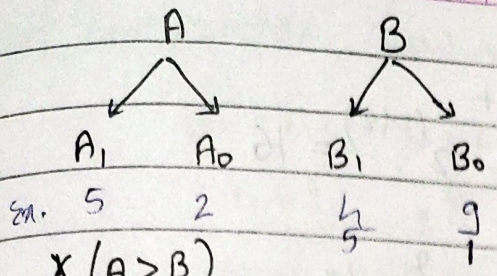
Step 5:- Hardware implementation (minimized)



Step 5:- Semi-minimized



Short Logic
2 bit



{ pehle hamne 5 or 4 compare kiya.

$$= A_1 \bar{B}_1 + (A_1 \odot B_1) A_0 \bar{B}_0$$

{ pehle wale barabar ho gaye. And next wale check kiye }

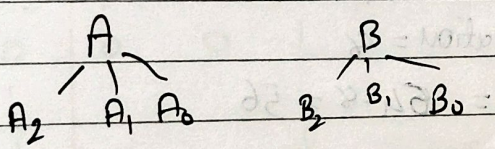
$A > B$ ke liye B chhota he.

$$Y(A < B) = \bar{A}_1 B_1 + (A_1 \odot B_1) \bar{A}_0 B_0$$

logic :- { pehle A, B compare kiya jo chhota uske uper ban or pehle do same then next bit me jao }

$$Z(A = B) = (A_1 \odot B_1) (A_0 \odot B_0)$$

3 bit



$$X(A > B) = A_2 \bar{B}_2 + (A_2 \odot B_2) A_1 \bar{B}_1 + (A_2 \odot B_2) (A_1 \odot B_1) A_0 \bar{B}_0$$

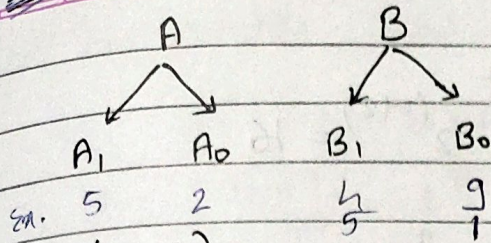
$$Y(A < B) = \bar{A}_2 B_2 + (A_2 \odot B_2) \bar{A}_1 B_1 + (A_2 \odot B_2) (A_1 \odot B_1) \bar{A}_0 B_0$$

$$Z(A = B) = (A_2 \odot B_2) (A_1 \odot B_1) (A_0 \odot B_0)$$

Chhota ke uper baar and Equal Ki condition XOR Karta hai.

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

2 bit



{ pihale hamne 5 or 4 compare kiya.

$$X(A > B) = A_1 \bar{B}_1 + (A_1 \odot B_1) A_0 \bar{B}_0$$

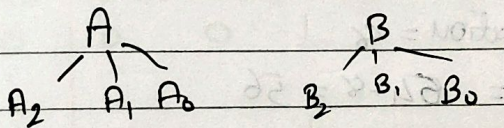
{ pihale wale barabar ho gaye. And next wale check kiye }

$A > B$ ke liye B chhota he.

$$Y(A < B) = \bar{A}_1 B_1 + (A_1 \odot B_1) \bar{A}_0 B_0$$

logic :- { pihale A, B_1 compare kiya jo chhota uske uper barabar or pihale do same then next bit me jao }

$$Z(A = B) = (A_1 \odot B_1) (A_0 \odot B_0)$$



$$X(A > B) = A_2 \bar{B}_2 + (A_2 \odot B_2) A_1 \bar{B}_1 + (A_2 \odot B_2) (A_1 \odot B_1) A_0 \bar{B}_0$$

$$Y(A < B) = \bar{A}_2 B_2 + (A_2 \odot B_2) \bar{A}_1 B_1 + (A_2 \odot B_2) (A_1 \odot B_1) \bar{A}_0 B_0$$

$$Z(A = B) = (A_2 \odot B_2) (A_1 \odot B_1) (A_0 \odot B_0)$$

3 bit

2 bit

$$A = 2 \text{ bit} \quad B = 2 \text{ bit}$$

$$\text{Total combination} = 2^{(2+2)} = 16$$

$$\text{Equal combination} = 2^2 = 4$$

$$\text{unequal} \quad 16 - 4 = 12$$

$$(A > B) = \frac{12}{2} = 6$$

$$A < B \text{ combination} = \frac{12}{2} = 6$$

3 bit

$$A = 3 \text{ bit} \quad B = 3 \text{ bit}$$

$$\text{Total combination} = 2^6 = 64$$

$$\text{Equal combination} = 8$$

$$\text{unequal} = 64 - 8 = 56$$

$$A > B \text{ combination} = 28$$

$$A < B \text{ combination} = 28$$

n bit comparator

$$A = n \text{ bit} \quad B = n \text{ bit}$$

$$\text{Total combination} = 2^{2n}$$

$$\text{Equal combination} = 2^n$$

$$\text{unequal combination} = 2^{2n} - 2^n$$

$$A > B \text{ combination} = \frac{2^{2n} - 2^n}{2}$$

$$A < B \text{ combination} = \frac{2^{2n} - 2^n}{2}$$

Q. 4 bit comparator, how many condition for $A > B$?

$A = 4 \text{ bit}$ $B = 4 \text{ bit}$

Total = 2^8

Equal = 2^4

$\therefore \text{unequal} = 256 - 16 = 240$

$\therefore A > B = 120$

Q. $A = 3 \text{ bits}$.

$B = 2 \text{ bits}$

Total = $2^5 = 32$

Equal = 4

unequal = 28

	A_2	A_1	A_0	B_1	B_0	
	0	0	0	1		} Equal = 4 unequal = $16 - 4 = 12$ $A > B = 6$ $A < B = 6$
	0	0	0	0		
	0	0	0	0		
	0	0	0	0		
	0	1	1			} 16 ($A > B$)
	0	1	0			
	0	0	0			
	0	0	0			

Total = $16 + 6 = 22$ ($A > B$)
 6 ($A < B$)

Must do every time

H.W

① $A = 5$ bit, $B = 3$ bit

$$\text{Total bit} = 2^{(5+3)} = 256$$

Equal

$$(A > B) =$$

$$\text{Total combination} = 99$$

$$\text{Equal combination} = 99$$

$$\text{unequal combination} = 99$$

$$A > B \text{ combination} = 99$$

$$A < B \text{ combination} = 99$$

$A_0 A_1 A_2 A_3 A_4$ $B_0 B_1 B_2$

0 0 0 0 0 0 0 0 0

0 0 1 1 1 1 1 1 1

0 1 0 0 0

0 1 1 1 1

1 1 1 1

3 bit

$$a = b \rightarrow 8$$

$$64 \text{ not equal} \rightarrow 64 - 8 =$$

$$a > b = 28$$

$$a < b = 28$$

$$(256 - 64) \text{ always}$$

$$a > b$$

$$\text{Total combination} = 256$$

$$a = b \Rightarrow 8$$

$$\text{not equal} = 256 - 8 = 248$$

$$a > b = 192 + 28$$

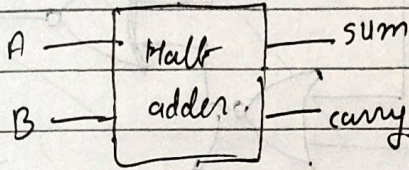
$$a > b = 220$$

$$a < b = 28$$

(2) Half adder

↳ 2 bit adder is known as half adder

Step 1:-



Step 2:-

A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Step 3:-

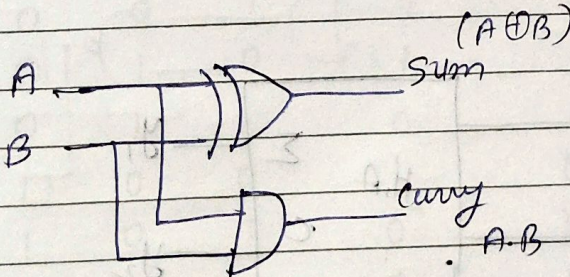
$$\text{sum} = \bar{A}B + A\bar{B} = A \oplus B$$

$$\text{carry} = AB$$

Step 4:-

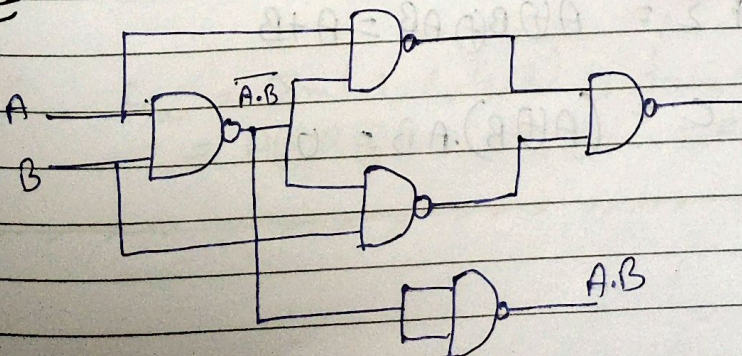
~~the above~~ already minimized

Step 5:-

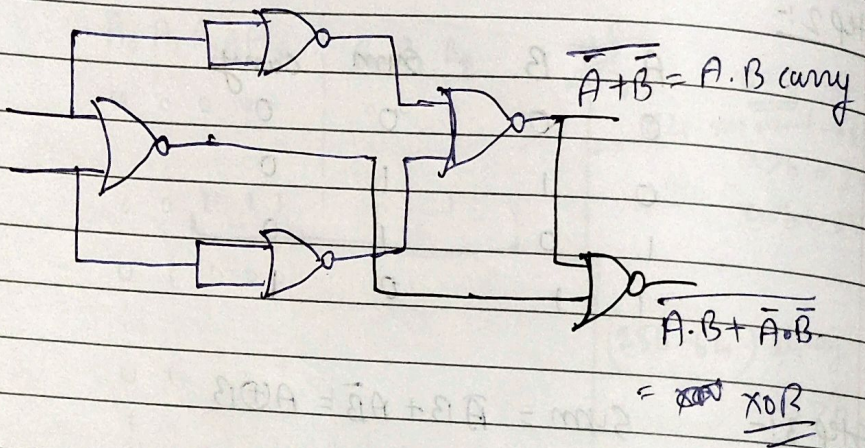
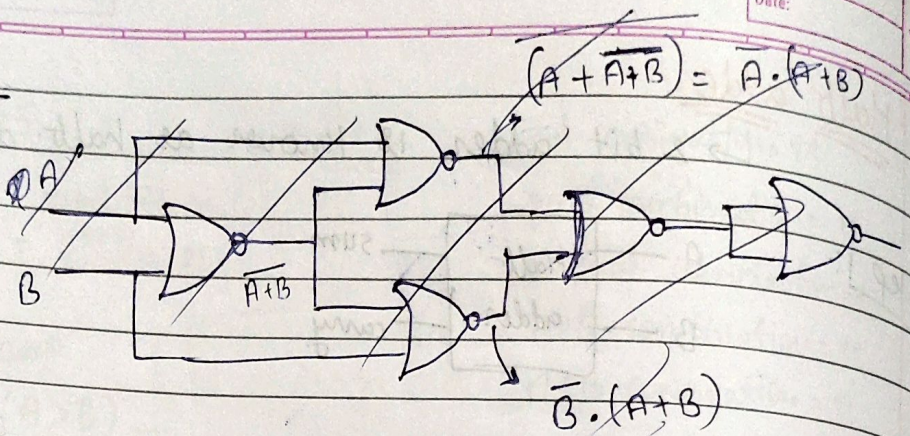


5 NAND

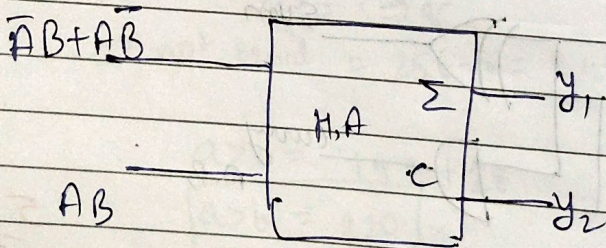
using NAND



using NOR



Q. What is the value of y_1 and y_2



$$\Sigma = A \oplus B \oplus AB = A + B$$

$$C = (A \oplus B) \cdot AB = 0$$

$$1247 = A \oplus B \oplus C$$

no. carry majority high logic

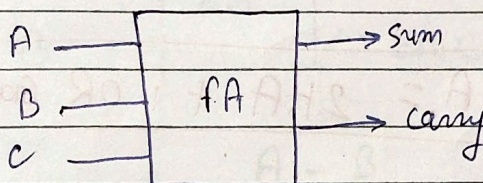
③ full adder

L> 3 bit adder are known as full adder.

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

carry sum

step 1:-



step 2:- Truth table

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

$$\begin{aligned}
 &= \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} \\
 &= (\bar{A}B + A\bar{B}) \cdot C + AB(C + \bar{C}) \\
 &= (A \oplus B)C + AB
 \end{aligned}$$

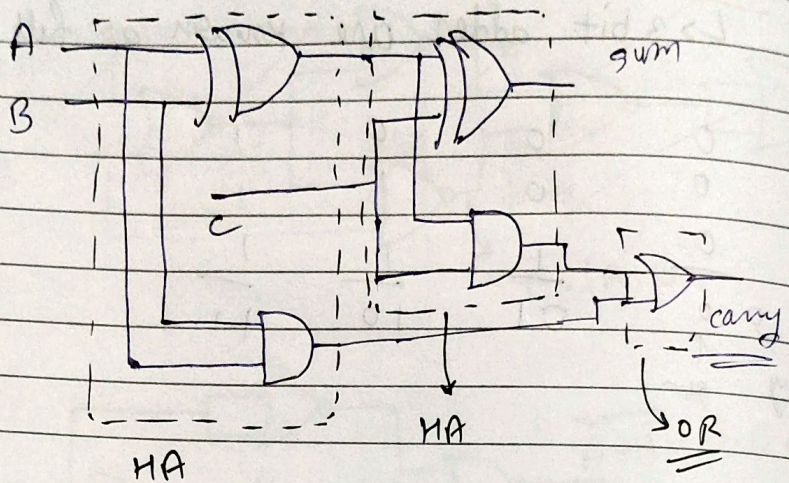
semi minimized

step 3:-

$$\begin{aligned}
 f_{\text{sum}} &= \sum m(1, 2, 4, 7) ; f_{\text{carry}} = \sum m(3, 5, 6, 7) \\
 &= A \oplus B \oplus C = AB + BC + CA
 \end{aligned}$$

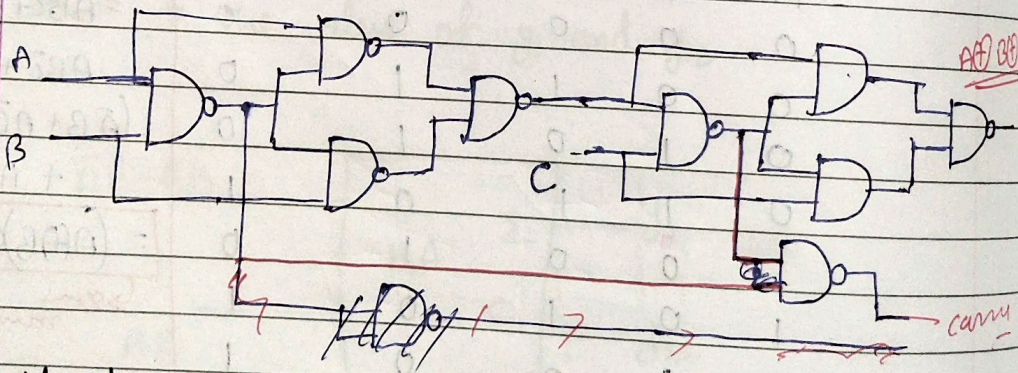
majority high logic

step 5



$$FA = 2HA + 1 \text{ OR gate}$$

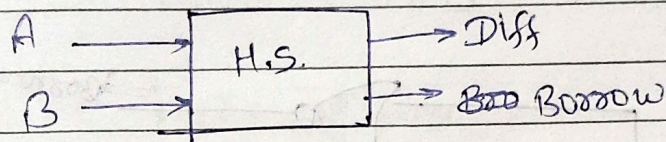
full adder by using NAND gate $\Rightarrow$ 9 NAND



full adder by using NOR Gate

Halt Adder Subtractor

Step 1:-



Step 2:-

A - B			
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Step 3:-

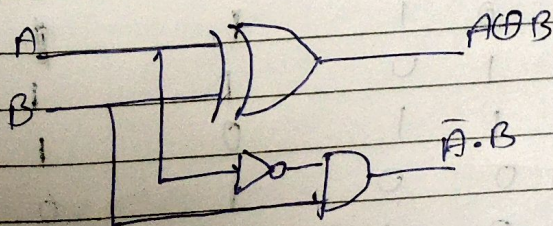
$$\text{Diff} = A \oplus B$$

$$\text{Borrow} = \bar{A} \cdot B$$

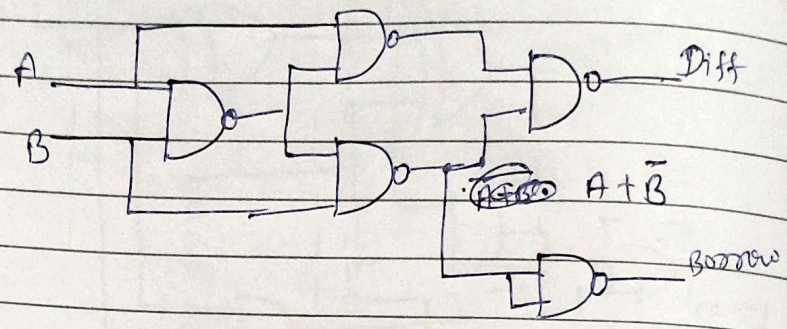
Jisme se minus kiya uske uper baad.

Step 4:- already minimized.

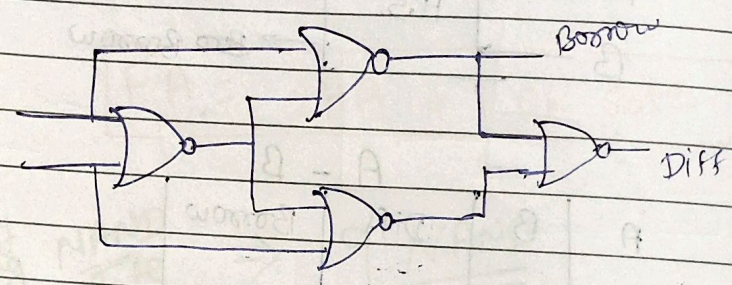
Step 5:-



using NAND (5)

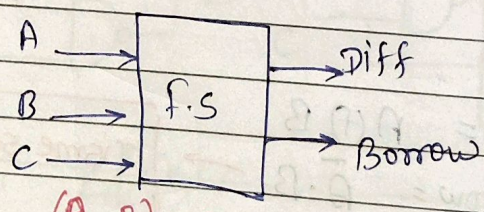


using NOR (5)



full subtractor

Step 1:-



Step 2:-

$(A - B) - C$

A	B	C	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

Step 3

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

$$\text{Borrow} = \Sigma(1, 2, 3, 7)$$

Step 4

$$\text{Borrow} = \Sigma m(1, 2, 3, 7)$$

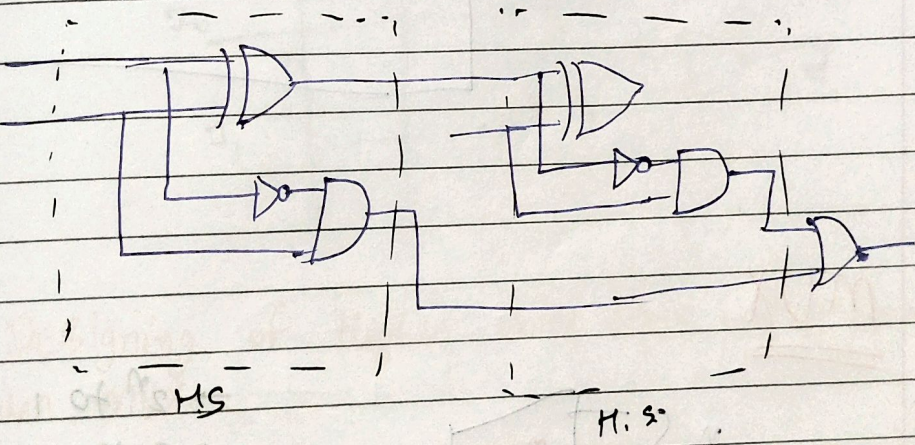
$$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$

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

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

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

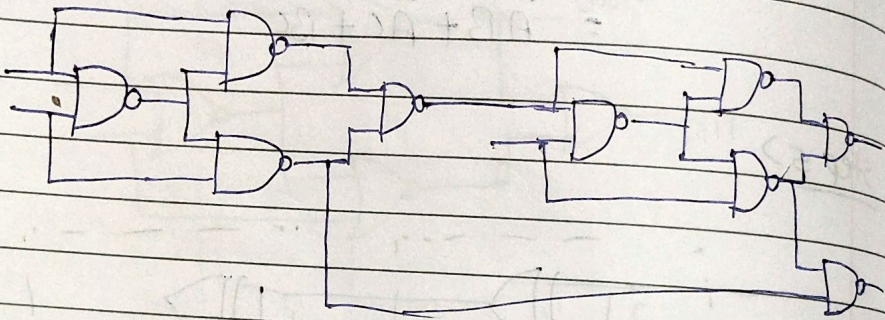
Step 5



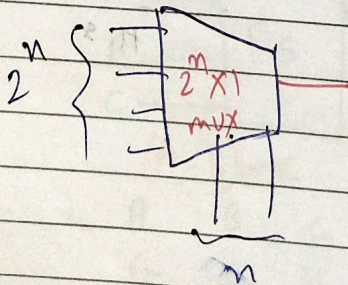
M.A } → 5
H.S }

F.A (9)	F.S (9)
Sum = $A \oplus B \oplus C$	Diff = $A \oplus B \oplus C$
Carry = $\Sigma m(3, 5, 6, 7)$	Borrow = $\Sigma m(1, 2, 3, 7)$
$= (A \oplus B)C + AB$	$= (\bar{A} \oplus B)C + \bar{A}B$
	$= (\bar{A} \odot B)C + \bar{A}B$

using NAND



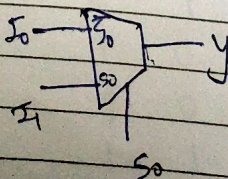
MUX



- 2^n to 1 converter
- AND-OR logic
- universal logic

Ques Design 2×1 MUX

step 1



Step:-2

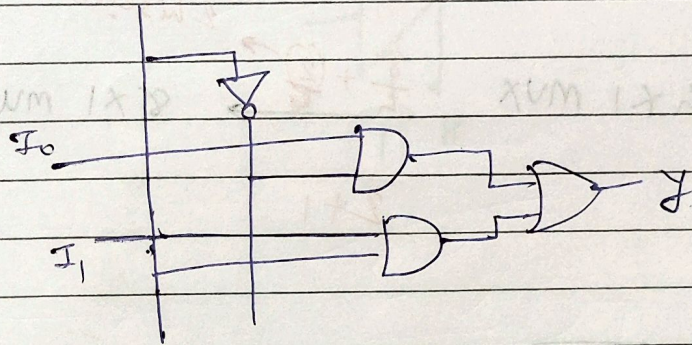
S_0	y
0	I_0
1	I_1

Step 3

$$y = \bar{S}_0 \cdot I_0 + S_0 \cdot I_1$$

Step:-4

Step 5

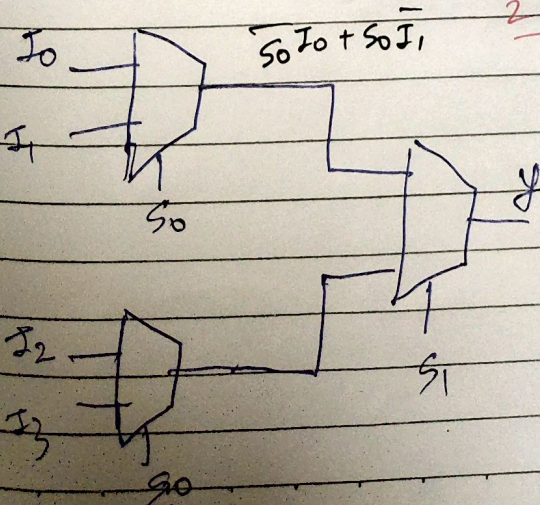


Type 1:- Designing of Higher order MUX using lower order MUX.

2x1 MUX $\longrightarrow$ 4x1 MUX

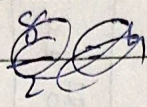
$$\frac{4}{2} + \frac{2}{2}$$

$$\frac{2+1}{2}$$



$2 \times 1 \xrightarrow[2^n - 1]{\quad} 2^n \times 1 \text{ mux}$

$2 \times 1 \text{ mux} \xrightarrow{\quad} 8 \times 1 \text{ mux}$



$$\frac{8}{2} + \frac{4}{2} + \frac{2}{2}$$

$$= 4 + 2 + 1 = 7$$

$2 \times 1 \text{ mux} \xrightarrow{\quad} 16 \times 1 \text{ mux}$

$$\frac{16}{2} + \frac{8}{2} + \frac{4}{2} + \frac{2}{2}$$

2 select karte hai.

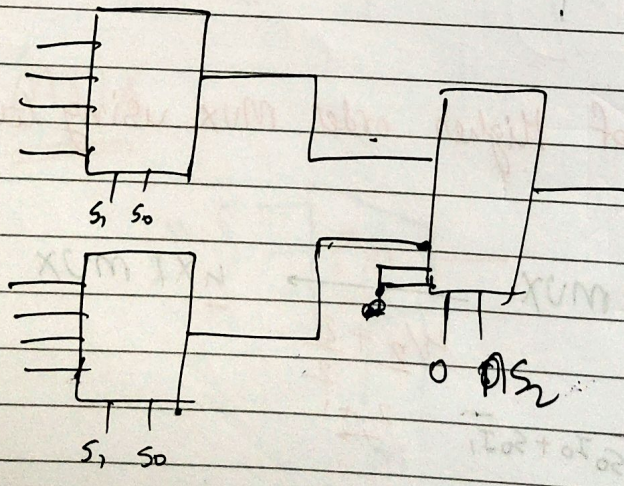
11

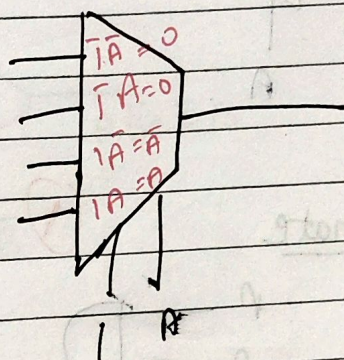
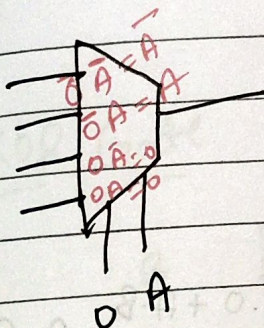
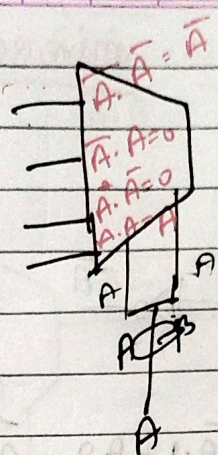
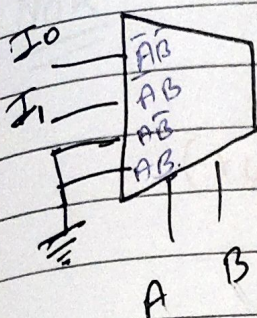
$4 \times 1 \text{ mux}$

$\frac{8}{2} + \frac{2}{2}$

$8 \times 1 \text{ mux}$

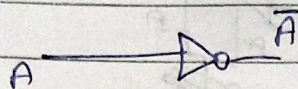
$2 + 1$



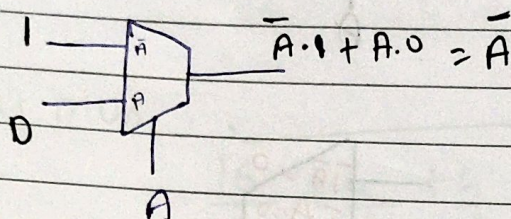


Type 2: MUX as a universal logic

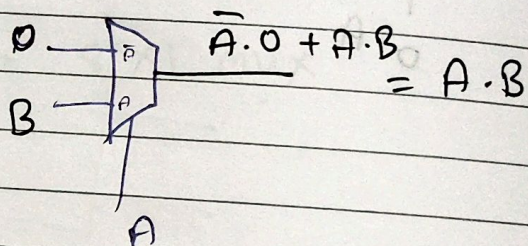
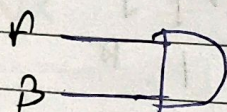
(1) NOT gate



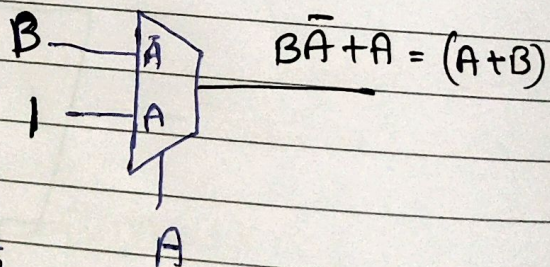
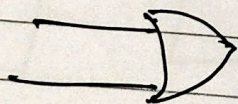
(2x1 mux)



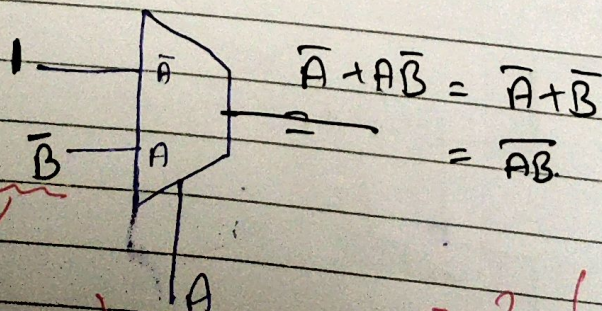
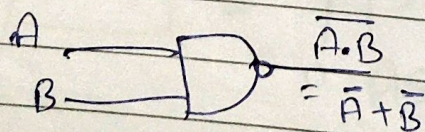
(2) AND Gate



(3) OR Gate



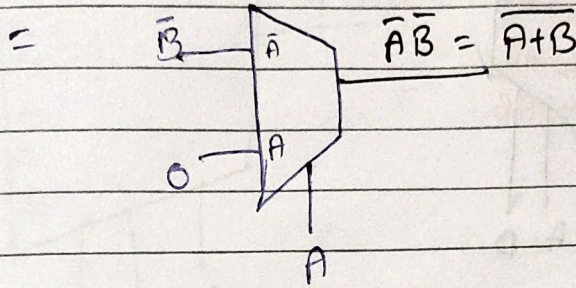
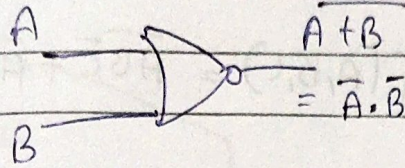
(4) NAND



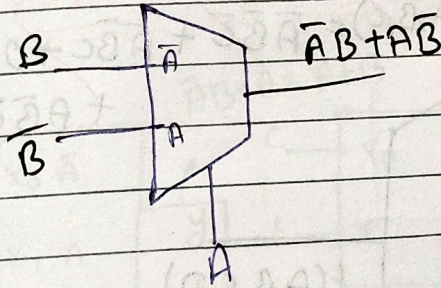
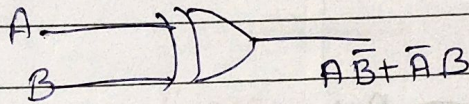
1 + 1 = 2 (2x1 mux)

- ① To design NOT, AND, OR $\rightarrow$ one 2x1 MUX
 ② To design NAND, NOR, XOR, XNOR $\rightarrow$ two 2x1 MUX

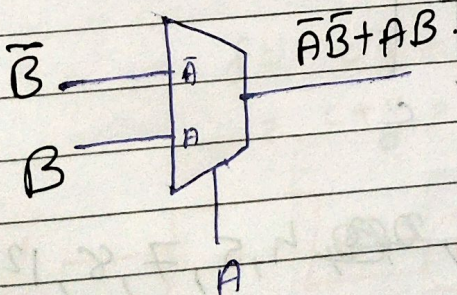
⑤ NOR



⑥ XOR Gate



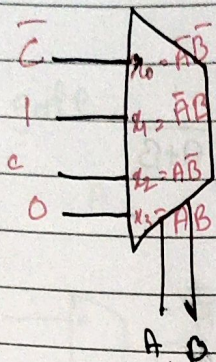
⑦ XNOR Gate



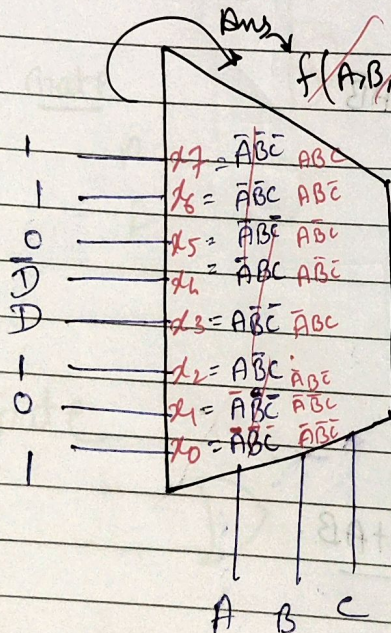
Type 3 minimization

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

①



②



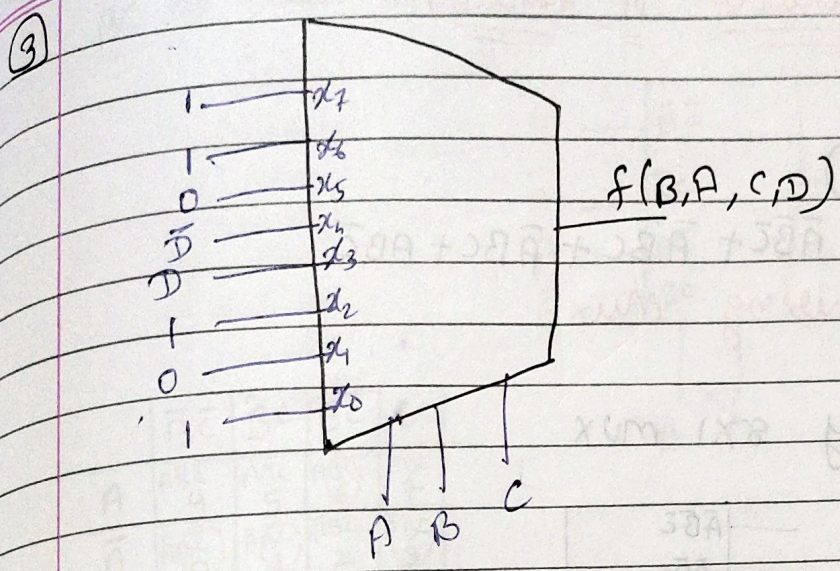
$$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + 0 + \bar{A}B C D + \bar{A}B \bar{C} D + \bar{A}B C + 0 + \bar{A}B C$$

$$f(A,B,C,D)$$

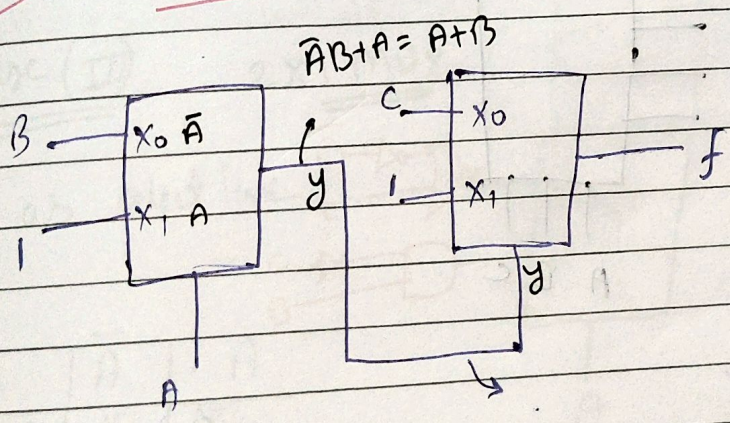
$$f(A,B,C,D) = \sum m (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$$

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

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

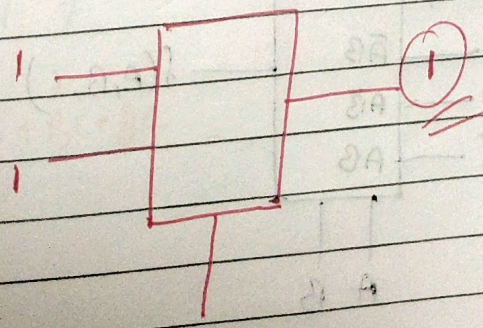


Type-4 cascading of Mux



$$f = c\bar{y} + y$$

$$= c + y = \boxed{A + B + C}$$



always one if both input are 1..

Type:-5 Implementation of function

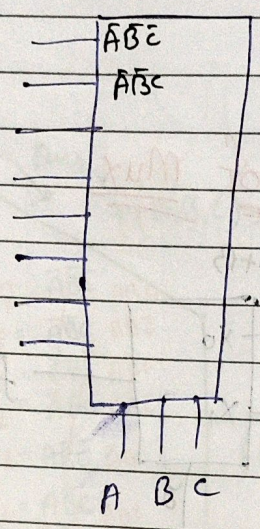
example

$$f(A, B, C)$$

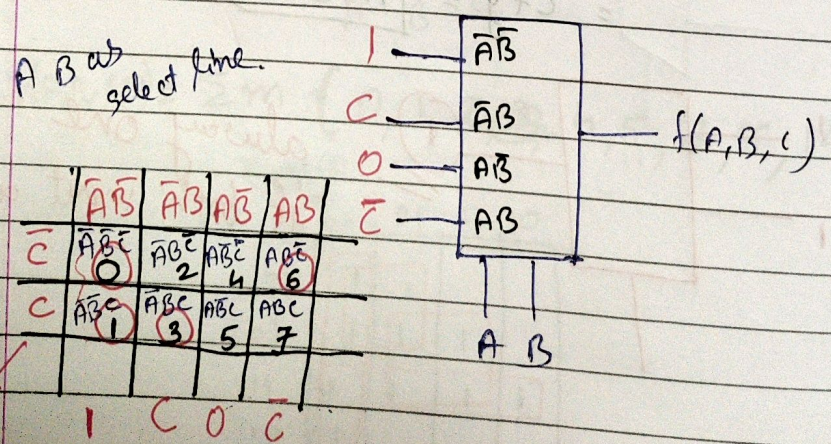
$$= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

Implement using mux

case(I):- By 8x1 mux



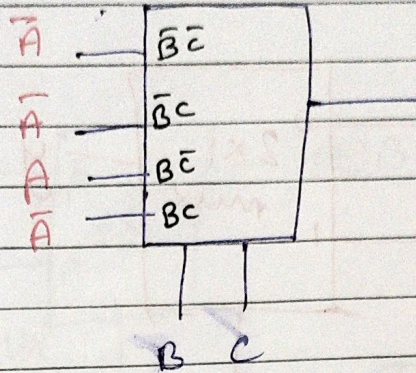
case(II) :- using 4x1 mux $\Sigma m(0, 1, 3, 6)$



$\bar{A}\bar{B}$ re with $\bar{C}$ & C jaise hi value

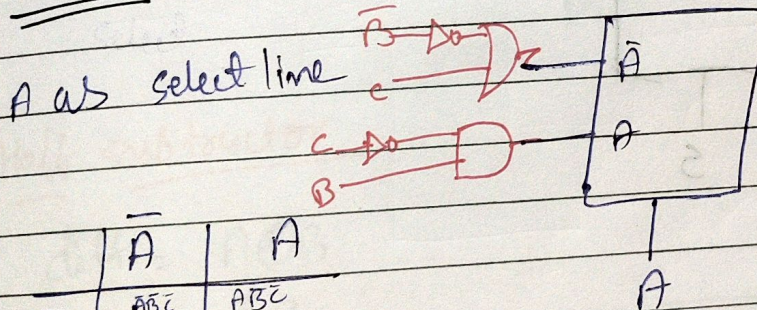
BC as select line.

$\Sigma m(0, 1, 3, 6)$



	$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	BC
A	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5	$AB\bar{C}$ 6	ABC 7
$\bar{A}$	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$\bar{A}B\bar{C}$ 2	$\bar{A}BC$ 3
	$\bar{A}$	$\bar{A}$	A	A

case (II) 2x1 MUX



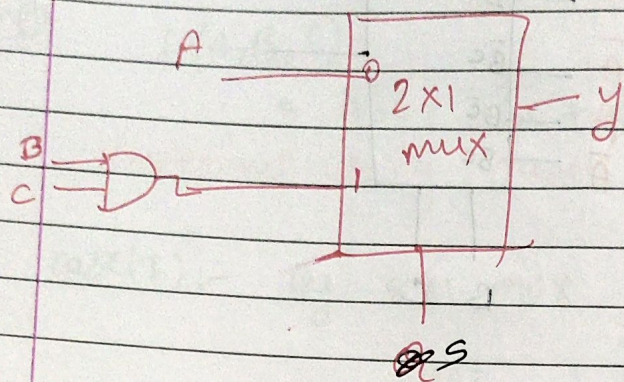
	$\bar{A}$	A
$\bar{B}\bar{C}$	$\bar{A}\bar{B}\bar{C}$ 0	$A\bar{B}\bar{C}$ 4
$\bar{B}C$	$\bar{A}\bar{B}C$ 1	$A\bar{B}C$ 5
$B\bar{C}$	$\bar{A}B\bar{C}$ 2	$AB\bar{C}$ 6
BC	$\bar{A}BC$ 3	ABC 7

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

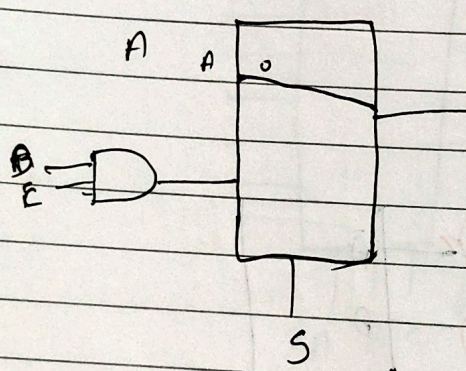
$$= \underline{\underline{\bar{B} + C}}$$

Type 6: Delay in mux

$T_{AND} = 1 \mu s$
 $T_{mux} = 2 \mu s$

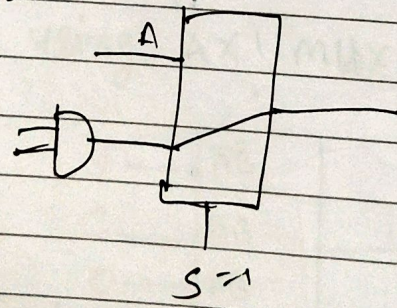


case (I) $S=0$



Delay = $T_{mux} = 2 \mu s$

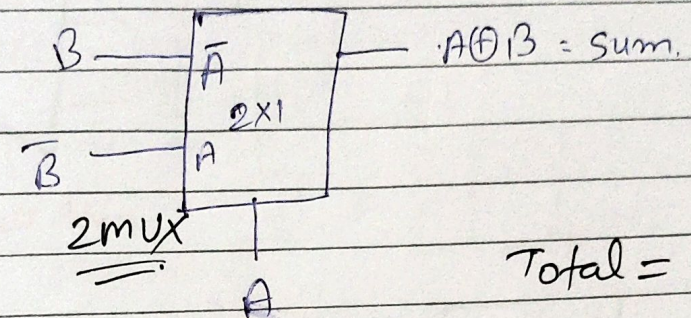
case (II) $S=1$



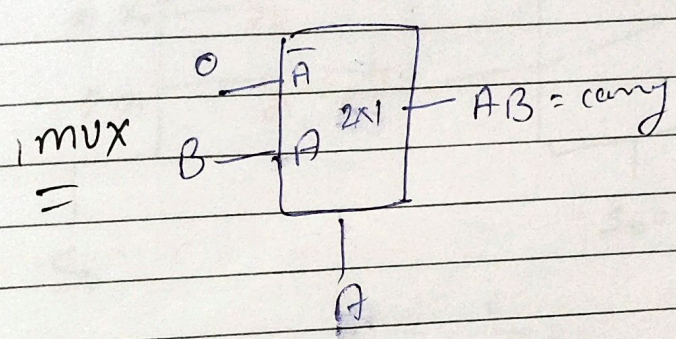
Delay = $T_{AND} + T_{mux}$
 $= 1 + 2$
 $= 3 \mu s$

Ques To Design H.A How many 2X1 mux are required.

Sum = $A \oplus B$
 Carry = AB

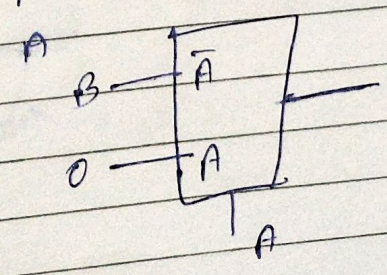
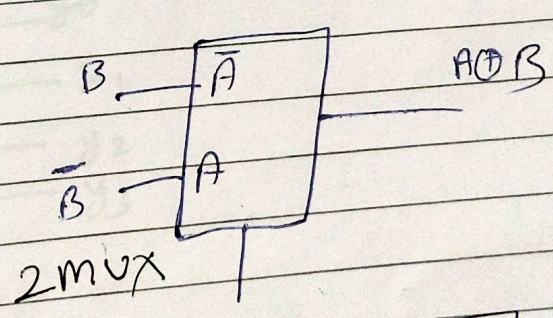


Total = 3mux



Half subtractor

diff = $A \oplus B$
 Borrow = $\bar{A}B$



4

How

FA & FS using 2x1 MUX

$X_{MS} = 2 \log T$

X_{MS}

Delay = $Mux + 2 \log T$

rotational delay

$2 \oplus A = A \oplus A$

$2 \oplus B = A \oplus B$

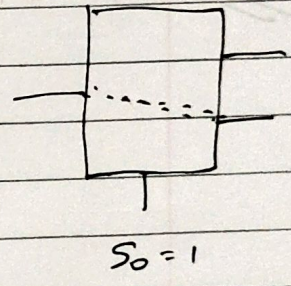
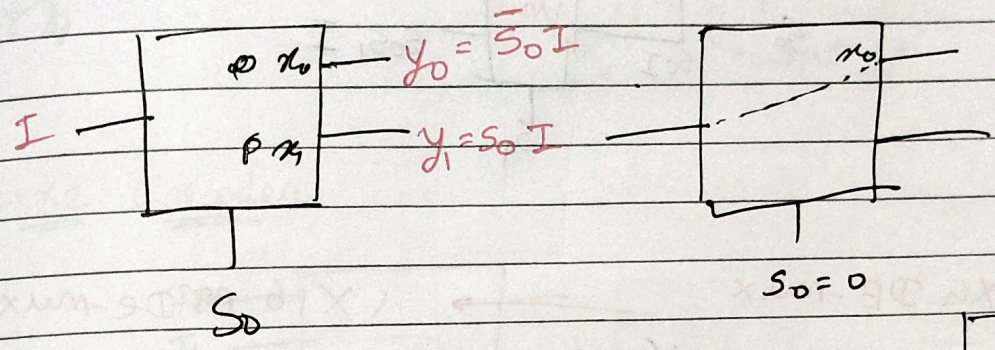
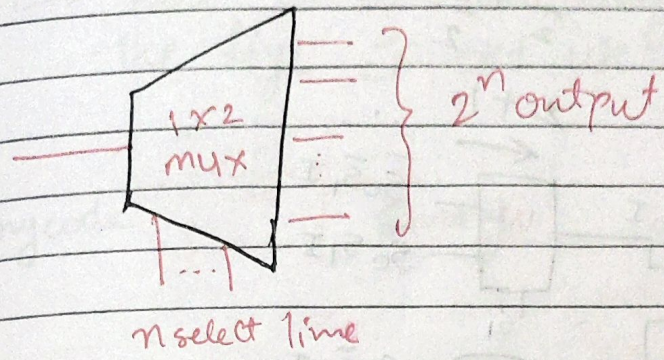
Delay = $T_{mux} + T_{rot}$

$= 10 + 10$

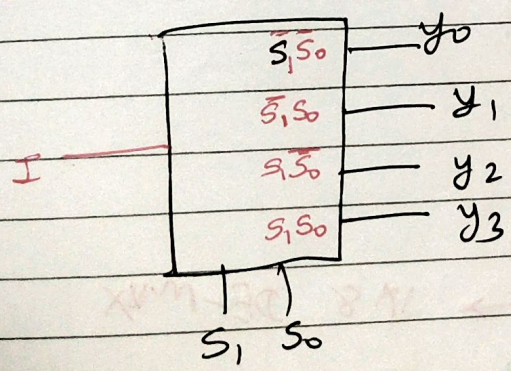
$= 20$

X_{MS}

DE-MUX



⇒ Design 1x4 De-mux

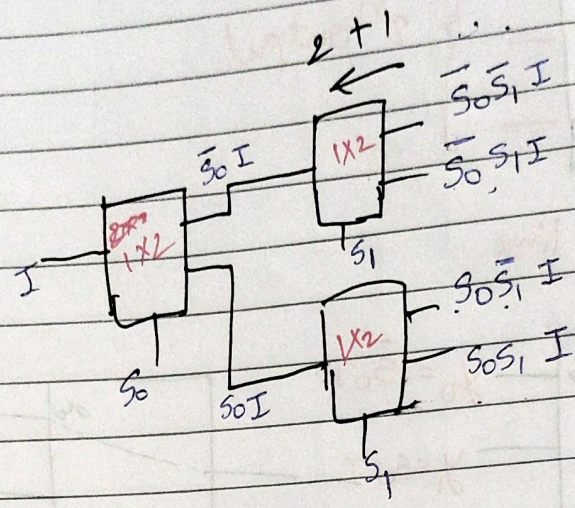


Que

1x2 De-mux

1x4 De-mux

$$\frac{4}{2} + \frac{2}{2}$$



Que

1x4 De-mux

1x16 De-mux

$$\frac{16}{4} + \frac{4}{4}$$

4+1

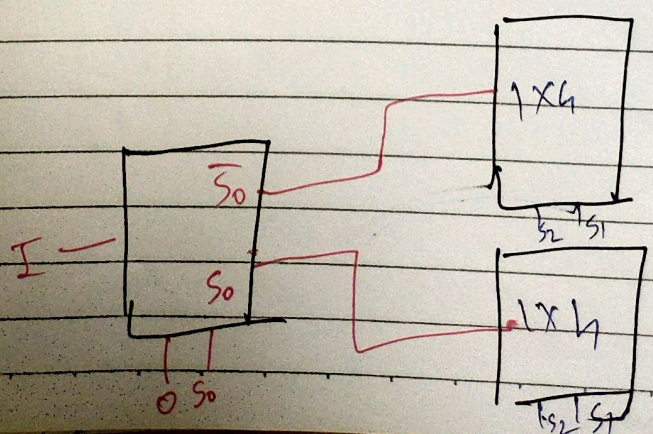
Que

1x4 Demux

1x8 De-mux

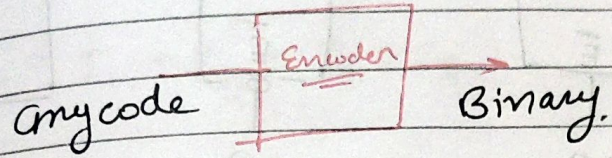
$$\frac{8}{4} + \frac{2}{4}$$

2 hi output change



Encoder

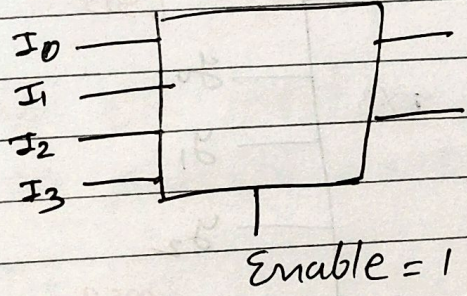
→ Kisi ko code ko binary me convert karne ke liye jo ckt use hoti hai woh Encoder



682

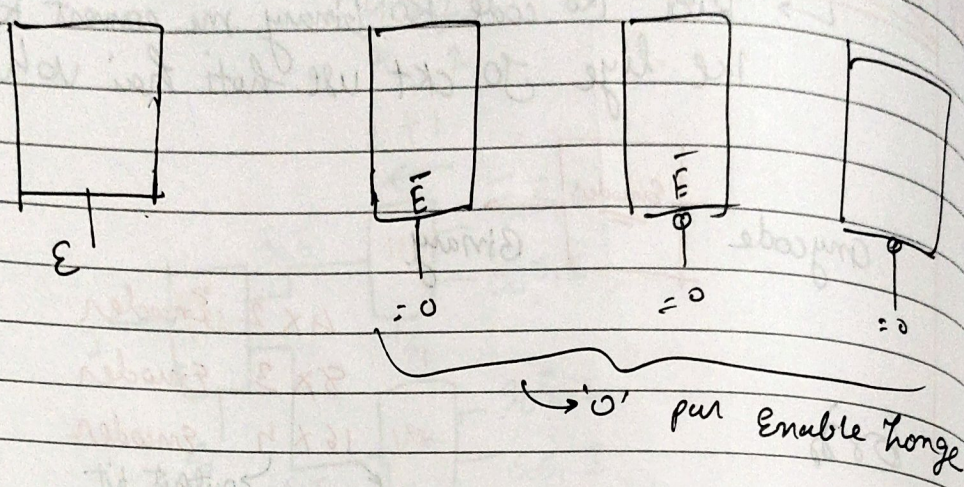
- 4x2 Encoder
 - 8x3 Encoder
 - 16x4 Encoder
- input → → output bit

→ 4x2 Encoder



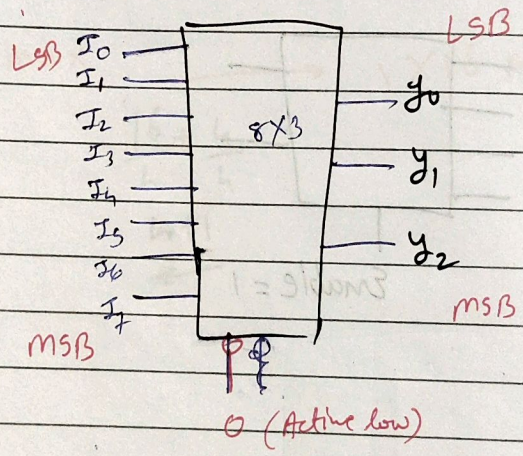
Active high

Active low



⇒ 8x3 MSB Encoder

Step 1:-



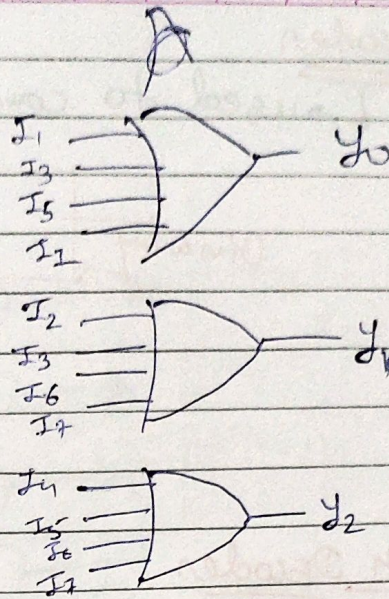
Step 2:-

MSB									LSB	LSB			
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		y_2	y_1	y_0		
0	0	0	0	0	0	0	1		0	0	0	0	0
0	0	0	0	0	0	1	0		0	0	1	1	1
0	0	0	0	0	1	0	0		0	1	0	2	2
0	0	0	0	1	0	0	0		0	1	1	3	3
0	0	0	1	0	0	0	0		1	0	0	4	4
0	0	1	0	0	0	0	0		1	0	1	5	5
0	1	0	0	0	0	0	0		1	1	0	6	6
1	0	0	0	0	0	0	0		1	1	1	7	7

$$y_0 = I_1 + I_3 + I_5 + I_7$$

$$y_1 = I_2 + I_3 + I_6 + I_7$$

$$y_2 = I_4 + I_5 + I_6 + I_7$$



→ multiple input high

↳ kisi ek input ko priority dena hai

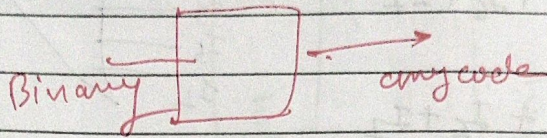
→ If LSB priority encoder

x can be
0 or 1

I_3	I_2	I_1	I_0	y_1	y_0
x	x	x	1	0	0
x	x	1	0	0	1
x	1	0	0	1	0
1	0	0	0	1	1

Decoder

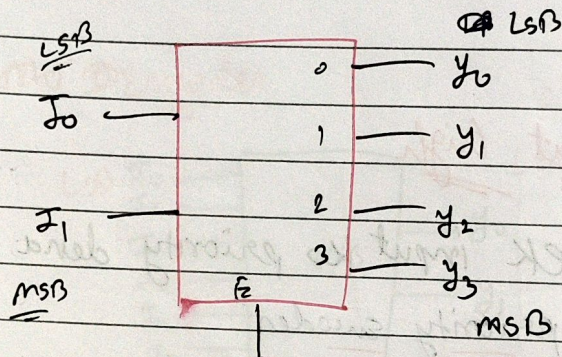
→ used to convert Binary to any code



- 1) 2×4 decoder
- 2) 3×8 decoder
- 3) 4×16 decoder

⇒ 2×4 Decoder

step 1)



step 2)

I_1	I_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

step 3)

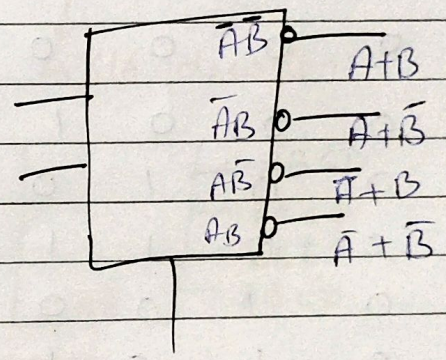
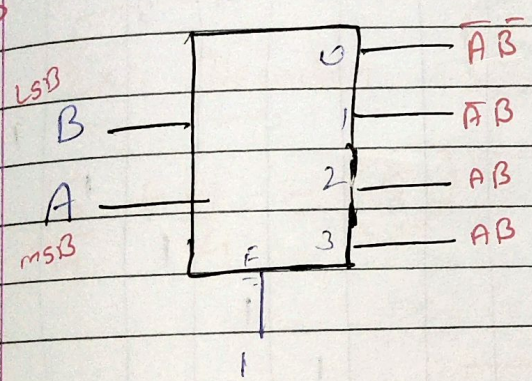
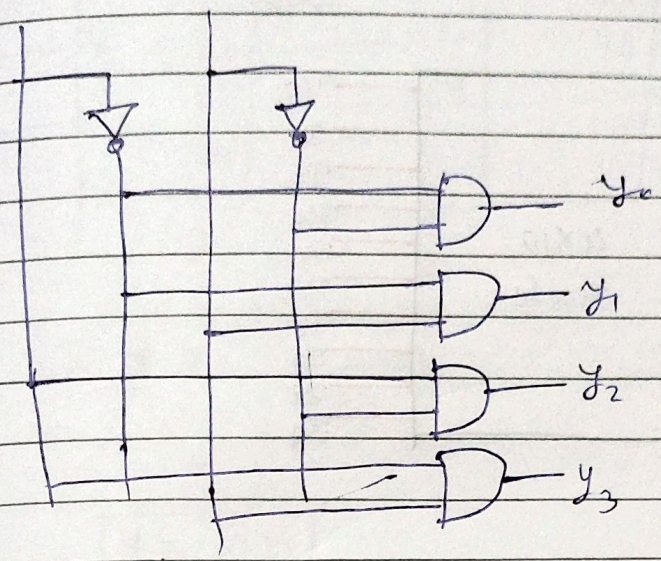
$$y_0 = \bar{I}_1 \bar{I}_0$$

$$y_1 = \bar{I}_1 I_0$$

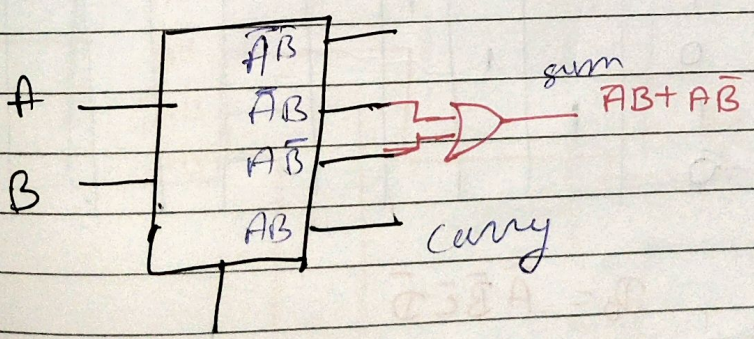
$$y_2 = I_1 \bar{I}_0$$

$$y_3 = I_1 I_0$$

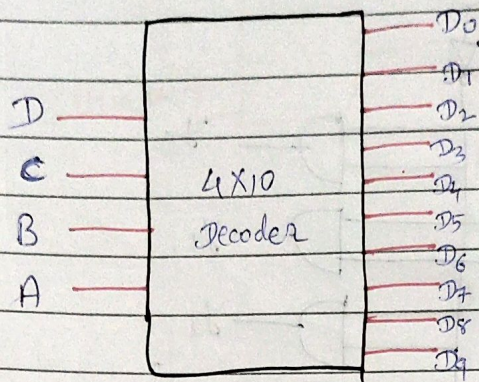
step 4 already minimized
 step 5



⇒ Half adder



⇒ Design a Binary to Decimal Decoder



A	B	C	D	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	0										1
0	0	0	1									1	
0	0	1	0								1		
0	0	1	1							1			
0	1	0	0					1					
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	0	1	1										

Do

cd

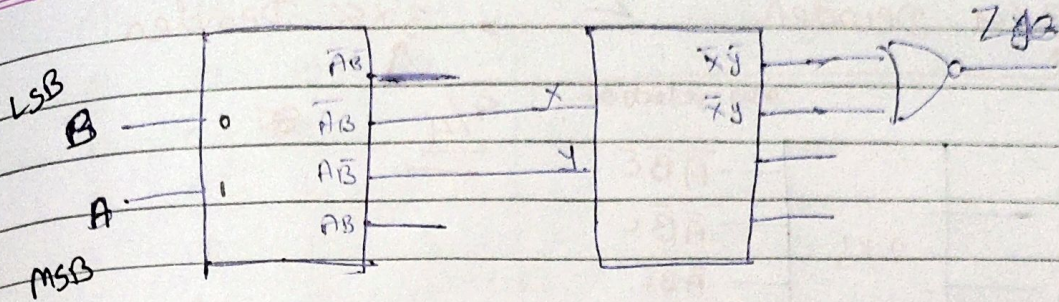
AB	00	01	10	11
0	1			
1				
2	X	X	X	X
3			X	X

- $D_0 = \bar{A}\bar{B}\bar{C}\bar{D}$
- $D_1 = \bar{A}\bar{B}\bar{C}D$
- $D_2 = \bar{B}C\bar{D}$
- $D_3 = \bar{B}CD$
- $D_4 = B\bar{C}\bar{D}$
- $D_5 = B\bar{C}D$
- $D_6 = B\bar{C}\bar{D}$
- $D_7 = BCD$

$D_8 = AD$
 $D_9 = A\bar{D}$

Make CKT of Don't care

Q



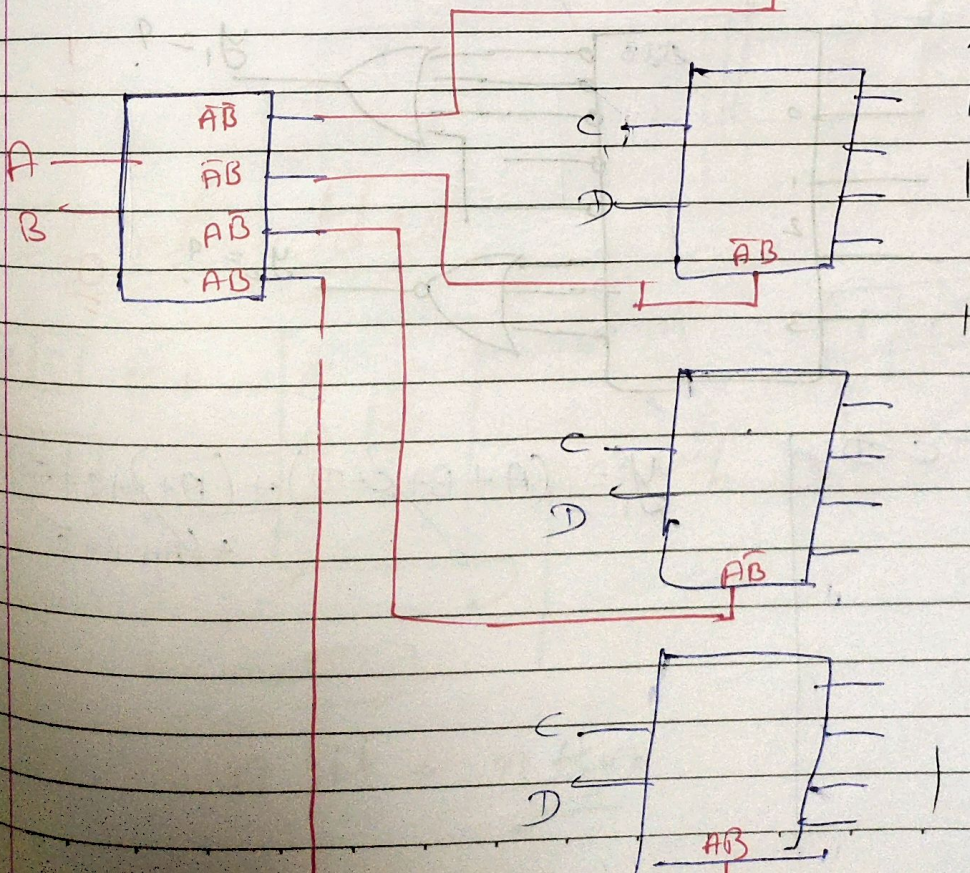
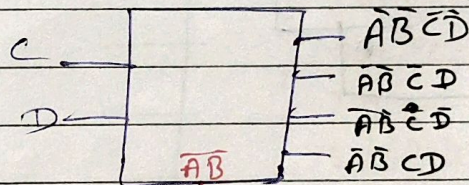
$$Z = \bar{x}\bar{y} + \bar{x}y$$

$$= x$$

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

⇒ 2x4 Decoder (5) → 4x16 decoder

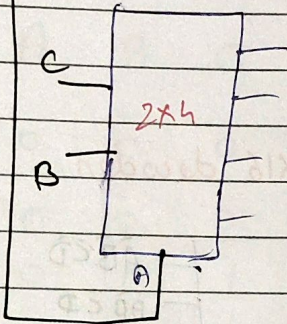
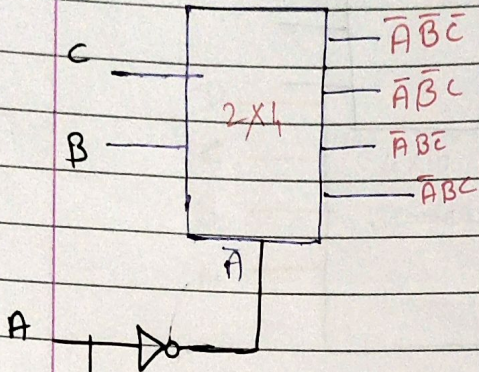
$$\frac{16}{4} + \frac{4}{4} = (5)$$



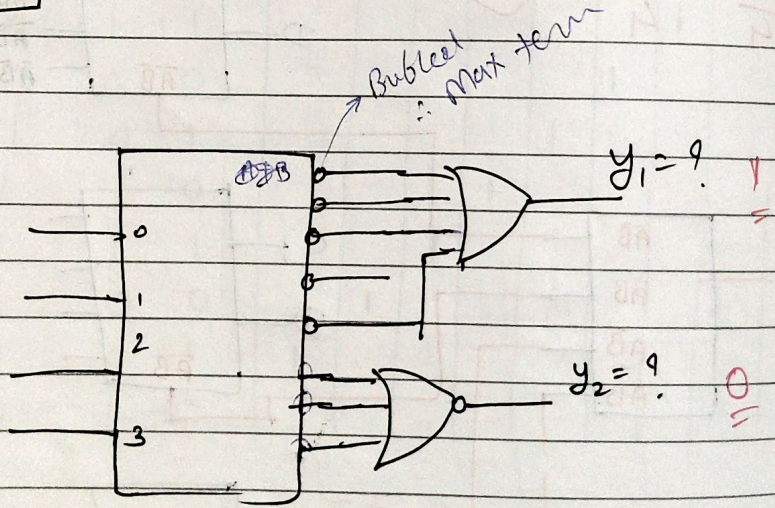
$$\frac{8}{4} = 2$$

⇒ 2x4 Decoder → 2 → 3x8 Decoder

no relation ∴ $\frac{8}{4}$



One)



A B C D

$$y_1 = (A + B + C + D) + (A + B + C + \bar{D}) + (A + B + \bar{C})$$

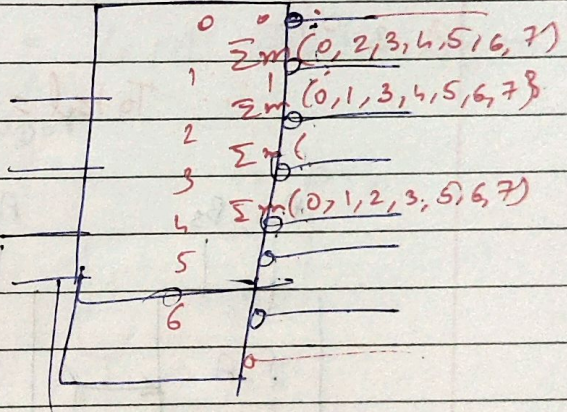
slowest adder:-

serial adder

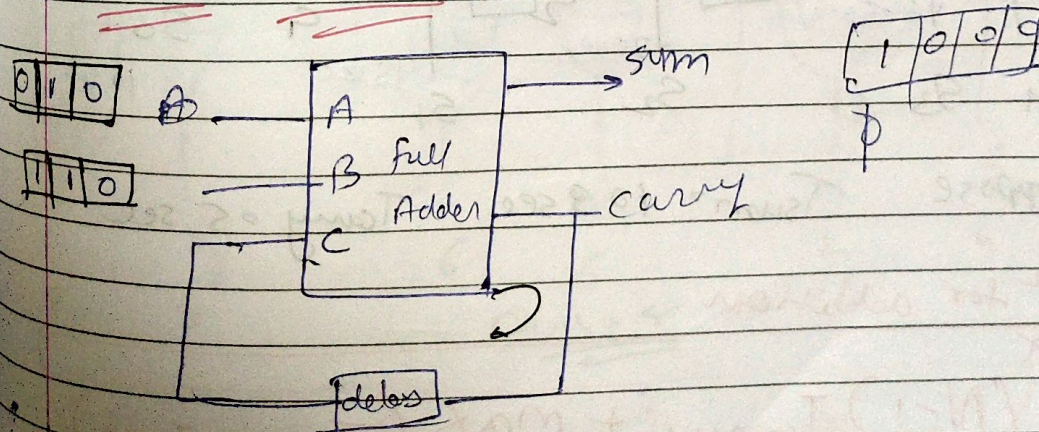
A	B	C	D	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	

AB	00	01	10	11
00	0	.	X	X
01	.	.	X	X
10	.	.	X	X
11	.	.	X	X

0 ke abhi sub
 $\Sigma m(1, 2, 3, 4, 5, 6, 7)$



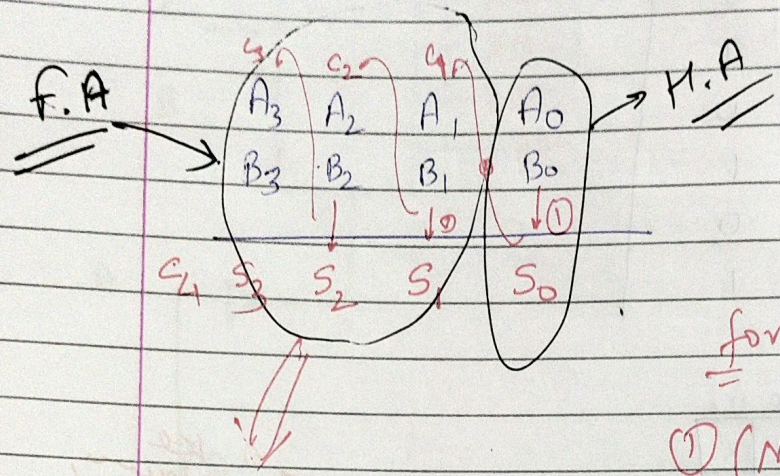
Serial adder



n bit = n term

① Parallel Adder / Ripple carry adder

↳ All the inputs are connected simultaneously

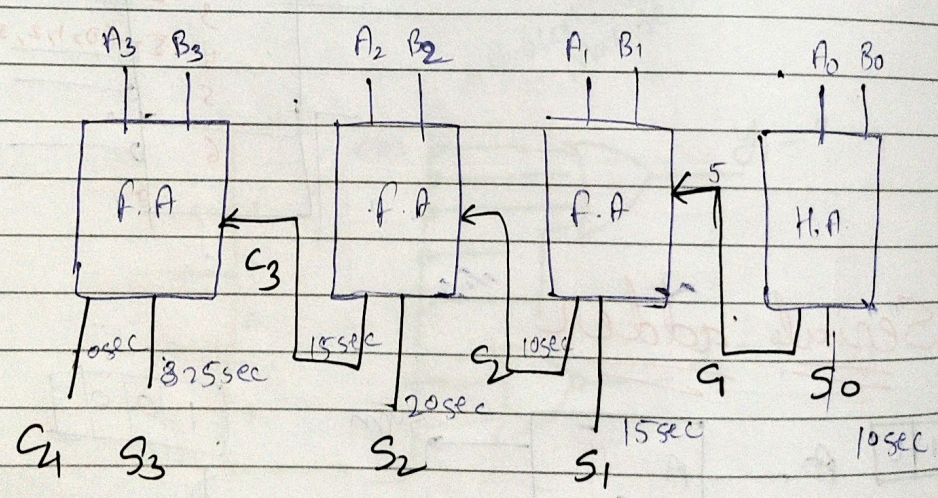


for n-bit

6. HA.
 & 3 OR

- ① $(N-1) \cdot FA + 1 \cdot HA$
- ② $N \cdot FA$
- ③ $2N-1 \cdot HA + (N-1) \cdot OR$

$\therefore \text{Total} = 7 \cdot HA + 3 \cdot OR$



Let suppose $T_{sum} = 10 \text{ sec}$ $T_{carry} = 5 \text{ sec}$

Total time for addition
 $T_{total} = (N-1) T_{carry} + \max(T_{sum}, T_{carry})$

Q2 For 16 bit ripple carry adder, $T_{sum} = 25 \text{ ns}$, $T_{carry} = 25 \text{ ns}$, find worst and delay and addition per second?

$$T = (N-1) T_{carry} + \max \{ T_{sum}, T_{carry} \}$$

$$= (15 \times 25 \text{ ns}) + 25 \text{ ns}$$

$$= 250 \text{ ns} \rightarrow \text{Time for addition of 16 bit}$$

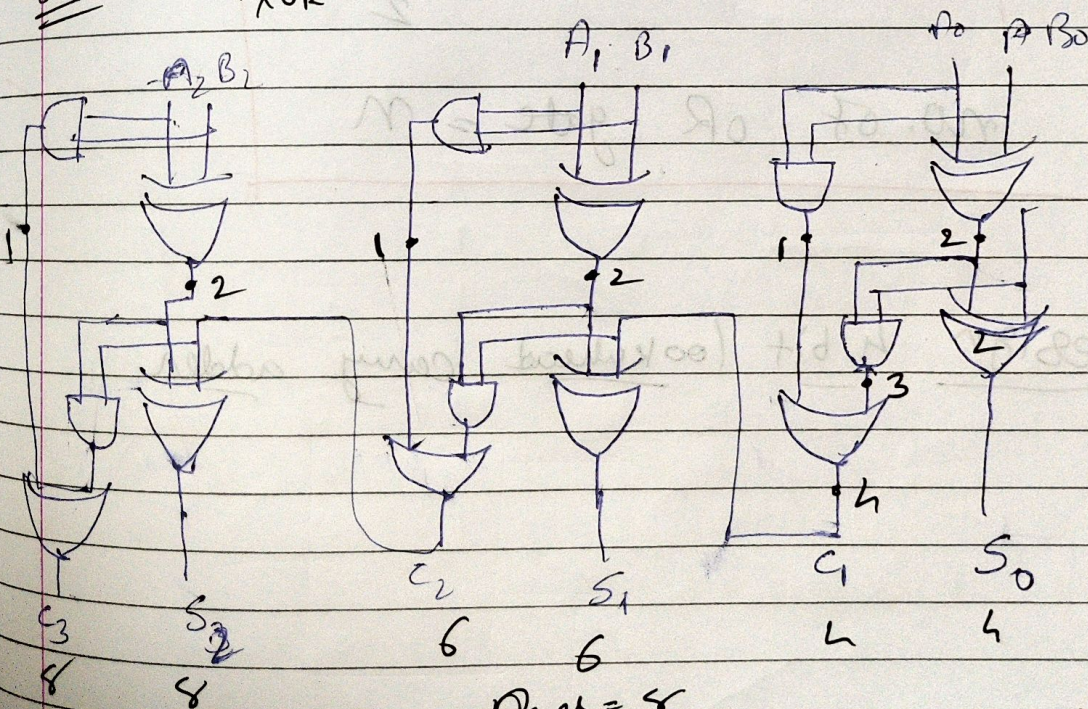
$$\therefore \text{addition per second} = \frac{1}{T}$$

$$= \frac{1}{250 \times 10^{-9} \text{ sec}}$$

$$= 4 \text{ M addition / sec}$$

3 bit

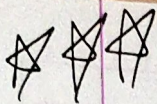
$$T_{XOR} = 2 \quad T_{OR} = 1 \quad T_{AND} = 1$$



$$D_{ms} = 8$$

if $T_{sum} > T_{carry}$

$$T = (N-1) \{ T_{AND} + T_{OR} \} + 2 T_{XOR}$$



Q1 Look Ahead carry $\rightarrow$ large delay.

$$\text{Total Delay} = T_{XOR} + T_{AND} + T_{OR} + T_{XOR}$$

$$\text{if } T_{XOR} = T_{AND} = T_{OR} = T$$

$$\text{Total Delay} = 4T$$

$$\text{Delay of carry block} = T_{AND} + T_{OR}$$

for carry Block (n bit)

$$\text{no. of AND gate} = \frac{N(N+1)}{2}$$

$$\text{no. of OR gate} = N$$

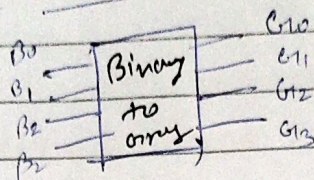
H.W

Design 4 bit lookahead carry adder

Code converter

⑪ ⇒ Design a circuit which convert Binary into Gray

Step 1:-



Step 2:- Truth table

B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0 = B_1 \oplus B_0$$

B ₂ B ₁	00	01	11	10
B ₃ B ₀				
00				
01	1	1	1	1
11				
10	1	1	1	1

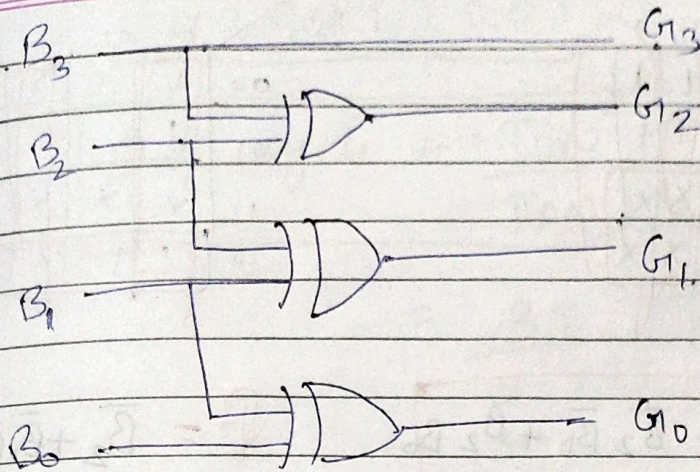
B ₂ B ₁	00	01	11	10
B ₃ B ₀				
00				
01				
11				
10				

$$G_3 = B_3$$

$$G_2 = B_3 \oplus B_2$$

$$G_1 = B_2 \oplus B_1$$

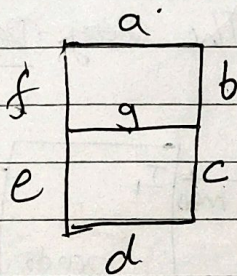
$$G_0 = B_1 \oplus B_0$$



(self complemented)

Binary to Excess-3 code

Binary to 7 segment display



0 → a, b, c, d, e, f

1 → b, c

2 → a, b, g, e, d

3 → a, b, g, c, d

4 → f, g, b, c, d

7 → ~~input~~ → ~~output~~

	A ₃	A ₂	A ₁	A ₀	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1		1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1					
5	0	1	0	1	1	0					
6	0	1	1	0	1	0					
7	0	1	1	1	1	1					
8	1	0	0	0	1	1					
9	1	0	0	1	1	1					

Duty cycle = $\frac{T_{ON}}{T_{ON} + T_{OFF}}$

$B_3 B_0$ (a)

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

$B_3 B_2$

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

$$A = B_3 + B_1 + \bar{B}_2 \bar{B}_0 + B_2 B_0$$

$$B = \bar{B}_2 + \bar{B}_1 \bar{B}_0 + B_3 B_0$$

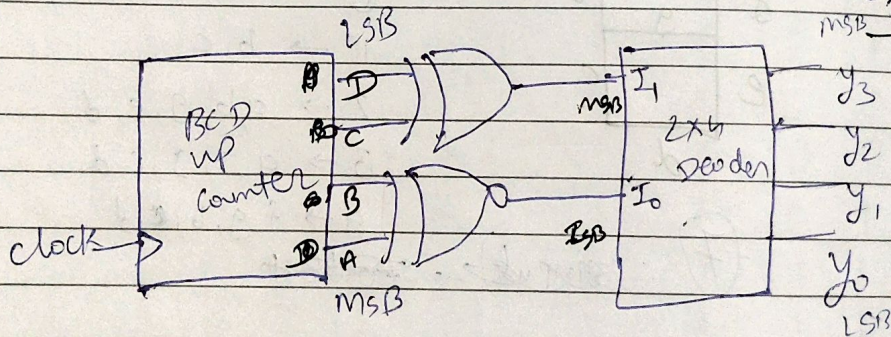
$$A = (B_2 \odot B_0) + B_3 + B_1$$

$$B = \bar{B}_2 + (B_1 \odot B_0)$$

Ques

Consider the given ckt. The Delay cycle of $\frac{1}{2}$??

$2, 10$



Clock	A	B	C	D	$I_1 = CD$	$I_0 = AB$	y_3	y_2	y_1	y_0
0	0	0	0	0	0	1	0	0	1	0
1	0	0	0	1	1	1	1	0	0	0
2	0	0	1	0	1	1	1	0	0	0
3	0	0	1	1	0	1	0	0	1	0
4	0	1	0	0	0	0	0	0	0	1
5	0	1	0	1	1	0	0	1	0	0
6	0	1	1	0	1	0	0	1	0	0
7	0	1	1	1	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	0	1
9	1	0	0	1	1	0	0	1	0	0

$$y_2 = \overline{000 \cdot 001 \cdot 1001} \dots$$

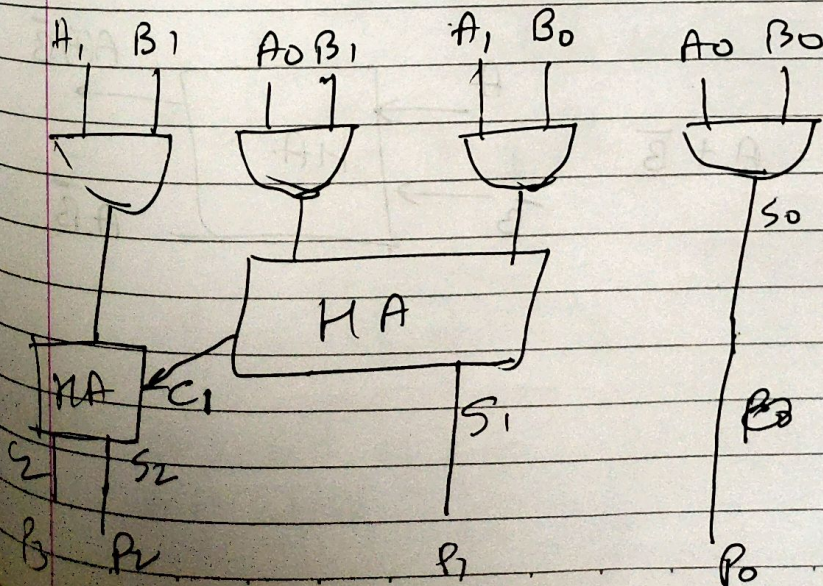
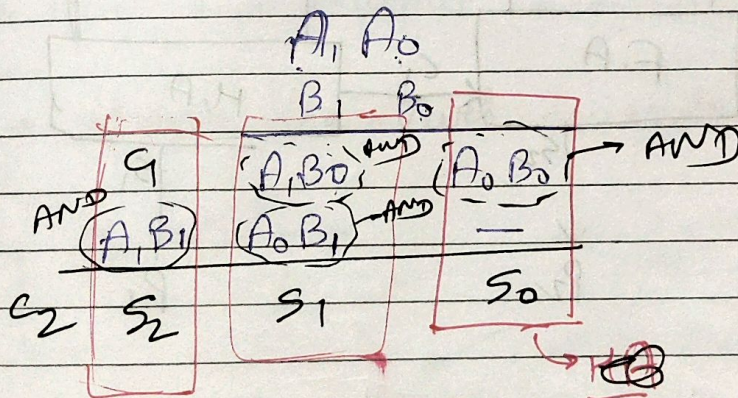
$$\text{Duty cycle} = \frac{T_{ON}}{T_{ON} + T_{OFF}}$$

$$= \frac{3T}{3T + 7T} \times 100\%$$

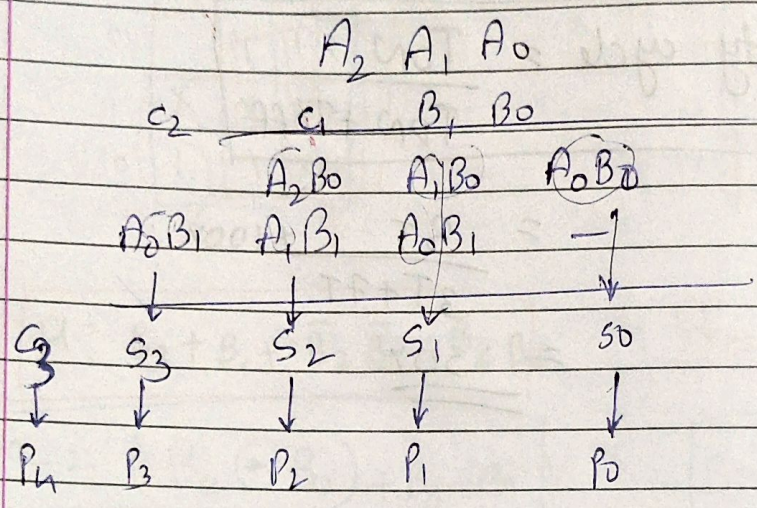
$$= \underline{\underline{30\%}}$$

Multiplication

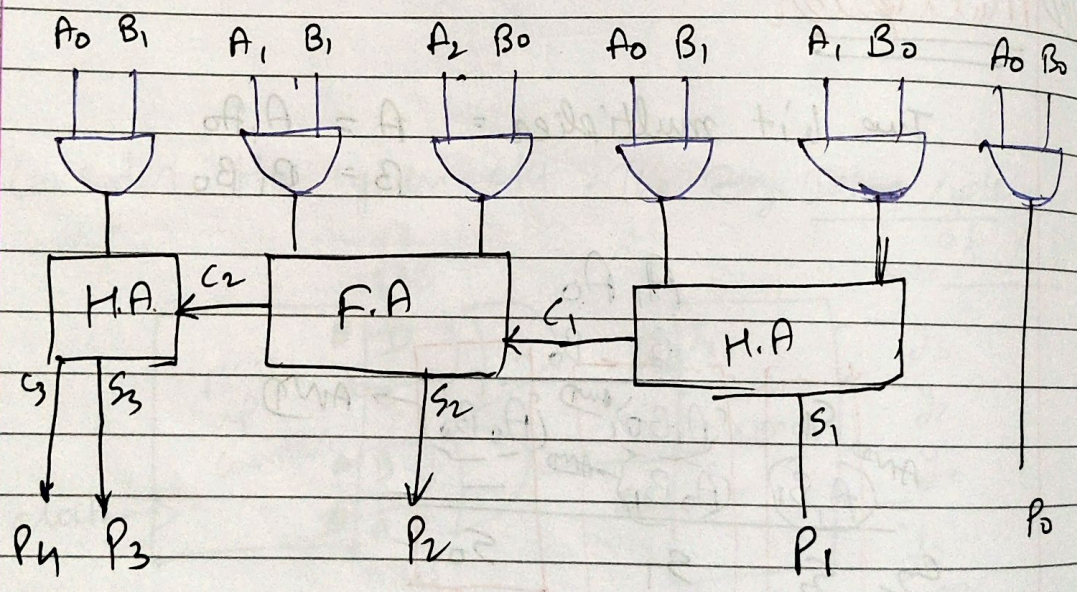
Two bit multiplier = $A = A_1 A_0$
 $B = B_1 B_0$



Q $A = A_2 A_1 A_0$ $B = B_1 B_0$ $A \times B$

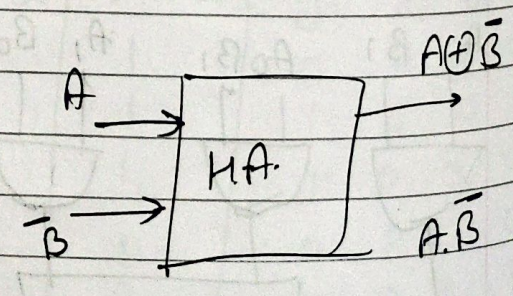


6 AND



Subtractors

~~$A \oplus B$~~ $A - B = A + \bar{B}$



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

Sequential ckt

- A circuit with feedback or memory are called sequential ckt.
- present output of sequential ckt depends on present input as well as previous state of output.
- ∴ Dynamic System

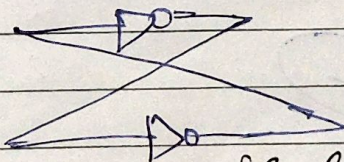
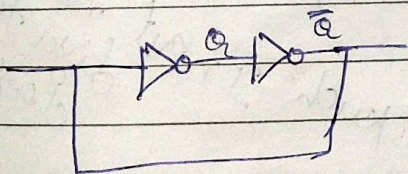
- (1) Latch
- (2) Flip-flop
- (3) Register
- (4) counters.

LATCHES

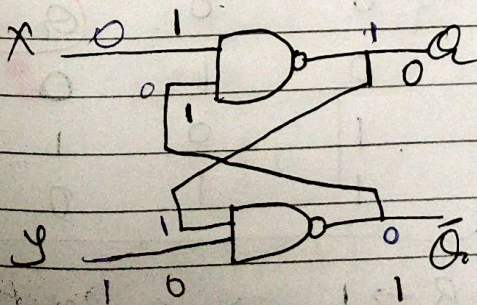
↳ Basic memory element

↳ Two ~~out~~ output which is complement of each other

↳ latches are level sensitive



output lagu panyo is liye NOT ko hata ke NAND laga diya



X	Y	Q	Q̄
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	Q̄

Invalid

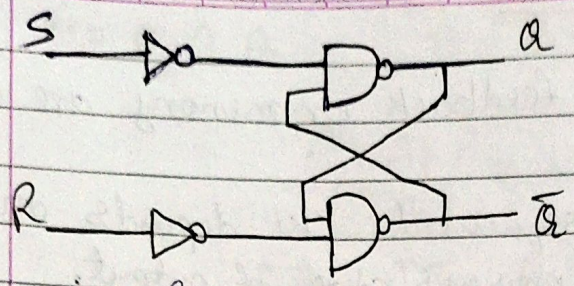
HOLD

NAND:- Any one input is '0' output is '1'

Jab $Q = 0$ ho Jaye
tab called RESET

SR LATCH

M T W T F S S
Page No.:
Date:
YOUVA



① $\begin{matrix} S & R \\ 0 & 0 \end{matrix} \rightarrow \bar{S} \rightarrow 1$
 $\bar{R} \rightarrow 1$

when ~~1~~ $\rightarrow$ depend on
 $\bar{Q}$ & Q

② $\begin{matrix} S & R \\ 0 & 1 \end{matrix} \rightarrow \bar{S} \rightarrow 1$
 $R \rightarrow 0$

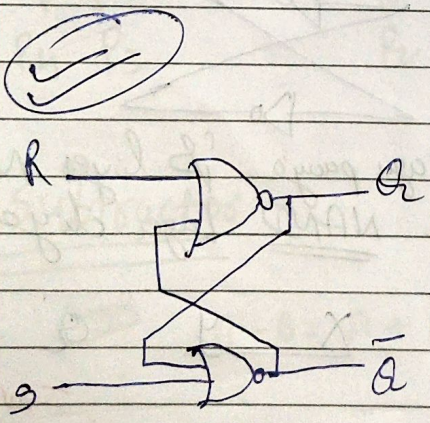
R vale ka o/p $\bar{Q} \rightarrow 1$ hogya

so uper vale AND $1 \rightarrow Q = 0$

Set $\leftarrow \begin{matrix} S & R \\ 0 & 0 \end{matrix} \rightarrow$ hold

Reset $\leftarrow \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \rightarrow$ reset { when $Q = 0$ }
dabaza $\rightarrow$ set { when $Q = 1$ }

Set press
or



S	R	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

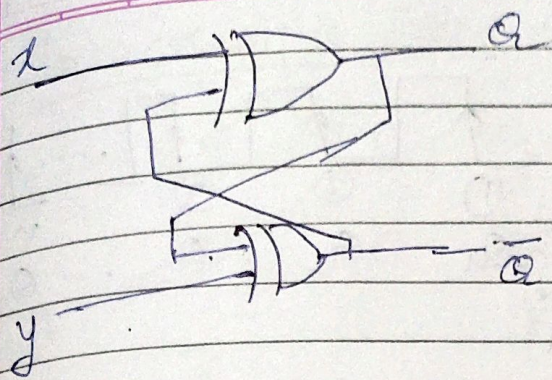
$\rightarrow$ HOLD

{ Reset
set
invalid

push Input nahie
dijay toh
change hui
naha

not
when any one i/p
"high" o/p "low"

$R \rightarrow 1 \Rightarrow$ Reset
 $S \rightarrow 1 \Rightarrow$ set



x	y	Q	Q̄
0	0	0	1
0	1	1	0
1	0	0	1
1	1	1	0

Toggle

HOLD

$$x \oplus x = 0 \Rightarrow x \oplus 0 = x$$

$$x \oplus \bar{x} = 1 \Rightarrow x \oplus 1 = \bar{x}$$

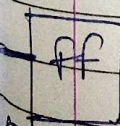
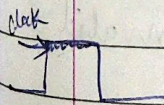
Flip-flop

Latches with control Phenomenon are called flip-flop

edge Trigger

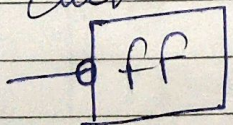
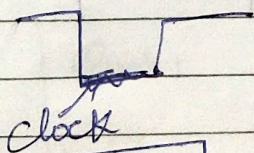
level

positive



positive level trigger
 (Kip par kadam kurega)

negative

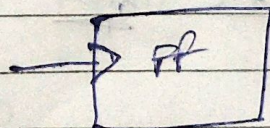
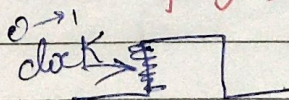


negative level trigger
 (zero par kadam kurega)

Edge

positive

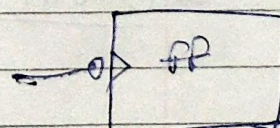
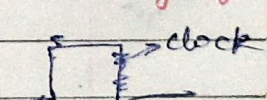
(Rising edge)



(0 → 1)
 Transition
 par (ON)

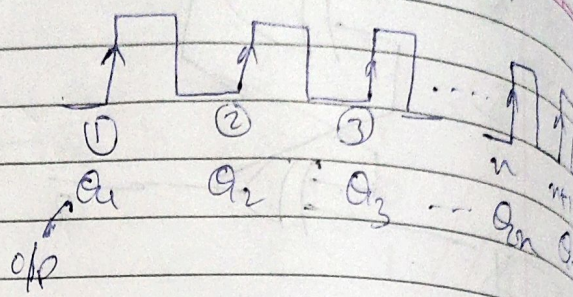
negative

(Falling edge)



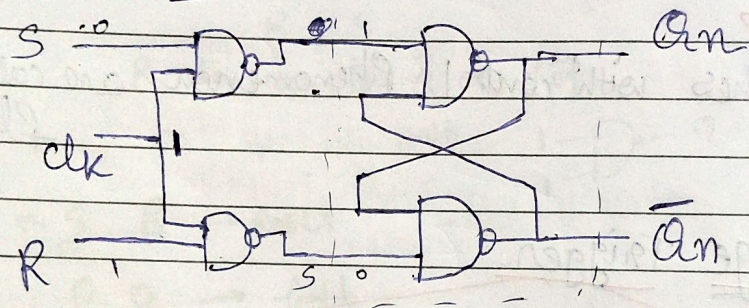
(1 → 0)
 par (ON)

- ① ckt Diagram
- ② Truth Table
- ③ characteristic Table
- ④ characteristic equation
- ⑤ ~~Excitation~~ Excitation Table
- ⑥ State diagram



① S-R flip-flop

⇒ CKT Diagram latch



NAND
 ↳ any 1/0
 low 0/p high 1
 latch on 1-1
 ↳ Hold state

(clk=0 Then ff will remain on previous state)

⇒ Truth Table

clk=1

S	R	Qn	Qn-bar	Remark
0	0	Qn	Qn-bar	Hold
0	1	0	1	RESET
1	0	1	0	SET
1	1	1	1	Invalid

o/p ka zero hona means → RESET

⇒ characteristic Table

$$Q_{n+1} = S + \bar{R}Q_n$$

⇒ Characteristic Table

	S	R	Q_n	Q_{n+1}	
hold	0	0	0	0	hold same
	0	0	1	1	
Reset	0	1	0	0	Reset both 0
	0	1	1	0	
set	1	0	0	1	set
	1	0	1	1	
	1	1	0	X	
			1	X	

⇒ Char. Equation

$$Q_{n+1}(S, R, Q_n) = \sum m(1, 4, 5) + \sum d(6, 7)$$

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

$$Q_{n+1} = S + \bar{R}Q_n$$

★ ★ Remember

⇒ Excitation Table

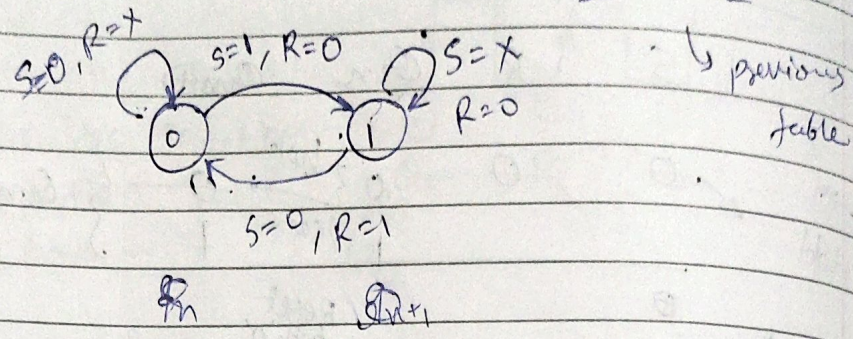
X → Don't care

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

SR

0 ↓ S ↑ R

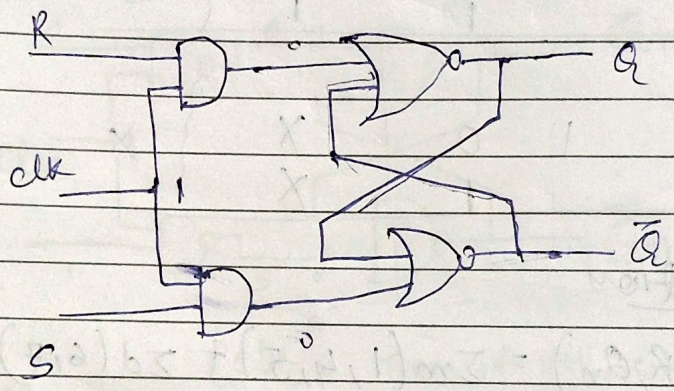
⇒ Excite state diagram



SR flip flop using NOR

Ques

Design the write truth table for circuit given



clk=0

SORE

S	R	Q	Q'	Reming	Qnt
0	0	Q	Q'	Toggel	Qn
0	1	0	1	Reset	0
1	0	1	0	set	1
1	1	0	0	Invalid	X

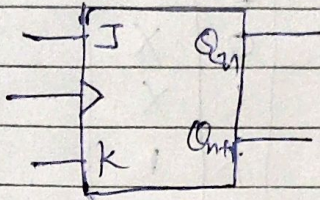
Draw back

↳ only 3 condition are working
 S=1 R=1 not valid

∴ JK flip flop

M T W T F S S	
Page No.:	YOUVA
Date:	

⑦ symbol



J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

→ Hold

→ reset

→ set

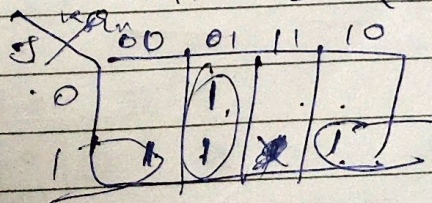
→ Toggle

⇒ Chara. Table

J	K	Q	Q _{nt+1}	
0	0	0	0	} Hold
		1	1	
0	1	0	0	} reset
		1	0	
1	0	0	1	} set
		1	1	
1	1	0	1	} Toggle
		1	0	

⇒ Chan Equation

$$Q_{n+1} = \sum m(1, 2, 5, 6)$$

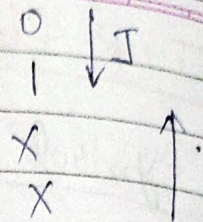


$$Q_{n+1} = J \bar{Q}_n + K Q_n$$

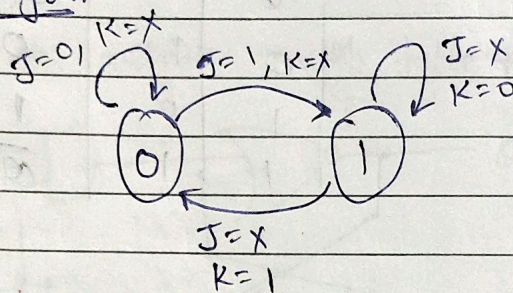
JK is also called universal ff.

⇒ Excitation Table

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



⇒ State diagram



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

SR flip flop → T flip flop
D flip flop

Ques Write characteristic Table, char. equation, excitation Table and state diagram of CJ FF whose truth Table is given below

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

① char. Table

C	J	Q	Q_{n+1}	
0	0	0	1	} set
		1	1	
0	1	0	0	} reset
		1	0	
1	0	0	1	} toggle
		1	0	
1	1	0	0	} Hold
		1	1	

② char equation

$$Q_{n+1} = \sum m(0, 1, 4, 7)$$

~~CJ~~

C	J	00	01	10	11
0	0	1	1		
1	0	1		1	
	1				

$$Q_{n+1} = C J \bar{Q} + \bar{C} \bar{J} Q + C \bar{J}$$

③ Excitation Table

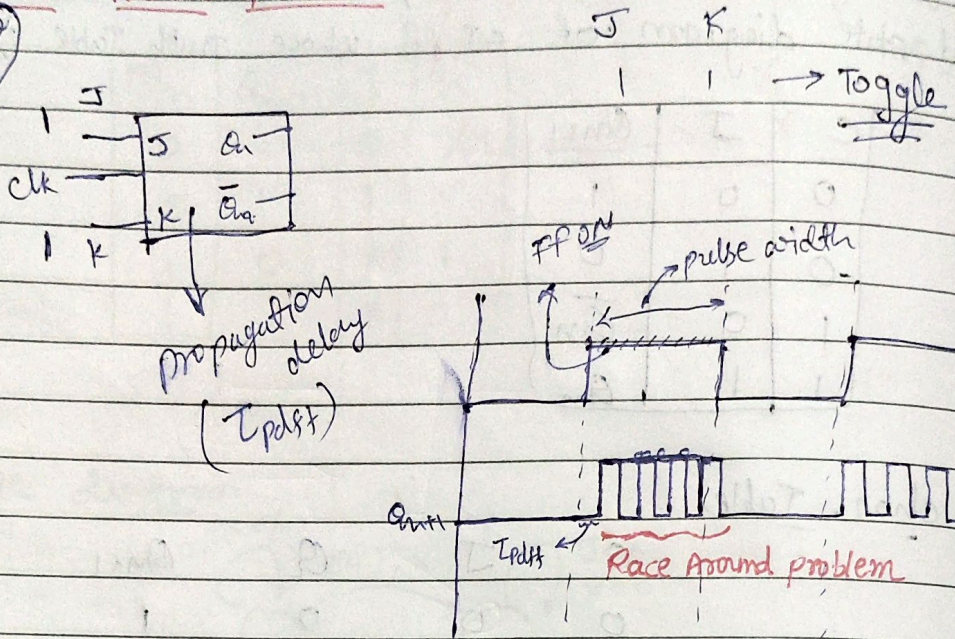
Q_n	Q_{n+1}	C	J
0	0	X	0
0	1	X	1
1	0	0/1	0
1	1	0/1	0/1

$C=0, J=0$
 $C=1, J=1$
 $C=0, J=1$
 $C=1, J=0$

① → level sensitive JK FF suffer from the problem of Race Around

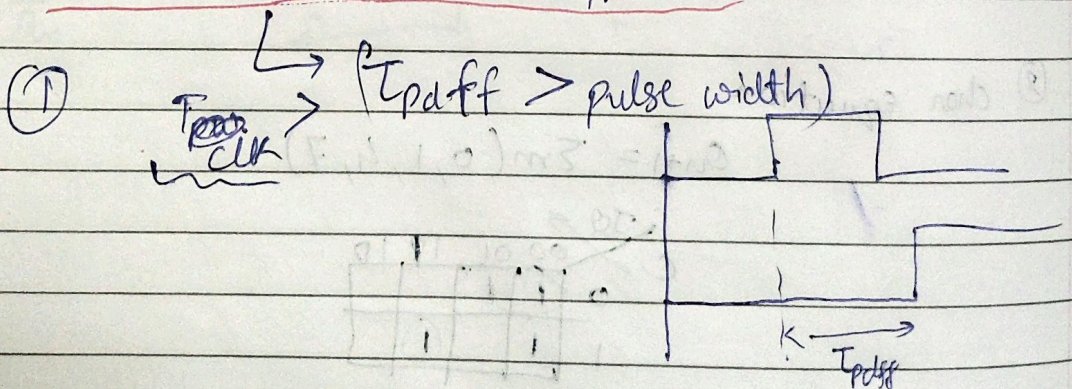
⇒ Race Around problem

Thapped brother example



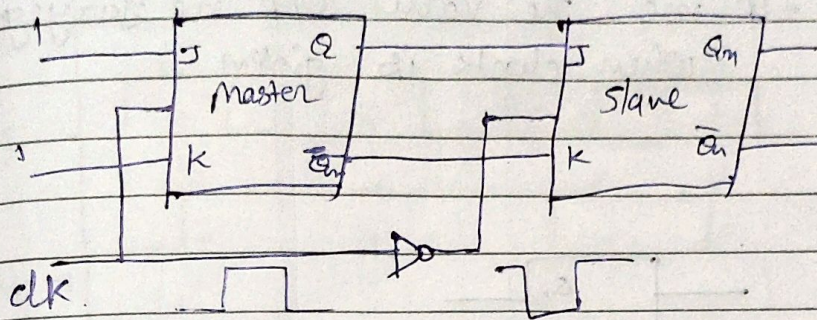
→ whenever JK is level triggered $J=K=1$ is applied then within duration of pulse width output of the ff Toggle so many times are called Race-Around problem

To Avoid Race Around problem



② By master slave ff.

③ ~~Master~~ Master Slave flip flop



- ① o/p of master is given as input of slave
- ② inverted clock is given to the slave.
- ③ Master slave flip-flop is used to store "single bit"

⇒ Tab clk positive he

↳ master ON Rahega and Toggle karoge & master ka

Tab clk of negative hogi (slave active)

Tab master ka o/p
 either 0, 1 or 1, 0 & hoga
 if 0, he toh o/p 0, 1 aayega
 agar 1, 0 he to set 1, 0 aayega

- ④ slave is following o/p of master when clock applied to slave.

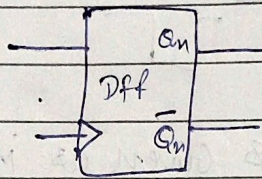
$$I/P = O/P \text{ (given clock)}$$

$$Q_{n+1} = D$$

④ D-flip-flop {Data, Transparent, Delay ff}

→ I/p me he vali o/p me jayega when clock is given.

Symbol



Truth Table

Clock enable

D	Q_{n+1}
0	0
1	1

Char. Table

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

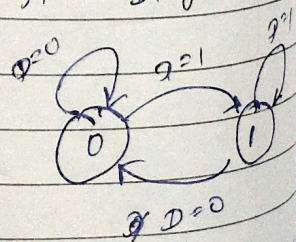
Char. Equation

$$Q_{n+1} = D$$

Excitation Table

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

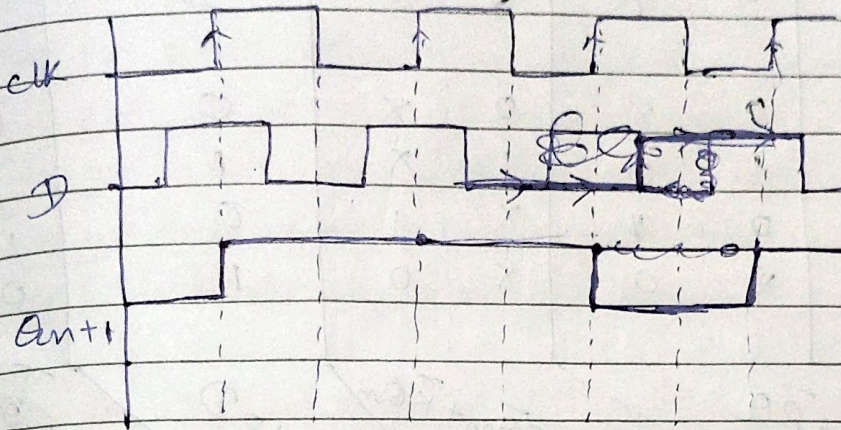
State Diagram



$$Q_{n+1} = JQ_n + \bar{K}Q_n$$

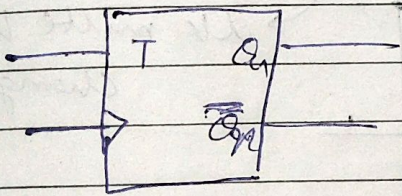
$$T.f.f \Rightarrow Q_{n+1} = Q_n \oplus T$$

D flip-flop:



(5) T-Flip flop {Toggle flip flop}:

$\hookrightarrow$ I/p 0 $\rightarrow$ O/p Hold
 $\hookrightarrow$ I/p 1 $\rightarrow$ O/p Toggle.



Truth table

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

③ char. Table

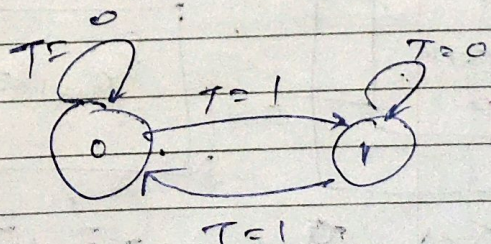
T	Q_n	Q_{n+1}
0	0	0
	1	1
1	0	1
	1	0

⑤ Excitation Table

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

④ char. Equation

$$Q_{n+1} = T \oplus Q_n$$





Excitation Table

Qn	Qn+1	SR flip flop		JK flip flop		D flip flop	T flip flop
		S	R	J	K	D	T
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0

char equation

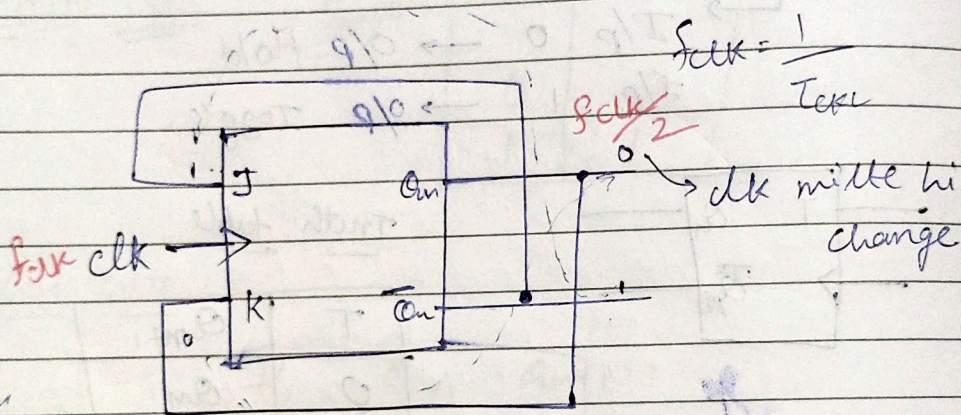
$$Q_{n+1} = S + \bar{R}A$$

$$Q_{n+1} = J\bar{A} + \bar{K}A$$

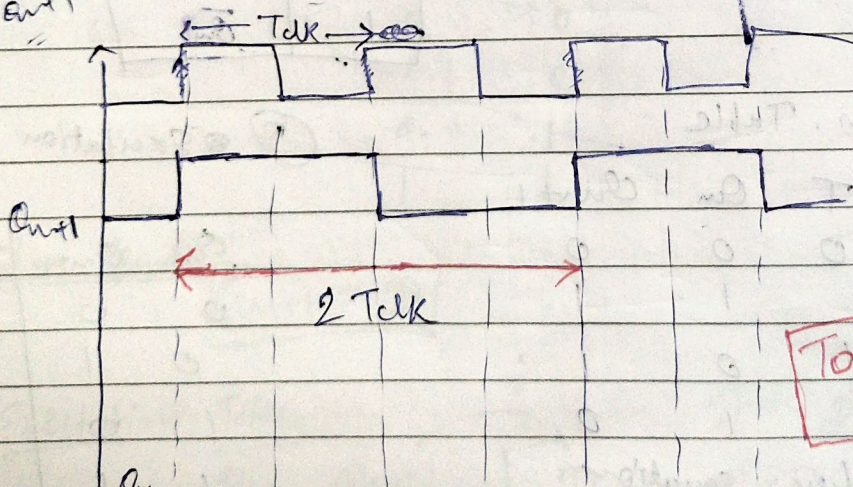
$$Q_{n+1} = D$$

$$Q_{n+1} = \bar{Q}_n$$

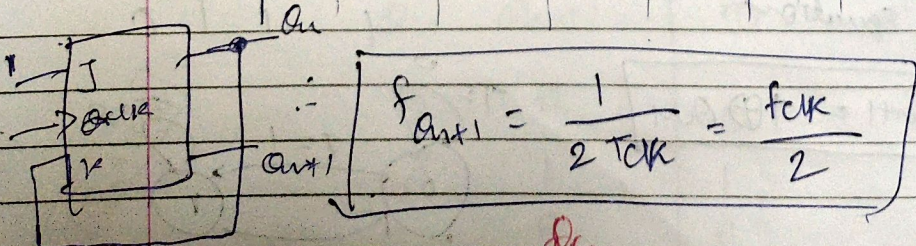
Ques



Get signal Qn+1



Toggle mode



$$f_{Q_{n+1}} = \frac{1}{2 T_{clk}} = \frac{f_{clk}}{2}$$

$J = 1$
 $K = Q_n$
 $Q_{n+1} = J\bar{A} + \bar{K}A$
 $= 1\bar{Q}_n + \bar{Q}_n \cdot Q_n = \bar{Q}_n = Q_{n+1} \rightarrow \text{Toggle}$

Designing of flip-flop.

Step 1:- write the char. Table of desired ff.

Step 2:- write the excitation Table of available ff.

Step 3:- write the logical expression.

Step 4:- minimize the logic expression.

Step 5:- Hardware implementation.

Q7 Design Design JK flip flop using SR ff

① char. of desired

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

Excitation { Q_n, Q_{n+1} given}

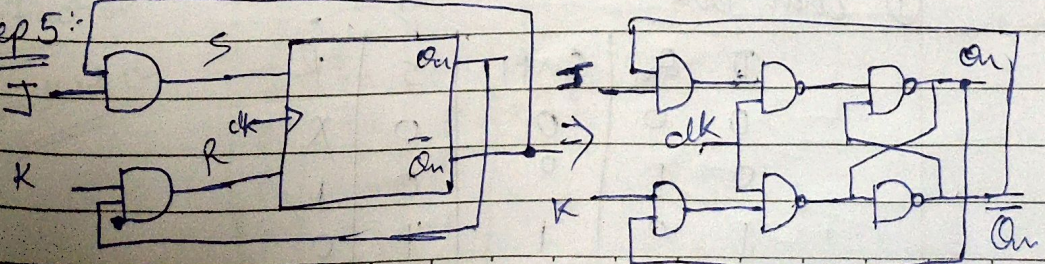
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Step 3:- $S = J \bar{Q}_n$

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

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

Step 5:-



Q Design a T-flip flop using SR flip flop available

Step 1:- char table of T

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

5

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

Step 2:-

$S(T, Q_n)$

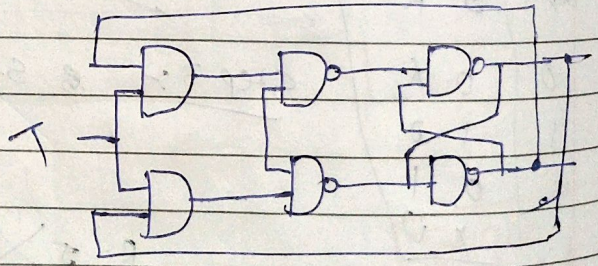
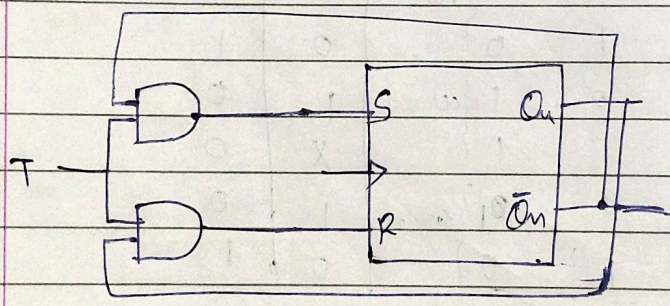
T	Q_n	S
0	0	0
0	1	X
1	0	1
1	1	0

$S = T\bar{Q}_n$

$R(T, Q_n)$

T	Q_n	R
0	0	0
0	1	X
1	0	0
1	1	1

$R = TQ_n$



Ques Design D-Flip flop using SR FF

① Char Table

excitation table

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

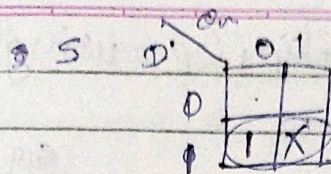
H.W

JK $\rightarrow$ SR, D, T
 D $\rightarrow$ SR, JK, T
 T $\rightarrow$ SR, JK, D

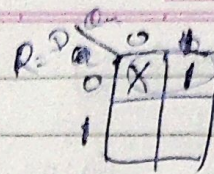
ek don't care
 page 56 don't care

Page No.:
 Date:
 YOUVA

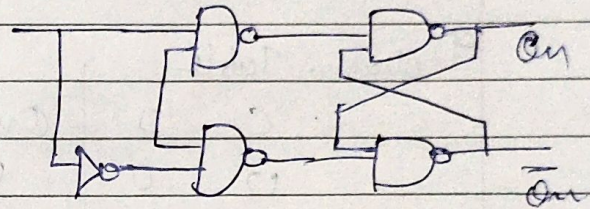
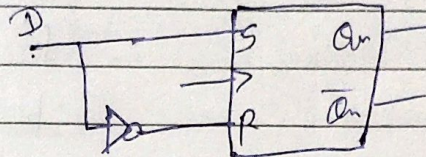
Step: 3



$S = D$



$R = \bar{D}$



Ques Design a D ff using T FF :
 ↳ derive ↳ Available

Justification

① Step 1: char

D	Qn	Qn+1	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

↳ Justification

Method 2

$$Q_{n+1} = T \oplus Q_n$$

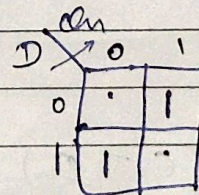
$$= D \oplus Q_n \oplus Q_n$$

$$= D \oplus 0$$

$$\Rightarrow Q_{n+1} = D$$

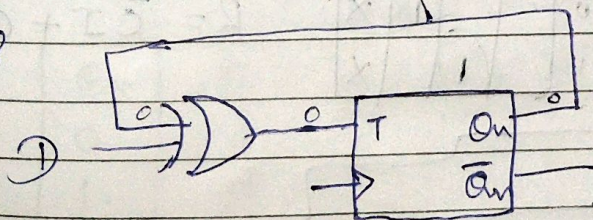
③ Step 3: equation

$T(D, Q_n)$



$T = D \oplus Q_n$

flip flop



$D \quad Q_n \rightarrow D \oplus Q_n \rightarrow T$
 $0 \quad 0 \rightarrow 0 \rightarrow 0$
 $0 \quad 1 \rightarrow 1 \rightarrow 1$
 $1 \quad 0 \rightarrow 1 \rightarrow 1$
 $1 \quad 1 \rightarrow 0 \rightarrow 0$
 When $T=0$
 it hold prev.

Justification

Method 1:

D	Qn	Qn+1
0	0 $T=0$	0
0	1 $T=1$	0
1	0 $T=1$	1
1	1 $T=0$	1

T flip flop hold

T flip flop hold

Ques Design CJ flip flop using SR ff:

C	J	Q_{n+1}	$\bar{Q}_n$	Q_n	S	R
0	0	1	0	0	0	X
0	1	0	0	1	1	0
1	0	$\bar{Q}_n$	1	0	0	1
1	1	Q_n	1	1	X	0

① char. Table

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

② $S(C, J, Q_n)$

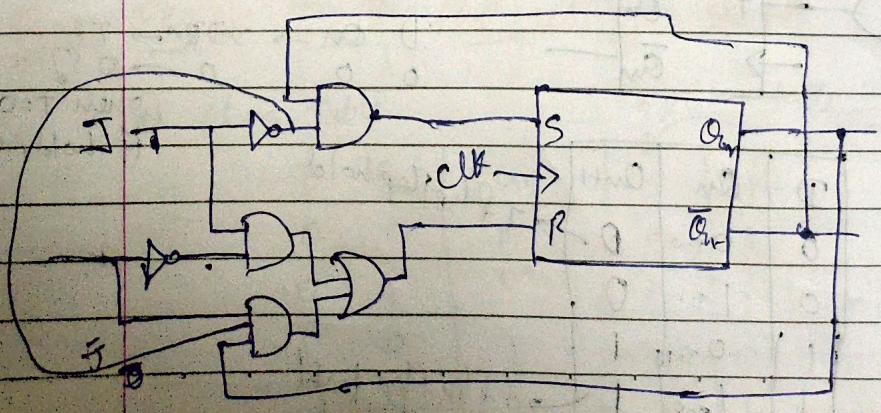
C	$\bar{Q}_n$	Q_n
0	1	X
1	1	X

$S = \bar{J}\bar{Q}_n$

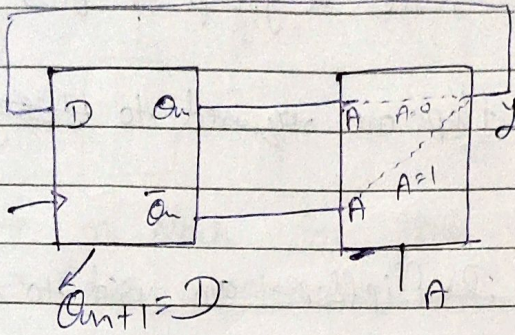
$R(C, J, Q_n)$

C	$\bar{Q}_n$	Q_n
0		1
1	1	X

$R = \bar{C}J + C\bar{J}Q_n$



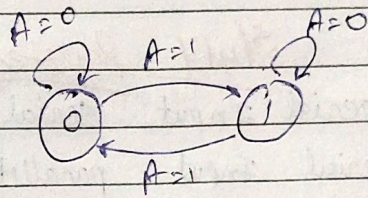
Ques: Sketch the state diagram with variation of A



$$Y = \bar{A}Q_n + A\bar{Q}_n$$

$$= A \oplus Q_n$$

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



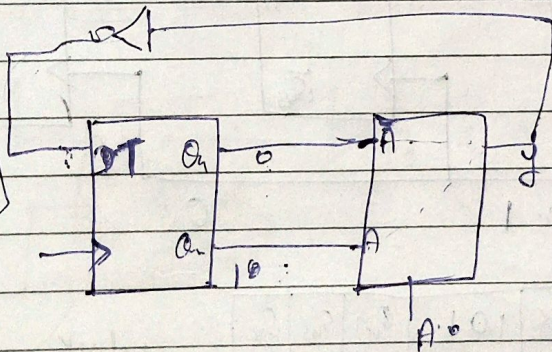
$$D = Y$$

$$Q_{n+1} = Y$$

$$Q_{n+1} = A \oplus Q_n$$

②

Toggle
 $T=0 \rightarrow \text{hold}$
 $T=1 \rightarrow \text{Toggle}$



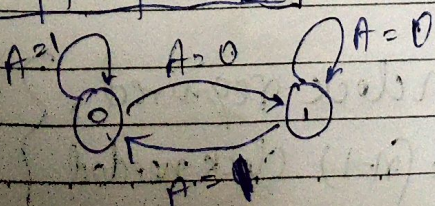
$$Y = \bar{A}Q_n + A\bar{Q}_n$$

$$= A \oplus Q_n$$

$$T = Y$$

$$Q_{n+1} = T \oplus Q_n$$

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



$$Q_{n+1} = (A \odot Q_n) \oplus Q_n$$

$$= A \oplus \bar{Q}_n \oplus Q_n$$

$$= A \oplus 1$$

$$Q_{n+1} = \bar{A}$$

Register

→ used to store a group of bits

→ n flip flop are required to Design Store n bits

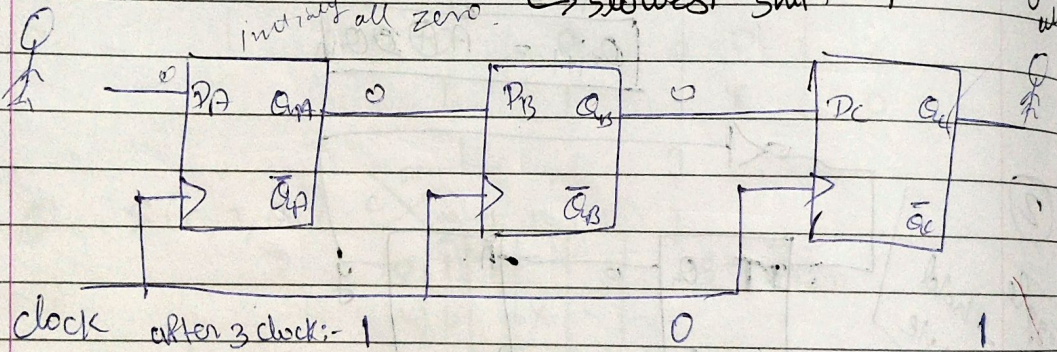
→ Generally D-flip flops are used to design Register

Shift Register

- ① Serial Input serial output shift register (SISO)
- ② Serial Input parallel output shift register (SIPO)
- ③ parallel Input serial output shift register (PISO)
- ④ parallel Input parallel output shift register (PIPO)

① Serial Input serial output (SISO) {3 bit}

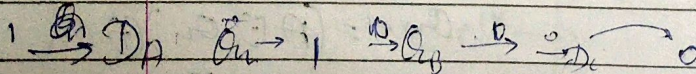
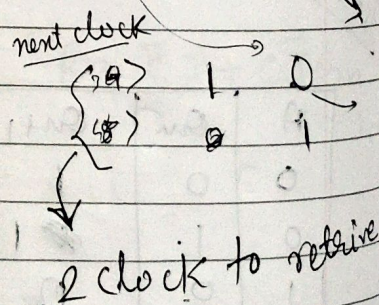
→ slowest shift register among all



101

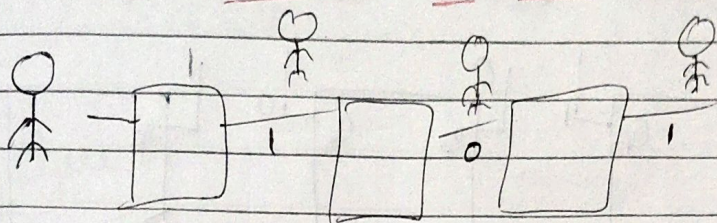
initially is '1' ko utheya and 2 pad me dediya input shifts all

clock	101	QA	QB	QC
0		0	0	0
1		1	0	0
2		0	1	0
3		1	0	1

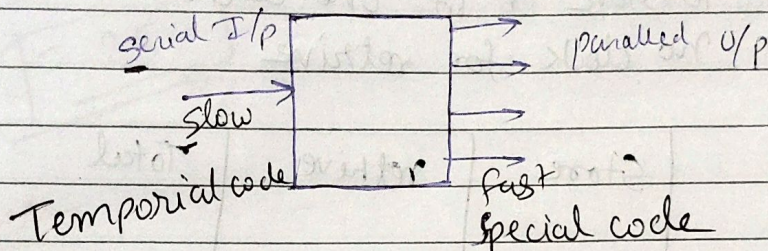


→ To store 'n' bit n clock required, } SISO
 → To retrieve 'n' bit (n-1) clock needed }

② Serial I/p parallel o/p. SR



- n clock for storing
- no clock required to retrieve

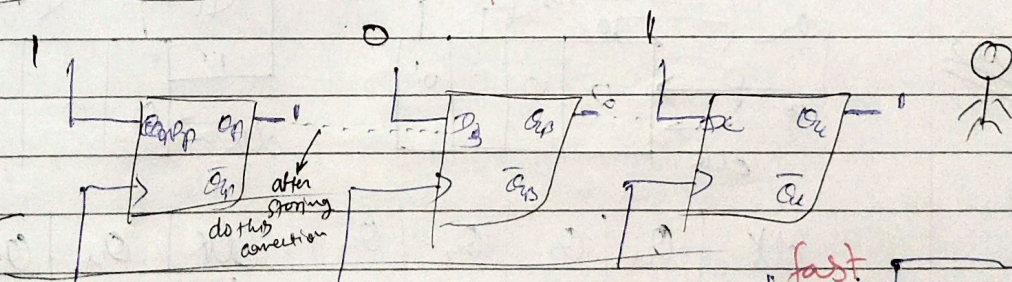


To remember

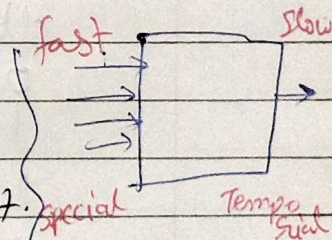
(:- saare 's' ek sath nahi aayenge)

↳ all arrive at same time
but different position.

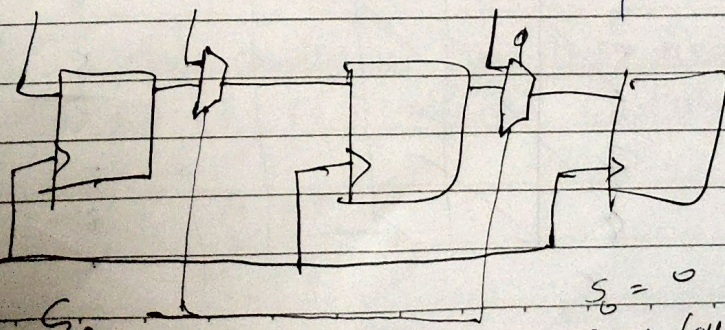
③ parallel input serial o/p shift register



- 1 clock for storing.
- $(n-1)$ clock for ~~sto~~ receive n bit.



~~to be added~~

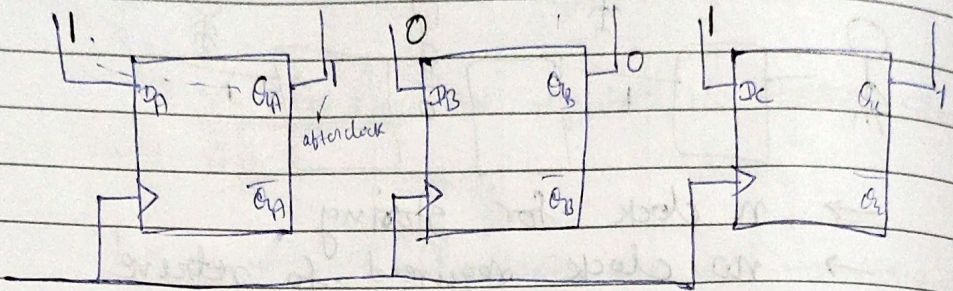


$S_0 = 0$ (input pure)
 $S_0 = 1$ (all shift connect)

50

fastest among all

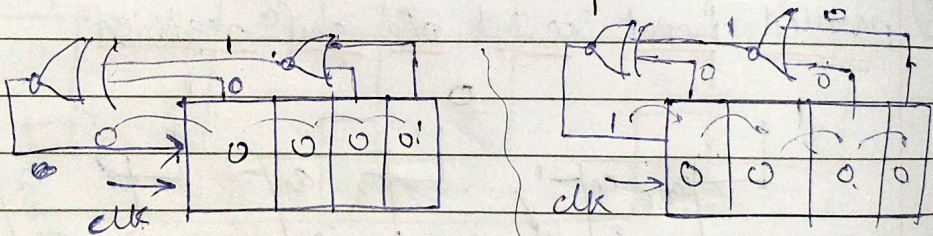
i) parallel input parallel o/p shift register



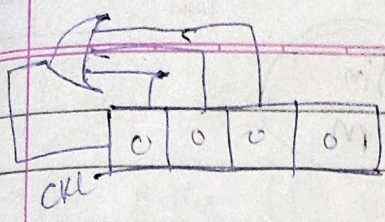
- To store 'n' bit 'one' clock
- no clock for retrieve

	Store	retrieve	Total
SISO	n	n-1	2n-1 → Slowest
SIPO	n	0	n
PIPO	1	n-1	n
PIPO	1	0	1 → fastest

Que



clk	Q ₁	Q ₂	Q ₃	Q ₄	clk	Q ₁	Q ₂	Q ₃	Q ₄
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	0	0	0
2	0	0	0	0	2	0	1	0	0
3					3	1	0	1	0
4					4	1	1	0	1
5					5	1	1	1	1
6					6	1	1	1	1
7					7	1	1	1	1
8					8	1	1	1	1



clk	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1

Duty cycle of

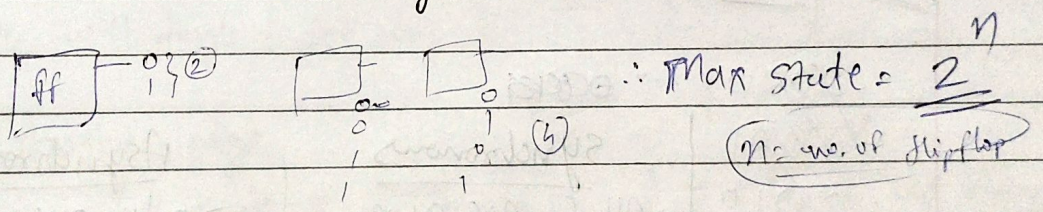
$$D_c = \frac{T_{ON}}{T_{ON} + T_{OFF}}$$

$$= \frac{T}{T + 3T} \times 100\%$$

$$T = 25\%$$

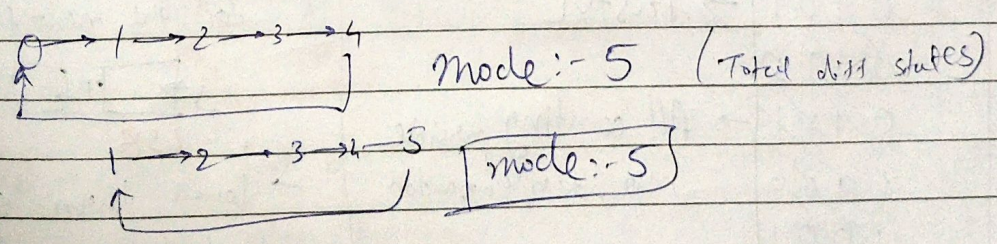
Counter

- used to count number of clocks
- frequency divider ckt.
- 1) synchronous counter
- 2) asynchronous counter



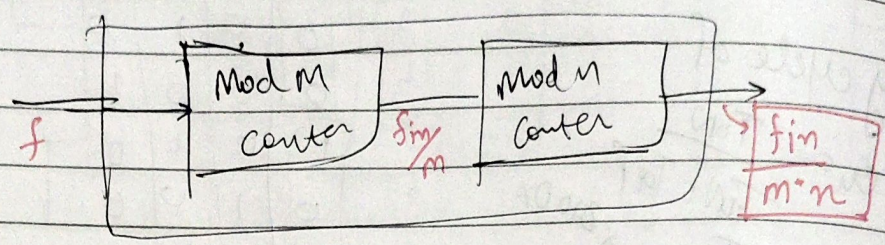
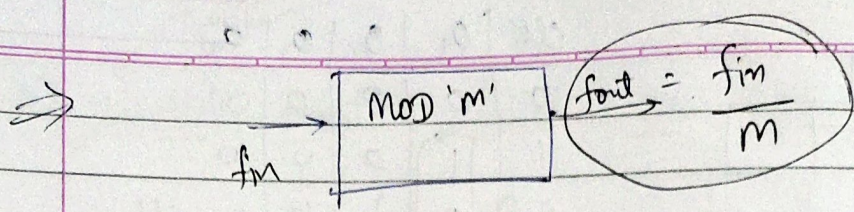
Modulus of counter (MOD)

→ max no. of state that can be used by counter



Q If Mod of the counter is 10. Then find the minimum no. of FF's required to design.

10 → 4 bit
4 FF



→ If mod 5 counter is cascaded with mod 2 counter, then will become.

→ 5×2
 $= \boxed{\text{MOD} - 10}$

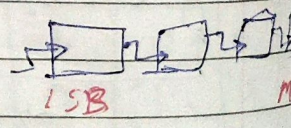
- A) MOD - 10
- B) BCD counter
- C) MOD - 7
- D) Both (A) & (B)

BED Counter

0000	→ 0000
0001	<u>Synchronous</u> → All ff are sync. with same clock.
0010	
0011	
0100	
0101	→ faster
0110	→ All counting possible inc., dec., random
0111	
1000	
1001	

→ no decoding / Transition error

Asynchronous
 → only one ff having external clock & o/p of that ff will be clock for the next ff and so on



→ slower than sync.
 → only inc., decrease
 → decoding / Transition error

Example → Ring counter, Johnson counter Example → Ripple counter

(0 → -ve
- → +ve)

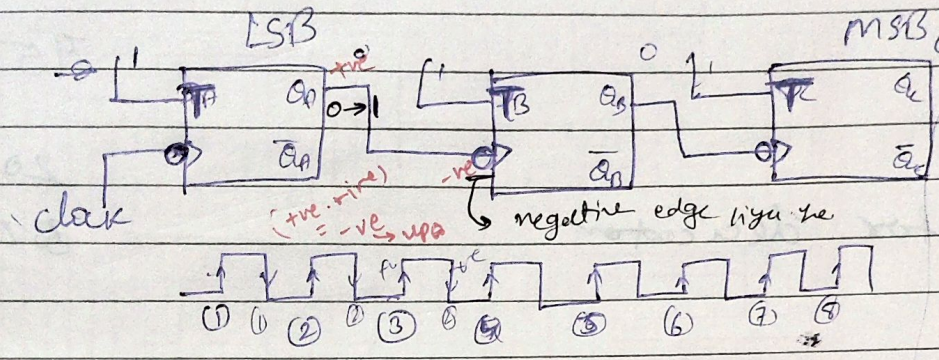
+ve → down
-ve → up

OS → new
note book
me aage ka

Ripple Counter

- All the ff's working in toggle mode (T-mode)
- 'n' ff are required for 'n' bit counter.

① 3 bit Ripple counter



clk	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0
9	0	0	1

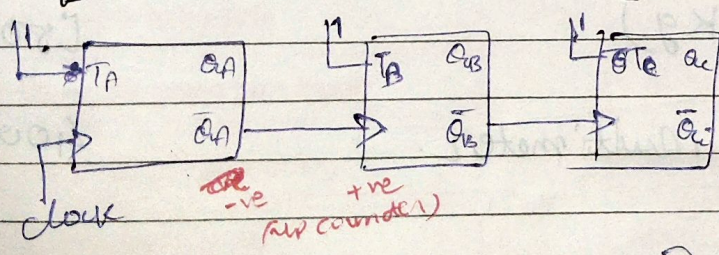
QA will toggle for every five edge of external clock

QB will toggle for every QA (1 → 0)

QC will toggle for every QB (1 → 0)

if positive edge.
→ mode-8 down ripple

if -mode increase.
→ mode-8 up ripple counter



clock	QC	QB	QA
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0
9	0	0	1

QA → change for every clock

QB → change when QA (0 → 1)
∴ QA (1 → 0)

QC → change when QB (0 → 1)
QB (1 → 0)