

## Number System

→ Decimal, → Binary, → octal, → Hexadecimal  
(10)                   (2)                   (8)                   (16)

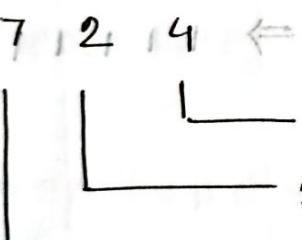
## Binary to Decimal

Binary to Decimal

$$\begin{array}{r}
 101011 \\
 \times 2^0 = 1 \\
 \times 2^1 = 2 \\
 \times 2^2 = 4 \\
 \times 2^3 = 8 \\
 \times 2^4 = 16 \\
 \times 2^5 = 32 \\
 \hline
 & 43_{10}
 \end{array}$$

## Octal to Decimal

Octal to Decimal

$1 \text{ } 0 \text{ } 7 \text{ } 1 \text{ } 2 \text{ } 1 \text{ } 4 \leftarrow$ 	$  \begin{array}{r}  1 \quad 0 \quad 7 \quad 1 \quad 2 \quad 1 \quad 4 \\  \times 8^0 \qquad \qquad \qquad = 4 \\  \hline  0 \quad 56 \quad 56 \quad 16 \quad 4 \\  \hline  468_{10}  \end{array}  $
---	--

$\rightarrow A = 10, B = 11, C = 12, D = 13, E = 14$   
 $F = 15$

## Hexadecimal to decimal

$$\begin{array}{r}
 A \quad B \quad C \\
 | \quad | \quad |
 \\ 
 C \rightarrow 12 \times 16^0 = 12 \\
 B \rightarrow 11 \times 16^1 = 176 \\
 A \rightarrow 10 \times 16^2 = 2560 \\
 \hline
 2748
 \end{array}$$

## Decimal to binary

$$\begin{array}{r}
 125 \\
 \hline
 2 | 62 \qquad 1 \\
 \hline
 2 | 31 \qquad 0 \\
 \hline
 2 | 15 \qquad 1 \\
 \hline
 2 | 7 \qquad 1 \\
 \hline
 2 | 3 \qquad 1 \\
 \hline
 1
 \end{array}
 \Rightarrow 11101$$

$P_1 = E, C_1 = F, S_1 = 5, H = 8, O_1 = A \leftarrow$   
 $\neg S_1 = E$

## Octal to Binary (421)

$$705 = 111000101_2$$

$$\begin{array}{r}
 7 \\
 \downarrow \\
 4 \ 2 \ 1
 \end{array}
 \qquad
 \begin{array}{r}
 5 \\
 \downarrow \\
 4 \ 2 \ 1
 \end{array}$$

~~100~~

## Hexadecimal to Binary (8 4 2 1)

```

graph TD
    A((A)) --- 0((0))
    A --- F((F))
    0 --- 0001((0001))
    0 --- 0000((0000))
    F --- 1010((1010))
    1010 --- 1111((1111))

```

	P	G	S	I	21
0		11	3		21
1		88		21	
			2		

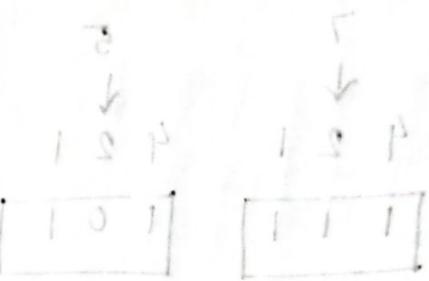
$$\Rightarrow 000100001010111_2$$

$$z(\Gamma \otimes \mathcal{E}) = \frac{\Gamma \Gamma \Gamma, \mathcal{E} \otimes \mathcal{E}, \Gamma \Gamma \mathcal{E}}{\Gamma \quad \mathcal{E} \quad \mathcal{E} \quad \Gamma}$$

$$S_1(000) = \frac{1101101000}{(0)11\quad (0)11\quad 000\quad 0}$$

Decimal to Octal

$$\begin{array}{r} 1234 \\ 8 \overline{)154} \\ 8 \overline{)19} \\ 8 \overline{)2} \\ \hline \end{array} = 23228$$



Decimal to Hexadecimal

$$\begin{array}{r} 1234 \\ 16 \overline{)617} \\ 16 \overline{)38} \\ \hline 2 \end{array}$$

$$\begin{array}{r} A01 \\ / \backslash \\ 1111010100001000 \\ / \backslash \\ 010100001000 \end{array}$$

Binary to Octal

$$\begin{array}{r} 001, 011, 010, 111 \\ | \quad 3 \quad 2 \quad 7 \\ = (1327)_8 \end{array}$$

Binary to Hexadecimal

$$\begin{array}{r} 0010, 1011, 1011 \\ 2 \quad 11(B) \quad 11(B) \\ = (2B0)_{16} \end{array}$$

## Decimal to Octal

$$\begin{array}{r}
 8 \Big| 1 \ 2 \ 3 \ 4 \\
 8 \Big| 1 \ 5 \ 4 \qquad 2 \\
 8 \Big| 1 \ 9 \qquad 2 \\
 8 \Big| 9 \qquad 1 \\
 \hline
 2 \qquad 3
 \end{array}
 \Rightarrow (2 \ 3 \ 2 \ 2)_8$$

\* দশমিক এর পরের  
রাশিতে value base দিয়ে গুণ  
রয়েলে remainder পাবো।

## Decimal to Hexadecimal

$$\begin{array}{r}
 16 \Big| 1 \ 2 \ 3 \ 4 \\
 16 \Big| 7 \ 7 \qquad 2 \\
 16 \Big| 4 \ 9 \times 1 \\
 16 \Big| 4 \ 9 \times 1 \\
 0 = 4 \ 9 \times 0
 \end{array}
 \Rightarrow 4 \ 13 \ 2$$

$$\Rightarrow (4D2)_{16}$$

\* Notes \*

\* Decimal এ সেতে হলে Base দিয়ে গুণ

\* Decimal দ্বারা সেতে হলে Base দিয়ে গুণ

\* Octal or Hexa থেকে Binary  $\Rightarrow$  3 or 4 bit  
দিয়ে ডাগ

\* Octal or Hexa conversion Binary হলে  
সেতে হবে

\* For fraction  $\Rightarrow$  Decimal এ সেতে হলে

দশমিক এর আগে Left থেকে power start 2<sup>-1</sup>

ଦକ୍ଷାମିର୍ତ୍ତ ଏବଂ ପର୍ଯ୍ୟନ୍ତ Right (ଯେତେବେଳେ) ହିସ୍ତି power  
start ହବେ,

## Binary to Decimal

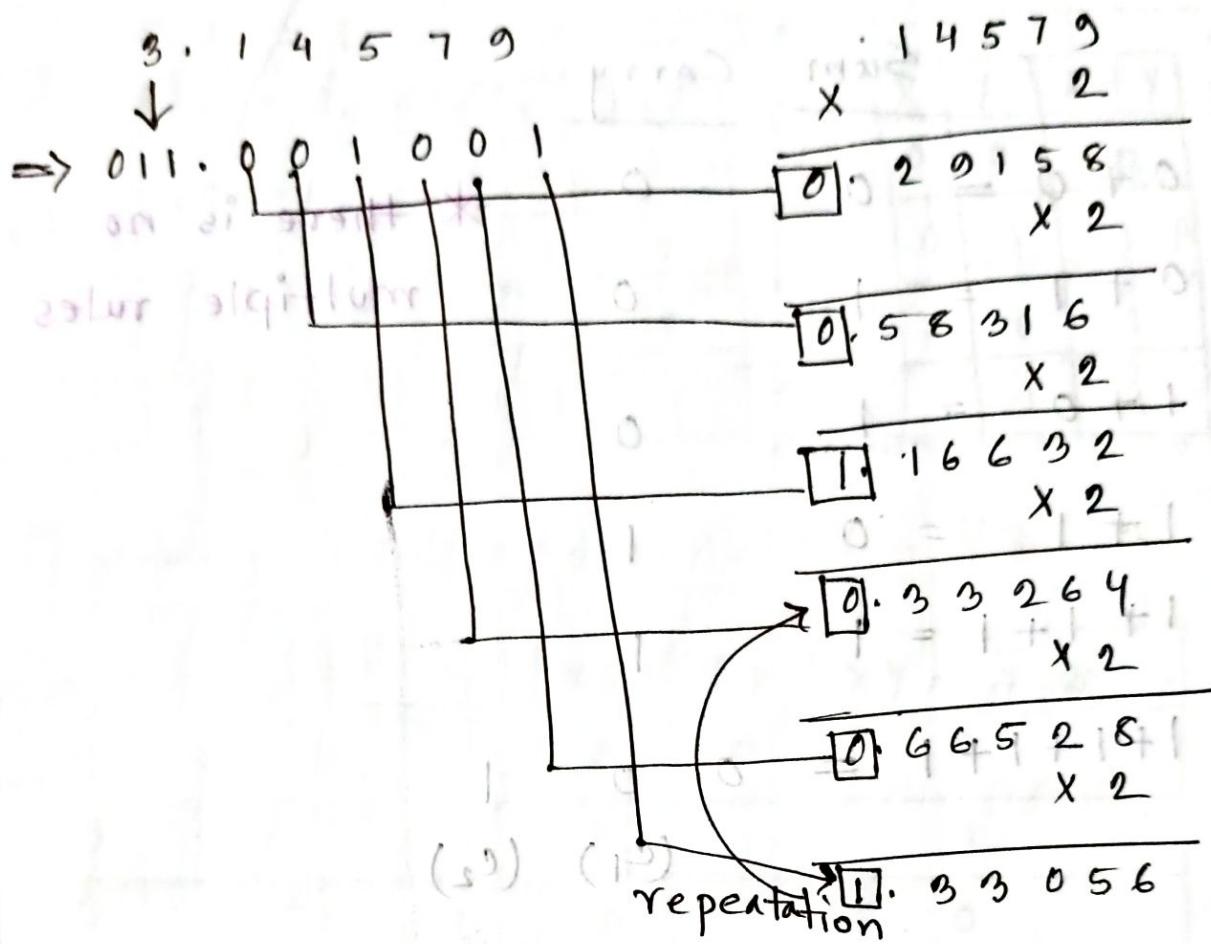
+1er Prol.  $\Leftrightarrow$  Fermat  $x^2 - x + 1 = 2$  ro. Integ. \*

~~(2. G875)~~

6. Barrier northernmost extent to 1000 ft to  
the animal. At 1000

miss one & lamissa < miss one rot \*  
miss one & lamissa < miss one rot \*

## Decimal to Binary



Exercise  $\Rightarrow$  practise for Exam

$$\begin{array}{r}
 101 \\
 \hline
 1011101 \\
 \overline{0000000} \\
 \overline{1011101} \\
 \hline
 100010111
 \end{array}$$

$$\begin{array}{r}
 010111 \\
 \hline
 111100 \\
 \overline{100100}
 \end{array}$$

## Binary Additional Rule

	<u>Sum</u>	<u>Carry</u>
$0 + 0 =$	0	0
$0 + 1 =$	1	0
$1 + 0 =$	1	0
$1 + 1 =$	0	1
$1 + 1 + 1 =$	1	1
$1 + 1 + 1 + 1 =$	0	1
	(c <sub>1</sub> )	(c <sub>2</sub> )

\* there is no multiple rules

Example :

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

Multiple Example :

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

## truth table

### logical OR Rules (Multiple Rules)

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

AND gate

OR gate

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

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

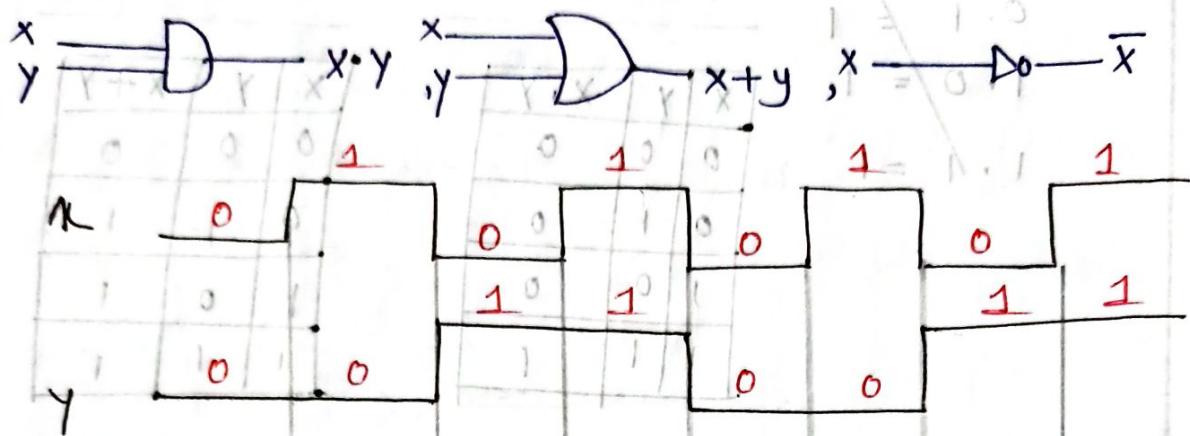
Example :  $F(x, y, z) = XY + \bar{Y}Z$

x	y	z	$\bar{y}$	$XY$	$\bar{Y}Z$	$XY + \bar{Y}Z$
0	0	0	1	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1

\* শুধুমাত্র অসমুক 0 মানলয়ে 0

\* স্থানের অসমুক 1 মানলয়ে 1

## Logic gate symbols and behaviour



(AND)

(OR)

(NOT)

0 1 0 1 0 1 0 1 \*

1 0 1 0 1 0 1 0 \*

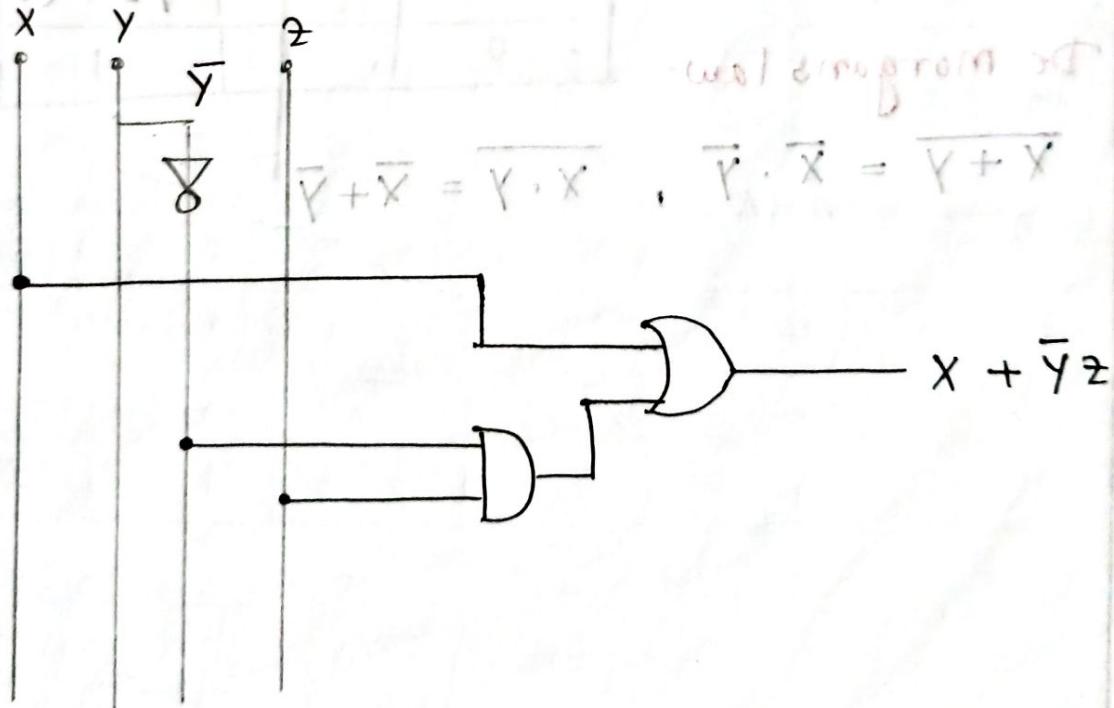
## Logic Diagram

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

$x$	$y$	$z$	$\bar{y}$	$\bar{y}z$	$x + \bar{y}z$
0	0	0	1	0	$x = 0 + x$
0	0	1	1	1	$1 = 1 + x$
0	1	0	0	0	$0 = \bar{x} + x$
0	1	1	0	0	$0 = x + \bar{x}$
1	0	0	1	0	$1 = 1 + 0$
1	0	1	1	1	$1 = x + \bar{x}$
1	1	0	0	0	$1 = x + y = y + x$
1	1	1	0	0	$1 = f + (x + y)$

$$(f'y)x = f(yx)$$

$$fx + yx = (f+y)x$$



## Boolean Algebra Properties

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

$$x + 0 = x$$

$$x + 1 = 1$$

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

$$\bar{x} = x$$

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

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

$$x \cdot x = x$$

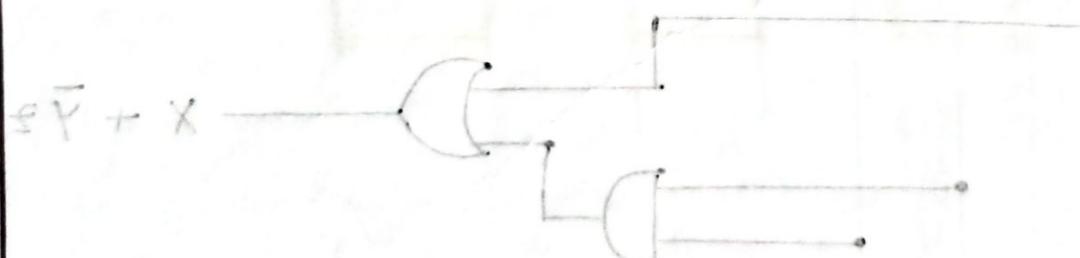
$$x + y = y + x$$

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

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

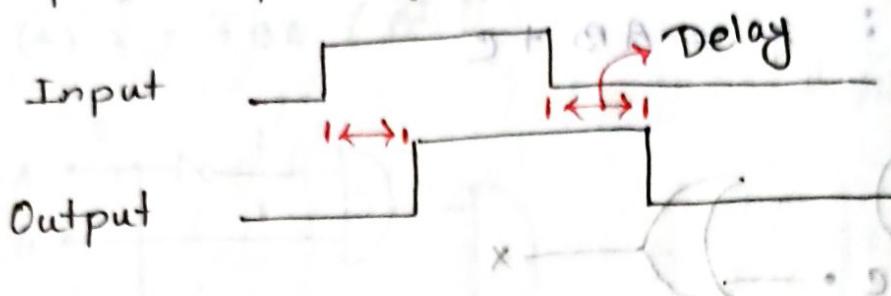
De Morgan's law

$$\overline{x+y} = \bar{x} \cdot \bar{y}, \quad \overline{x \cdot y} = \bar{x} + \bar{y}$$



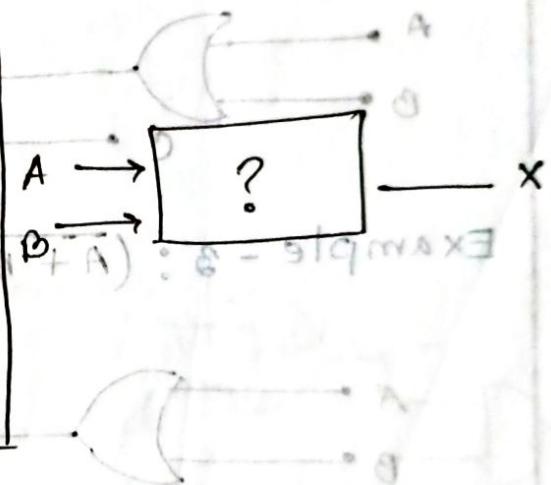
## Gate Delay

Input, Output টেক্স মাঝের বিপরীত।



\* Another type of truth table:

Input		Output
A	B	X
0	0	1
0	1	0
1	0	1
1	1	0

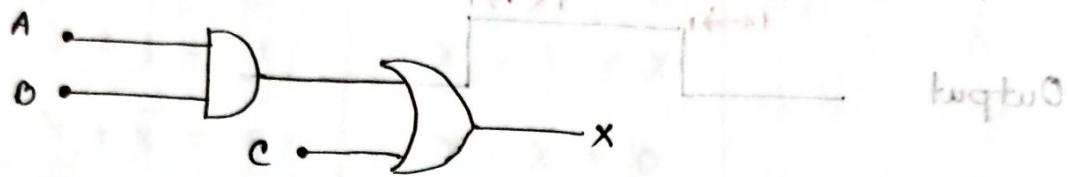


$$A + \bar{A} = X \quad (i)$$

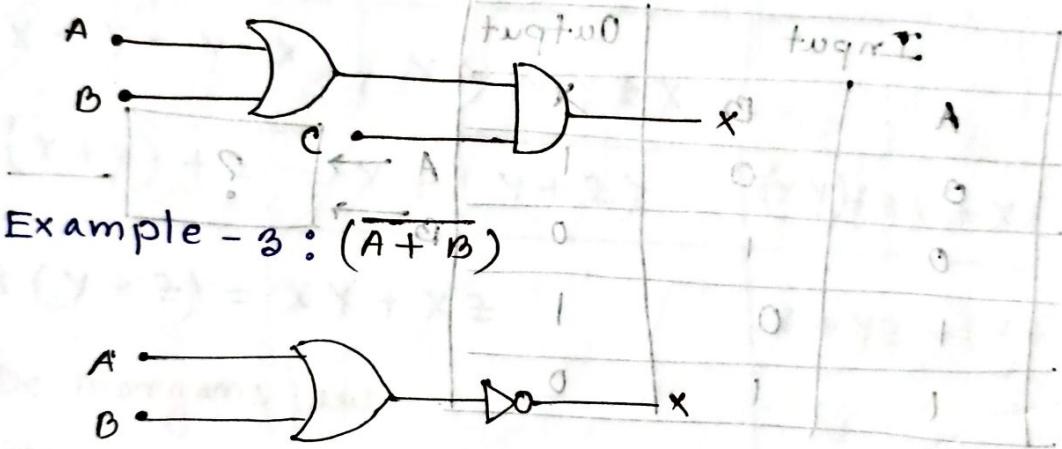


## \* Describing logic circuit Algebraically

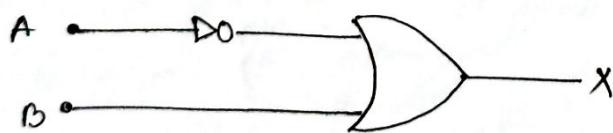
Example - 1 :  $x = AB + C$



Example - 2 :  $x = (A+B)C$

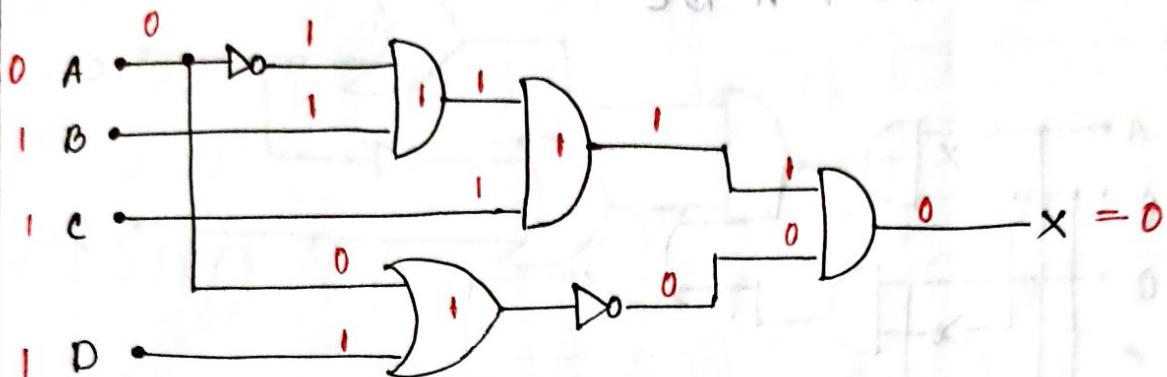


(b)  $x = \overline{A} + B$

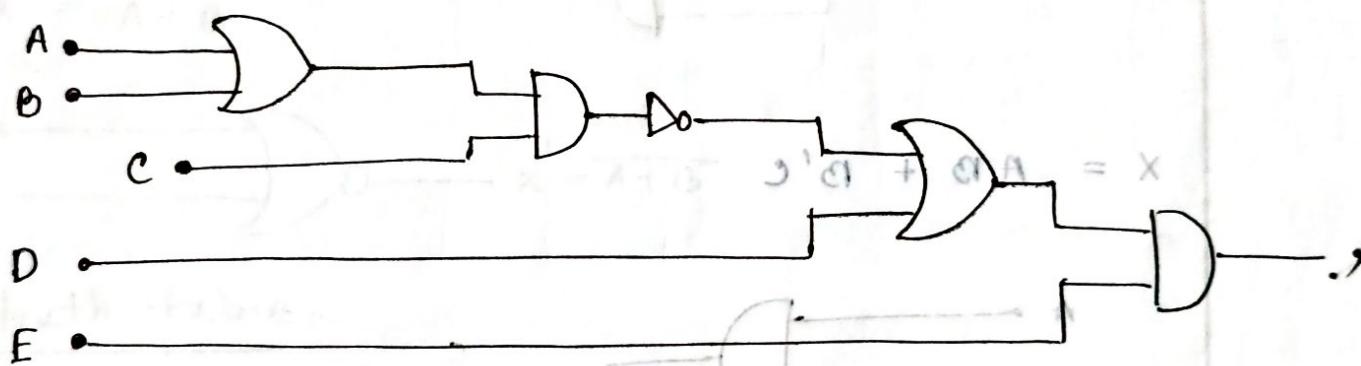


Example - 04

(a)  $X = \overline{A}BC (\overline{A} + D)$



(b)  $[D + (\overline{A} + B)C] \cdot E$



Evaluating logic circuit Output

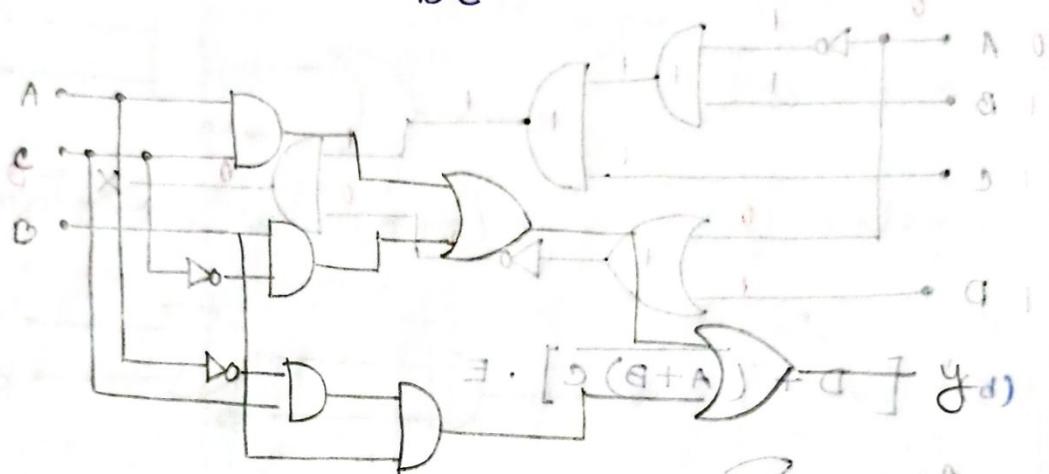
$A=0, B=1, C=1, D=1$

For Ex - 4(a)

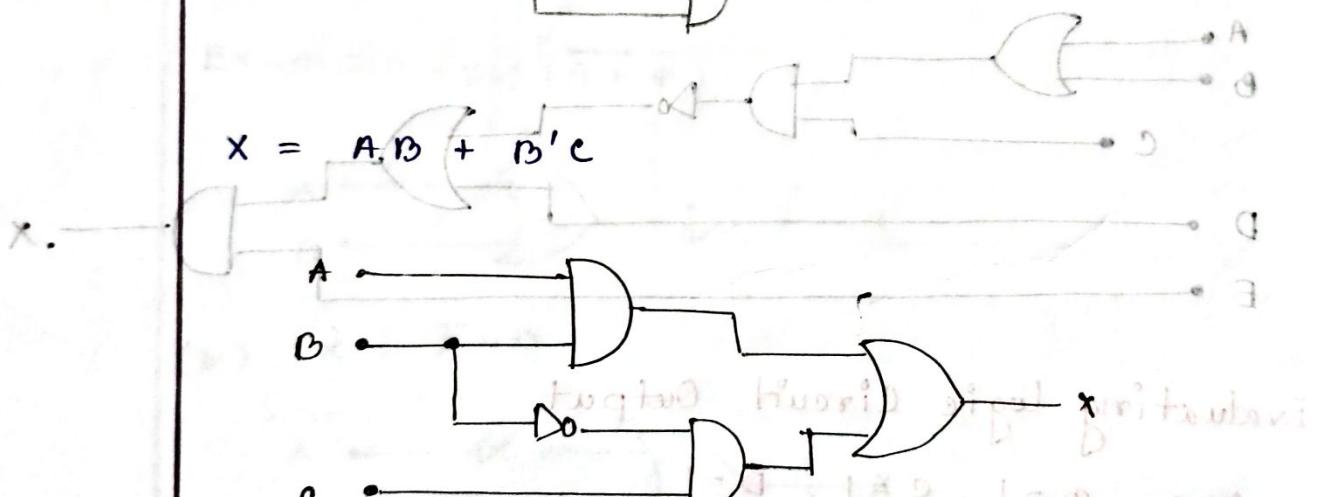
## Implementing Circuits from Boolean Expressions

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

$$Y = AC + BC' + A'B'C$$

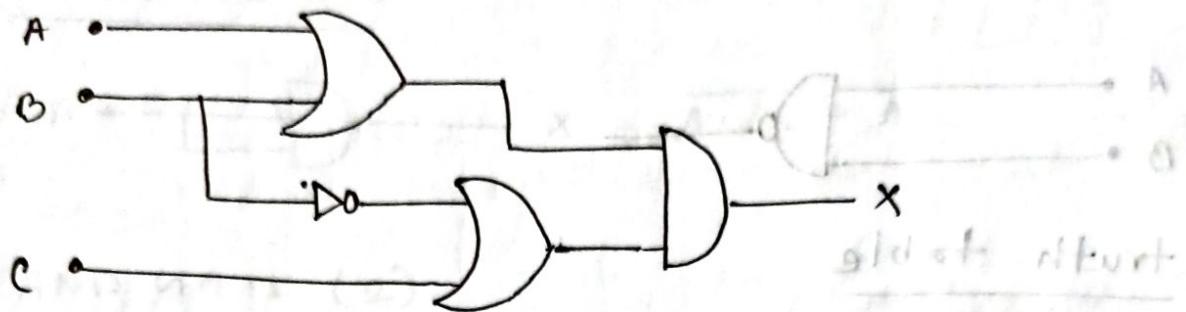


$$X = AB + B'C$$



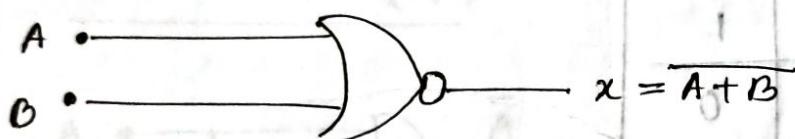
(a)  $H \rightarrow XE$  for

$$x = (A+B)(B'+C)$$



**NOR GATE**

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

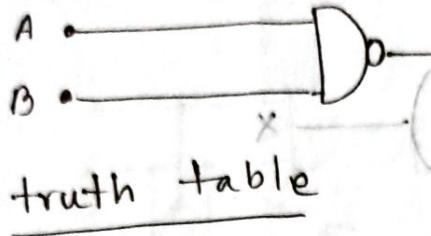


truth table

		OR	NOR
A	B	$A+B$	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

### NAND GATE

$$x = \overline{AB}$$



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

$$(a+a')(a'+a) = 1$$

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

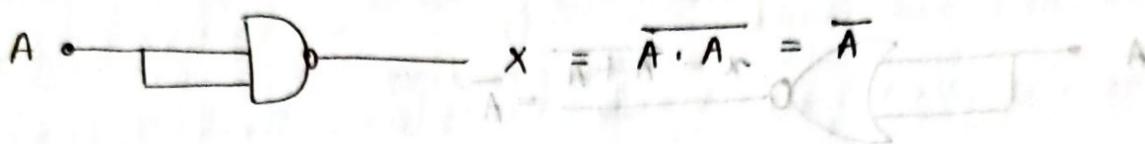


gildat n'tant

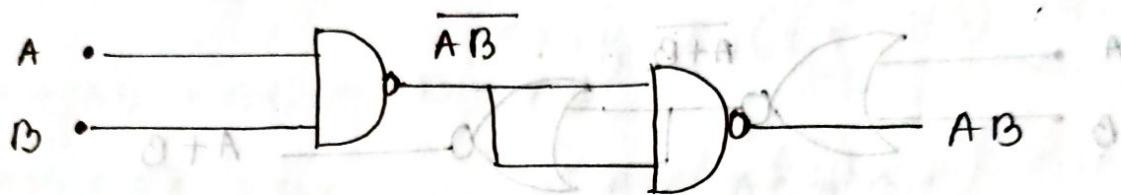
nor	$A$	$B$	$\overline{A+B}$	$\overline{A} \cdot \overline{B}$
1	0	0	1	1
0	1	1	0	0
0	1	0	1	0
0	0	1	1	0

## University of NAND GATES

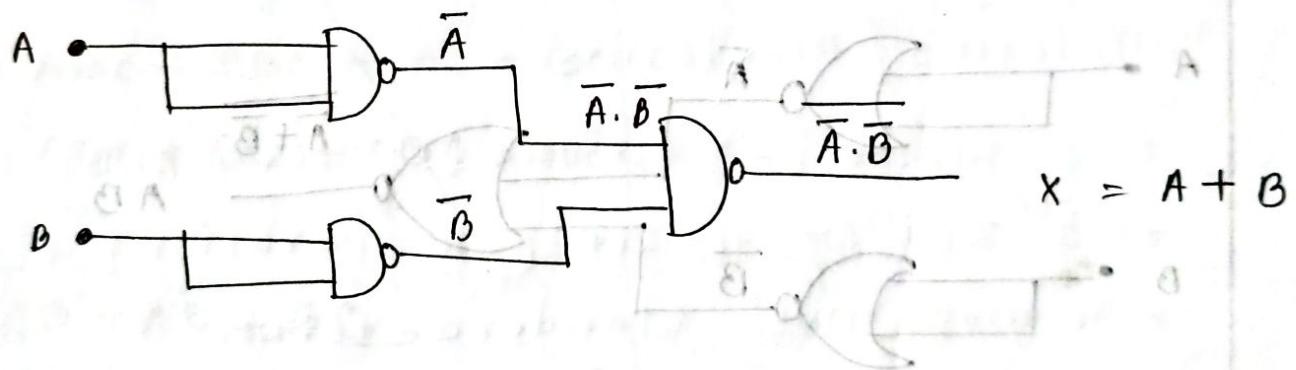
### NOT GATE (1)



### AND GATE (2)



### OR GATE (3)

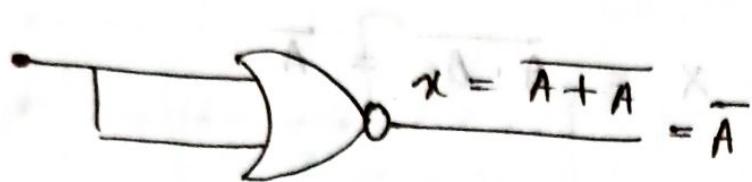


# University of NOR GATES

(1) INVERSION

NOT (I)

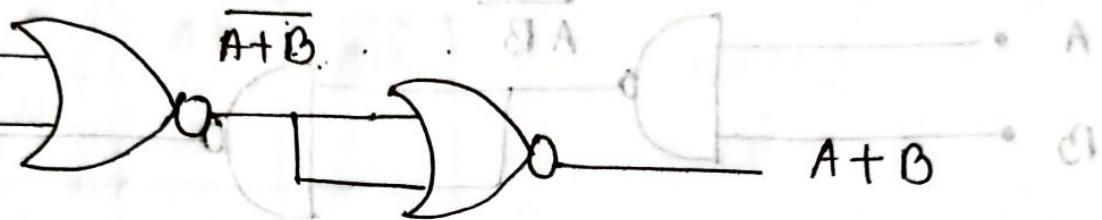
A



OR (II)

(2) INVERSION OR

A  
B



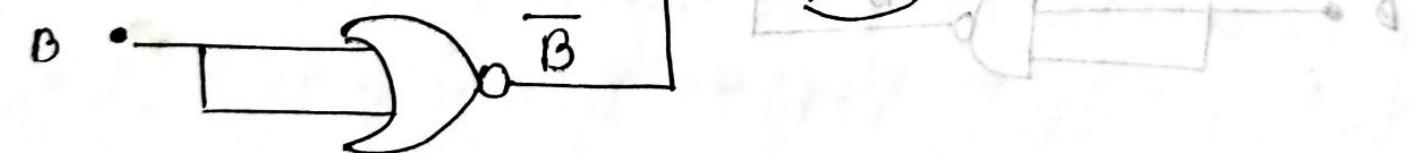
AND (3)

(3) INVERSION AND

A

B

$$+ A = X$$



## Boolean Algebraic Proof

$$A + AB$$

$$\Rightarrow A \cdot 1 + A \cdot B \quad [\text{identity element}]$$

$$\Rightarrow A(1+B) \quad [\text{Distributive}]$$

$$\Rightarrow A \cdot 1 \quad [1+B=1]$$

$\Rightarrow A$  [Proved]

$$AB + A\bar{C} + BC = AB + AC$$

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

$$\Rightarrow AB \cdot 1 + A\bar{C} + BC$$

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

$$\Rightarrow ABC + ABC + A\bar{C} + BC$$

$$\Rightarrow ABC + BC + A\bar{C} + A\bar{C}$$

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

$$= AB + A\bar{C} + BC \cdot 1$$

$$= AB$$

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

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

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

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

$$\Rightarrow AB + BC$$

# complementing Functions

Use DeMorgan's Theorem :

→ Interchange AND, OR operators.

→ Complement each constant and literal

$$\Rightarrow F = \bar{x}\bar{y}^2 + x\bar{y}^2$$

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

$$\Rightarrow h = (\bar{a} + bc)\bar{d} + e$$

$$= (a(\bar{b} + \bar{c}) + d)\bar{e}$$

## Minterms / Maxterms

- \* 0 আবশ্যিক (-) ওবে and (.) ওবে ( AND) for
  - \* 1 রাস্তা (-) বাৰু ওবে ( OR operation) for n
  - \* K-map must standerd form,  $\bar{A}B + \bar{A}A$
  - \* Some of Minterms =  $\sum m_i (m_0 + m_1 + m_2 + \dots)$
  - \* product of Max terms =  $\prod M_i (M_0 \cdot M_1 \cdot M_2 \cdot \dots)$

x	y	z	index	minterm	maxterm
0	0	0	0	$m_0 = \bar{x}\bar{y}\bar{z}$	$M_0 = x+y+z$
0	0	1	1	$m_1 = \bar{x}\bar{y}z$	$M_1 = x+y+\bar{z}$
0	1	0	2	$m_2 = \bar{x}yz$	$M_2 = x+\bar{y}+z$
0	1	1	3	$m_3 = \bar{x}y\bar{z}$	$M_3 = x+\bar{y}+\bar{z}$
1	0	0	4	$m_4 = x\bar{y}\bar{z}$	$M_4 = \bar{x}+y+z$
1	0	1	5	$m_5 = x\bar{y}z$	$M_5 = \bar{x}+y+\bar{z}$
1	1	0	6	$m_6 = xy\bar{z}$	$M_6 = \bar{x}+\bar{y}+z$
1	1	1	7	$m_7 = xyz$	$M_7 = \bar{x}+\bar{y}+\bar{z}$

"Standard Order"

- \* All variables should be present in a minterm or maxterm and should be listed in the same order.

Example : Maxterm  $\rightarrow (a+b+\bar{e})$ ,  $(\bar{a}+b+\bar{e})$  in SF  
 $(b+\bar{a}+e)$  not in SF  
 $(a+e)$  does not contain all variables

\* Some of Minterm

x	y	z	F	Minterm
0	0	0	0	
0	0	1	1	$m_1 = \bar{x} \bar{y} z$
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$m_6 = x y \bar{z}$
1	1	1	1	$m_7 = x y z$

\* Focus on 1  $\rightarrow$  then write equation

$$F = m_1 + m_6 + m_7 = \Sigma(1, 6, 7)$$

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

$$* F(a, b, c, d) = \sum(2, 3, 6, 10, 11)$$

8421

$$= 0010 + 0011 + 0110 + 1010 + 1011$$

$$= \bar{x}\bar{y}\bar{z}\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd$$

(\*)

$$* n(a, b, c, d) = \sum(0, 1, 12, 15) = m_0 + m_1 + m_{12} + m_{15}$$

$$= 0000 + 0001 + 1100 + 1111$$

$$= \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + ab\bar{c}\bar{d} + abcd$$

"product of maxterms"

x	y	z	F	Maxterm
0	0	0	1	
0	0	1	1	
0	1	0	0	$m_2 = x + \bar{y} + \bar{z}$
0	1	1	1	
1	0	0	0	$m_4 = \bar{x} + y + \bar{z}$
1	0	1	1	
1	1	0	0	$m_6 = \bar{x} + \bar{y} + \bar{z}$
1	1	1	1	

Focus on "0"

$$F = M_2 \cdot M_4 \cdot M_6 = \pi(2, 4, 6)$$

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

8 4 2 1

$$F = \pi(1, 3, 6, 11) = m_1 \cdot m_3 \cdot m_6 \cdot m_{11} = \\ b\bar{c}\bar{d} + \bar{b}cd + b\bar{c}d + \bar{b}c\bar{d} = \\ = 0001 + 0011 + 0110 + 1011 \\ = (a+b+c+d) + (a+b+\bar{c}+\bar{d}) + (a+\bar{b}+\bar{c}+d) + (\bar{a}+b+\bar{c}+\bar{d})$$

$$G = \pi(0, 4, 12, 15) = m_0 \cdot m_4 \cdot m_{12} \cdot m_{15} = \\ = 0000 + 0100 + \bar{b}\bar{d}100 + \bar{b}\bar{d}\bar{b}\bar{d} = \\ = (a+b+c+d) + (a+\bar{b}+c+d) + (\bar{a}+\bar{b}+c+d) + (\bar{a}+\bar{b}+\bar{c}+d)$$

(Converting)

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

x	y	z	$\bar{x}$	$\bar{y}$	$\bar{z}$	$\bar{x}\bar{z}$	F = $\bar{y} + \bar{x}\bar{z}$	Minterm	Maxterm
0	0	0	1	1	1	0	1	$m_0 = \bar{x}\bar{y}\bar{z}$	$m_0 = \bar{x}\bar{y}\bar{z}$
0	0	1	1	1	0	0	1	$m_1 = \bar{x}\bar{y}z$	$m_1 = \bar{x}\bar{y}z$
0	1	0	1	0	1	1	1	$m_2 = \bar{x}yz$	$m_2 = \bar{x}yz$
0	1	1	1	0	0	0	0	<del><math>m_3 = x+y+z</math></del>	$m_3 = x+y+z$
1	0	0	0	1	1	0	1	$m_4 = x\bar{y}\bar{z}$	$m_4 = x\bar{y}\bar{z}$
1	0	1	0	1	0	0	1	$m_5 = \bar{x}y\bar{z}$	$m_5 = \bar{x}y\bar{z}$
1	1	0	0	0	1	0	0	<del><math>m_6 = \bar{x}\bar{y}z</math></del>	$m_6 = \bar{x}\bar{y}z$
1	1	1	0	0	0	0	0	<del><math>m_7 = x+y+z</math></del>	$m_7 = x+y+z$

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

## “K-MAP”

“Steps”

→ Standard Form

→ truth table

→ K-map

\* Group - 1, 2, 4, 8, 16

\* nRP বাড়ে বজ্রণে হবে

\* Overlap করে গুপ বাড়ে হলে বৃদ্ধি ঘটবে।

\* বেশি Overlap বজান বজ্রণে হবে।

\* result এর সময় যেটা পার্থিবতন হবে সেটা  
লিখবো না।

\* Decimal value Box এ লিখবো

\* উপর মূল্য, পাশা মূল্যে, 2 IT view দিয়ে  
nRP result লিখবো।

\* main equation এর decimal value

বৃদ্ধি বৃদ্ধি বৃদ্ধি।

Example :

$$F = AB + \bar{A}BC$$

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

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

111 + 110 + 011

$$= m_3 + m_6 + m_7$$

$$= \Sigma (3, 6, 7) \quad [SF \text{ result}]$$

truth table :

A	B	C	$\bar{A}$	$\bar{C}$	$ABC$	$A\bar{B}\bar{C}$	$\bar{A}BC$	F
0	0	0	1	1	0	1	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	1	0	0	0	0
0	1	1	1	0	0	0	1	1
1	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	1	0	1	0	1
1	1	1	0	0	1	0	0	1

18 P.R

G 623373 039 01 Jan 1996

AB	C	S	0	X	Y	1	W	4	5	6	7
00			1	1	0	0	0	0	0	0	0
01			0	0	1	0	1	1	0	0	0
11			1	1	1	1	0	1	1	0	0
10			0	0	0	1	1	0	1	0	0
			4	0	0	5	1	0	1	1	0
			6	1	1	7	0	1	1	1	0
			2	0	0	3	0	1	0	0	1
			1	1	0	0	0	0	0	0	1
			0	0	1	0	0	0	0	0	0
			5	1	0	1	0	0	0	0	0
			4	0	1	0	1	0	0	0	0
			3	0	0	1	0	1	0	0	0
			2	0	0	0	1	0	0	0	0
			1	1	0	0	0	0	0	0	0
			0	0	1	0	0	0	0	0	0

$$hrp-10 = BC$$

$$hrp-21 = AB$$

$$F = AB + BC$$

"practise"

AB	CD	00	01X	11	X10
00		0	1	3	2
01		4	5	7	6
11		12	13	15	14
10		8	9	11	10

\* Minterm এর জন্য  
o মানে variable  
১১ উপর (-) বাবু হবে

$$\rightarrow hrp-3 = B\bar{C}\bar{D}$$

$$hrp-1 = AB$$

$$\rightarrow hrp-2 = A\bar{C}D$$

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

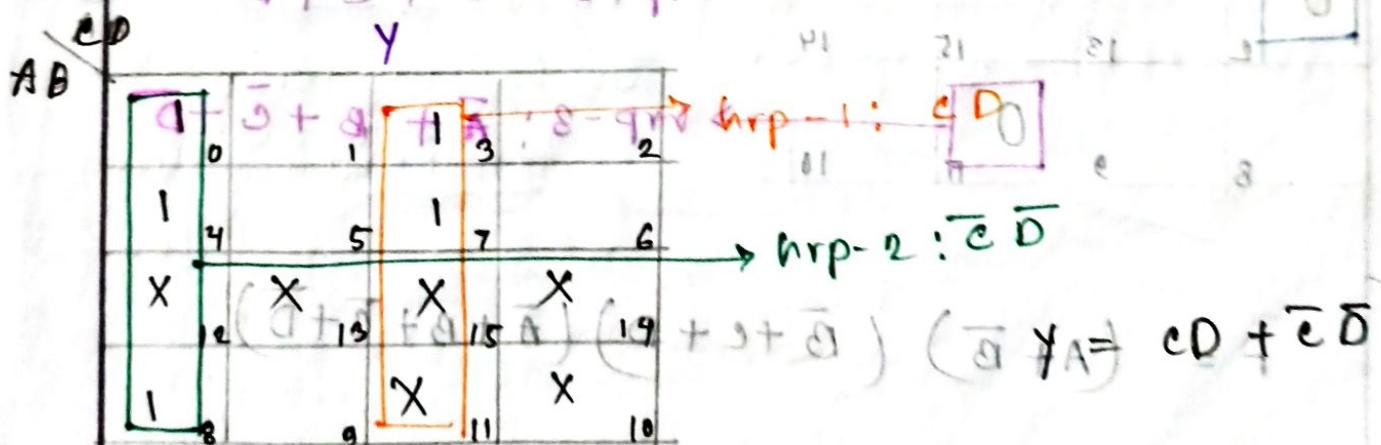
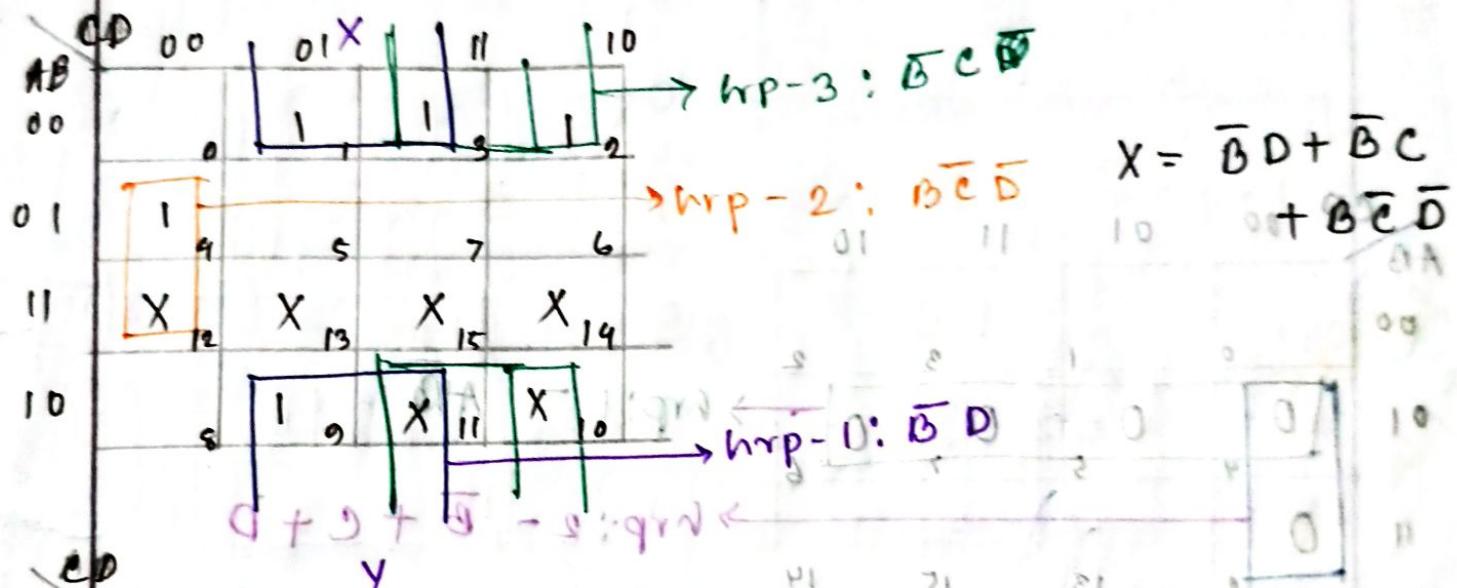
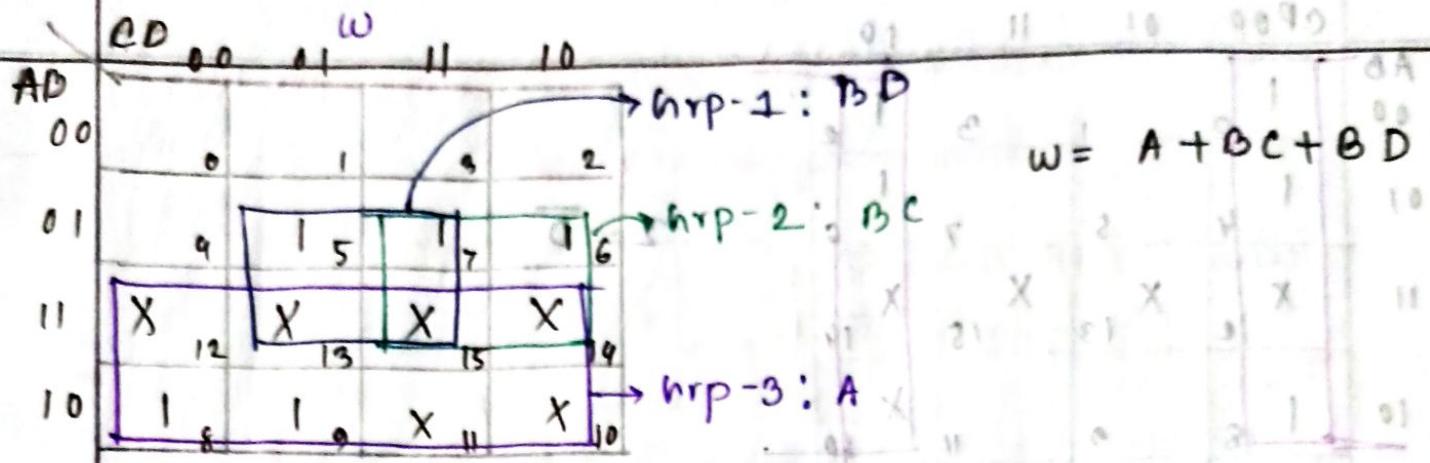
8 9 2]

## Decimal to BCD excess-3

	Input				Output				
	A	B	C	D	w	x	y	z	
0	0	0	0	0	0	0	1	1	$0+3=3$
1	0	0	0	1	0	1	0	0	$1+3=4$
2	0	0	1	0	0	1	0	1	$2+3=5$
3	0	0	1	1	0	1	1	0	6
4	0	1	0	0	0	1	1	1	7
5	0	1	0	1	1	0	0	0	8
6	0	1	1	0	1	0	0	1	9
7	0	1	1	1	1	0	$6A = 01-9m$	10	
8	1	0	0	0	1	0	$13A = 10-9m$	11	
✓9	1	0	0	1	1	1	0	$9A + 6A = 7$	12
10	1	0	1	0	X	X	X	X	
11	1	0	1	1	X	X	X	X	
12	1	1	0	0	X	X	X	X	
13	1	1	0	1	X	X	X	X	
14	1	1	1	0	X	X	X	X	
15	1	1	1	1	X	X	X	X	

$13A = 8-9m$

$13A + 9A + 13A = 7$



(2)

	CD 00	01	11	10
AB	1 0	1 4	3 5	2 7
00	X 1e	X 13	X 15	X 14
01	X 18	X 9	X 11	X 10
11				
10				

$$2 = \overline{D}$$

X	X	X	X
X	X	X	X
X	X	X	X
X	X	X	X

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

	CD	00	01	11	10
AB	00	0	1	3	2
00	0	0	0	0	0
01	0	0	0	0	0
11	0	0	0	0	0
10	0	0	0	0	0

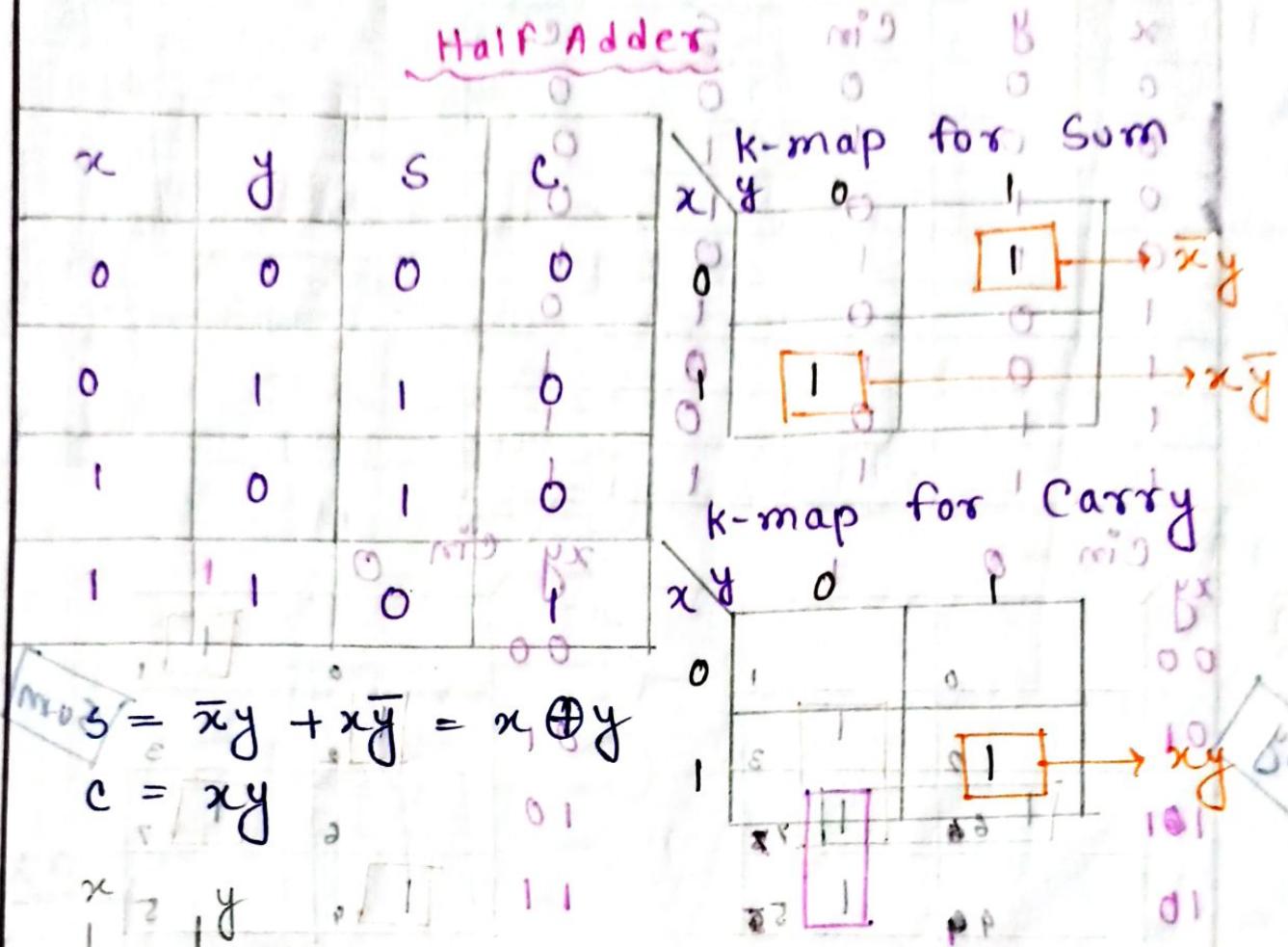
$$\text{hrp: } 1 = \overline{A}\overline{B}$$

$$\text{hrp: } 2 = \overline{B} + C + D$$

$$\text{hrp-3: } \overline{A} + \overline{B} + \overline{C} + \overline{D}$$

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

## Combinational Circuit:



A hand-drawn logic circuit diagram illustrating the implementation of a three-variable majority function ( $m_3$ ) using basic logic gates. The inputs are labeled  $A$ ,  $B$ , and  $C$ . The output  $s$  is determined by the following logic expression:

$$m_3 = \overline{A} \overline{B} C + \overline{A} B \overline{C} + A \overline{B} \overline{C}$$

The circuit consists of the following components:

- Three NOT gates (labeled  $\overline{A}$ ,  $\overline{B}$ , and  $\overline{C}$ ) receiving inputs  $A$ ,  $B$ , and  $C$  respectively.
- Three AND gates (labeled  $\overline{A} \overline{B} C$ ,  $\overline{A} B \overline{C}$ , and  $A \overline{B} \overline{C}$ ) receiving inputs from the NOT gates and input  $C$ .
- One OR gate (labeled  $s$ ) receiving inputs from the three AND gates.
- One NOT gate (labeled  $\overline{C}$ ) receiving input  $C$  and producing output  $c$ .

## Full Adder

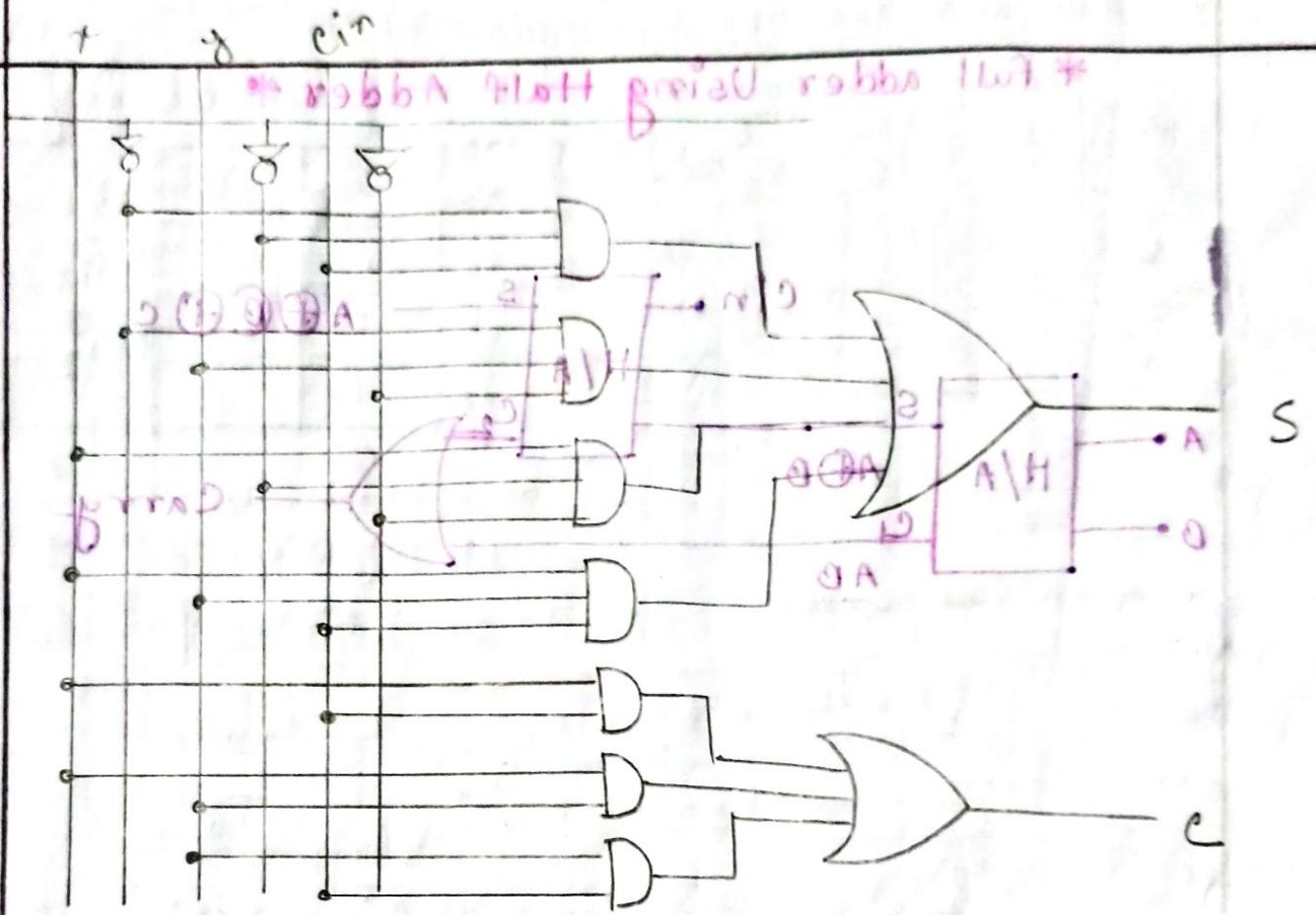
$x$	$y$	$c_{in}$	$S$	$c_{out}$
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
1	0	1	1	0
1	1	1	0	1
1	1	1	1	1

$x$	$y$	$c_{in}$	$c_{out}$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	1

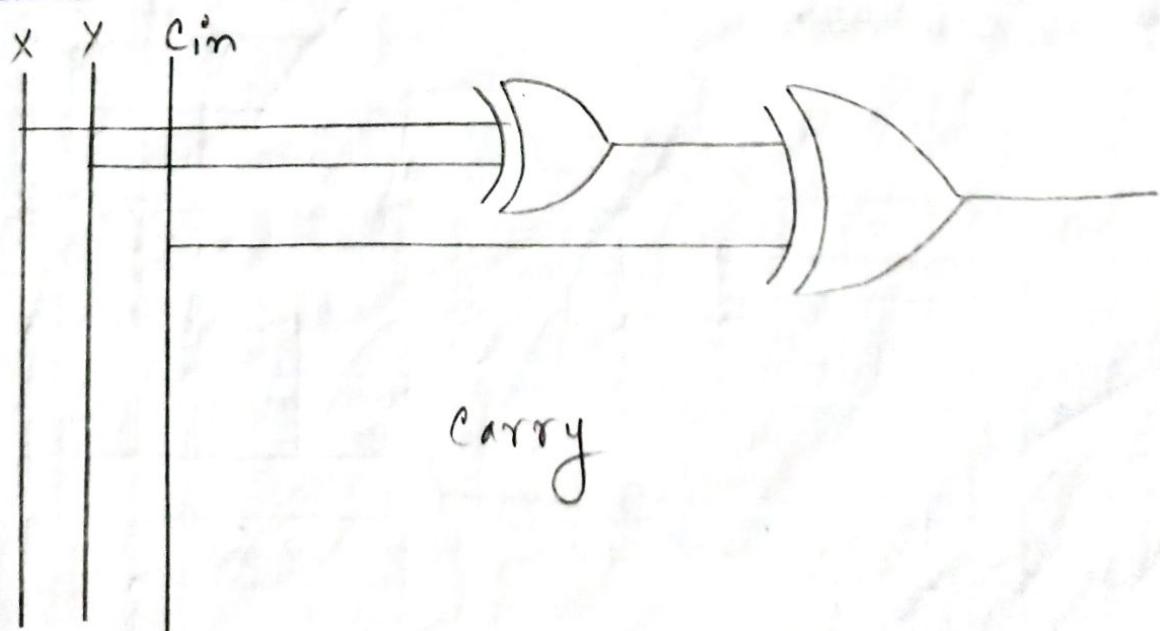
$$c = xc_{in} + xy + yc_{in}$$

$x$	$y$	$c_{in}$	$c_{out}$	$Sum$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	0	1	0

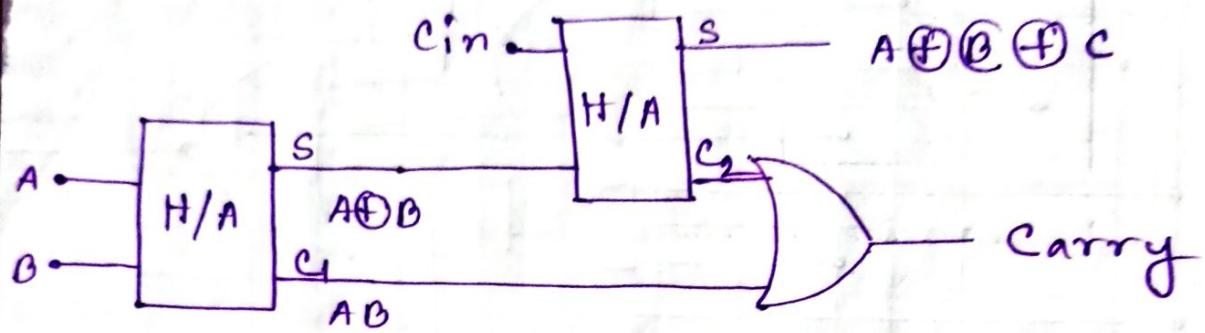
$$\begin{aligned}
 S &= \bar{x}\bar{y} c_{in} + \bar{x}y\bar{c}_{in} \\
 &\quad + x\bar{y} \bar{c}_{in} + xy c_{in} \\
 &= x \oplus y \oplus c_{in}
 \end{aligned}$$



XOR



## \* full adder Using Half Adder \*



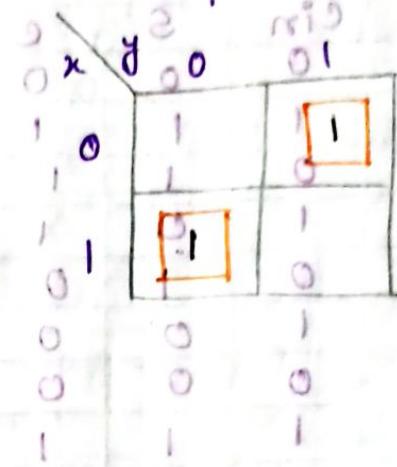
ROX

## Half Subtractor

K-map for sum:

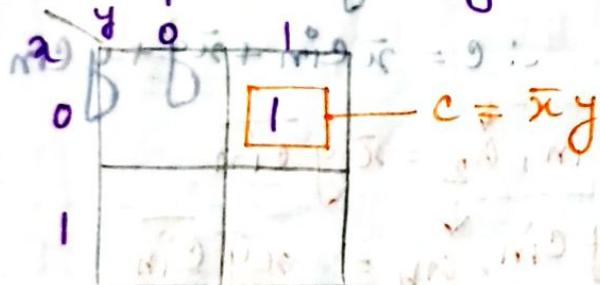
x	y	s	c
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map for sum:



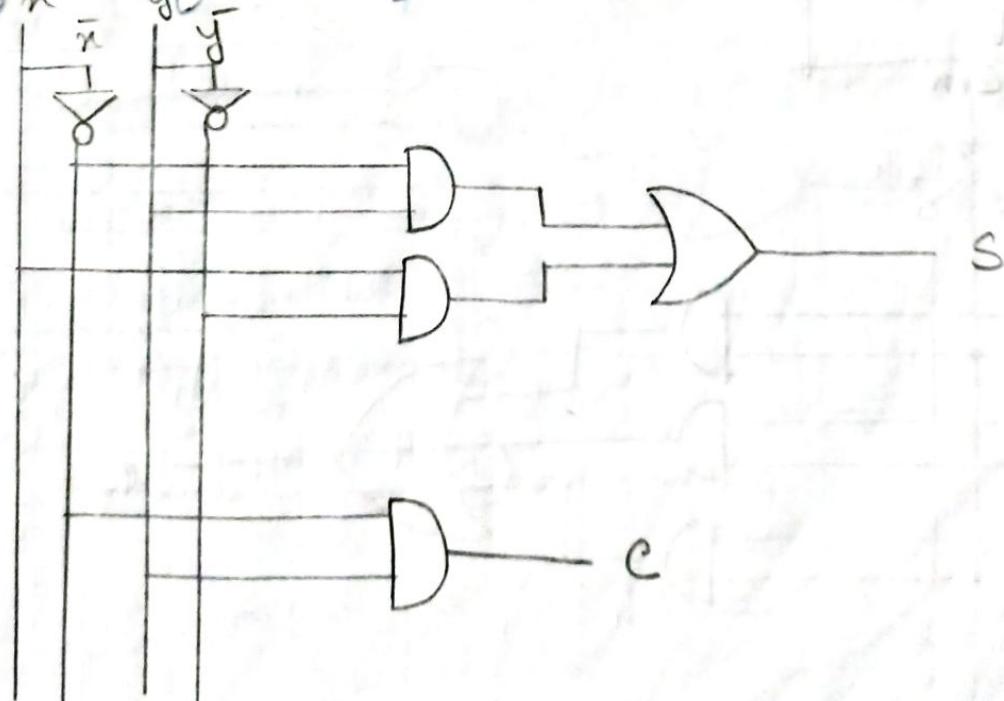
$$S = \bar{x}y + x\bar{y}$$

K-map for carry



K-map for sum:

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



## Full Subtractor

x	y	R	c <sub>in</sub>	S	c
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	1	1	1

K-map for β Carry

x <sub>1</sub>	x <sub>0</sub>	y <sub>1</sub>	y <sub>0</sub>	c <sub>1</sub>	c <sub>0</sub>
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	1	0
1	1	1	1	0	1

$$c_1 = \bar{x}c_{in}, c_2 = \bar{xy}$$

$$c_3 = y c_{in}$$

$$\therefore c = \bar{x}c_{in} + \bar{xy} + y c_{in}$$

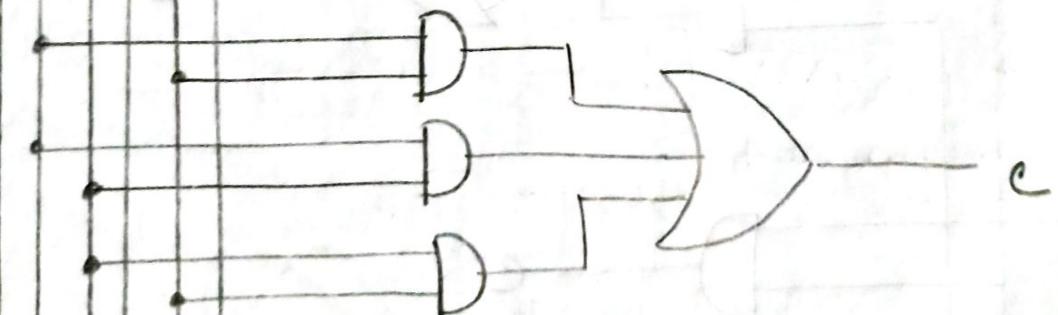
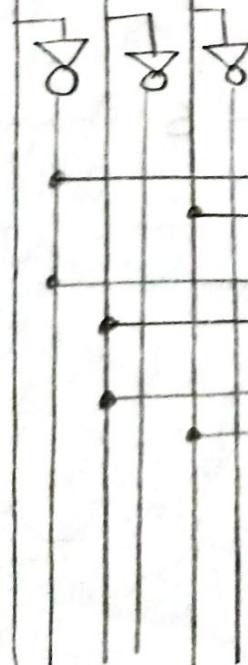
K-map for sum

x <sub>1</sub>	x <sub>0</sub>	y <sub>1</sub>	y <sub>0</sub>	c <sub>1</sub>	c <sub>0</sub>
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	1	0
1	1	1	1	0	1

$$a_1 = \bar{x}\bar{y}c_{in}, a_2 = \bar{x}y c_{in}$$

$$a_3 = xy c_{in}, a_4 = x\bar{y} c_{in}$$

$$S = \bar{x}\bar{y}c_{in} + \bar{x}y\bar{c}_{in} + xy\bar{c}_{in} + x\bar{y}c_{in}$$



rotate right shift left bid out

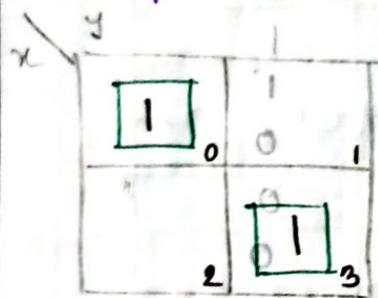
### One-bit Magnitude Comparator

(L) E (H) (L)

x	y	$x = y$	$x > y$	$x \leq y$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	0	0	0

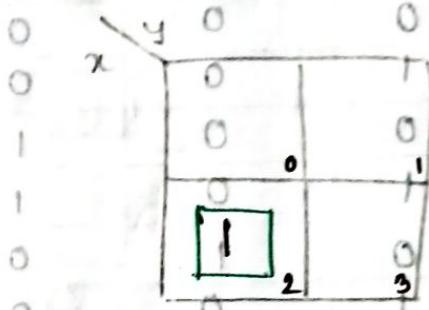
B	$\oplus B$	$\oplus x$	$\oplus y$
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	0
0	0	1	0
1	1	1	0
0	0	0	1
1	1	0	1

K-map  $\Rightarrow E$



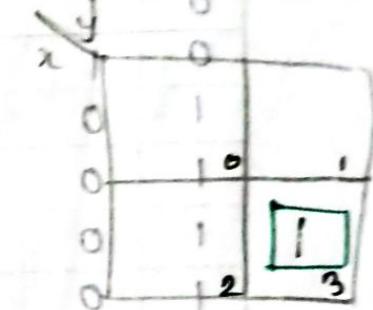
$$E = \bar{x}\bar{y} + xy$$

K-map  $\Rightarrow h$

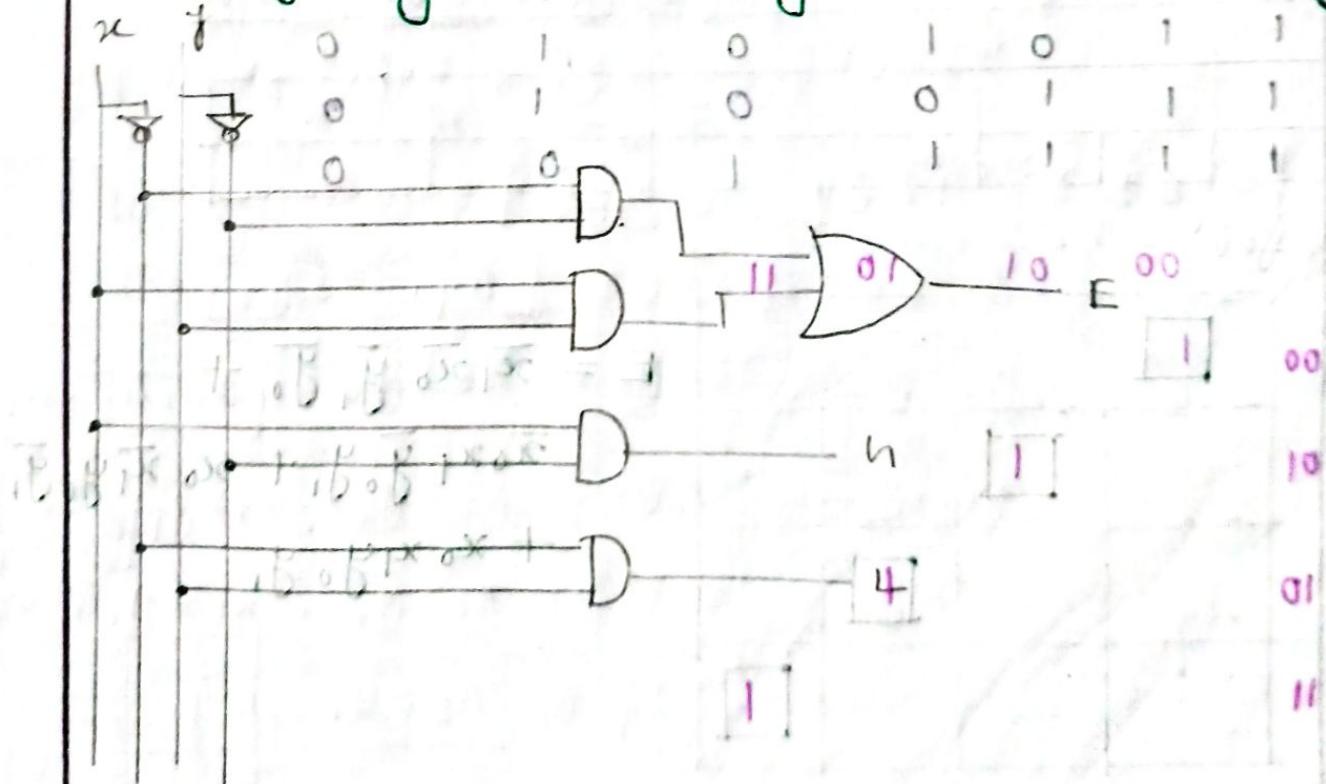


$$h = x\bar{y}$$

K-map  $\Rightarrow L$



$$L = \bar{x}y$$



two bit Magnitude Comparator

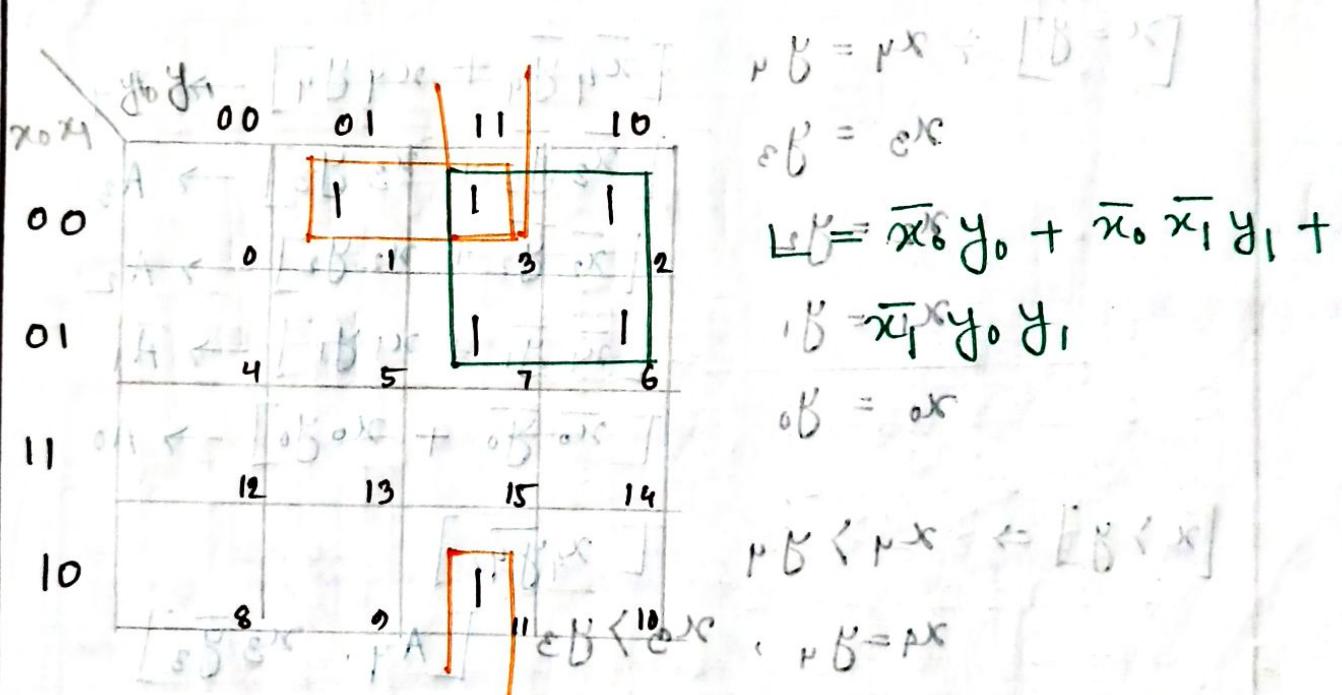
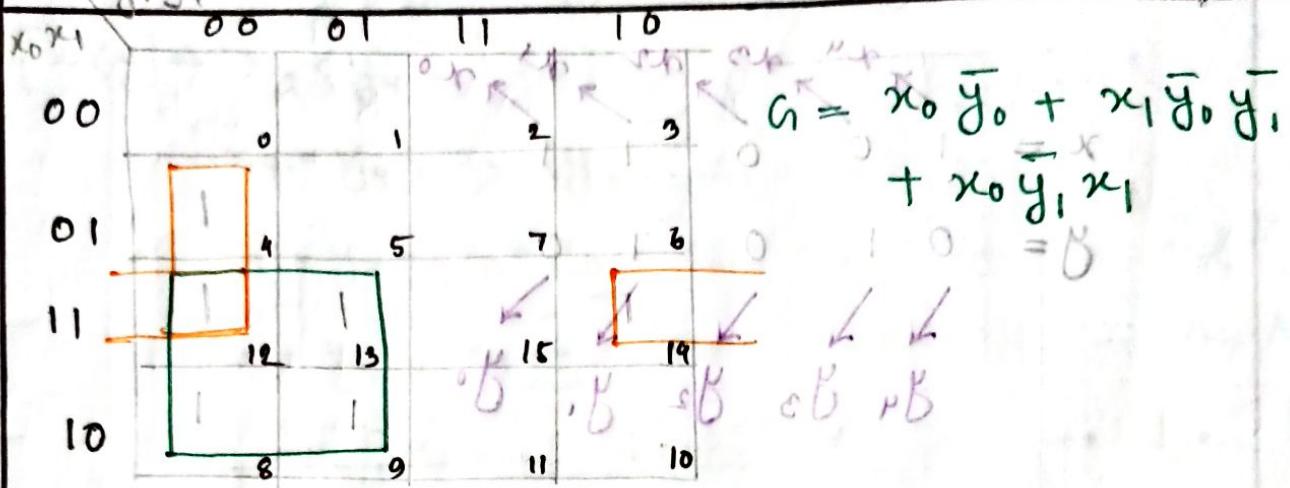
take negated strings in 1's complement

$x_0$	$x_1$	$y_0$	$y_1$	$B = x = y$	$B = x > y$	$B = x < y$	$B$
0	0	0	0	1	0	1	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	1
0	1	1	0	0	0	1	1
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	1	0	0	0
1	0	1	1	0	0	1	1
1	1	0	0	0	1	0	0
1	1	0	1	0	1	0	0
1	1	1	0	0	1	0	0
1	1	1	1	1	0	0	0

$y_0 \backslash x_1$	00	01	10	11
$x_0 \backslash y_1$	00	1		
00	1			
01		1		
10			1	
11				1

$$\begin{aligned}
 E = & \bar{x}_1 \bar{x}_0 \bar{y}_1 \bar{y}_0 + \\
 & \bar{x}_0 x_1 \bar{y}_0 \bar{y}_1 + x_0 \bar{x}_1 y_0 \bar{y}_1 \\
 & + x_0 x_1 y_0 y_1
 \end{aligned}$$

\* rotoregmas abutirgur tie sit



$\neg B < p\bar{x} \Leftrightarrow [B < p\bar{x}]$

$\neg B < c\bar{x} \Leftrightarrow [B = c\bar{x}]$

$\neg B < p\bar{x} \Leftrightarrow [B = p\bar{x}]$

**Must draw Circuit**

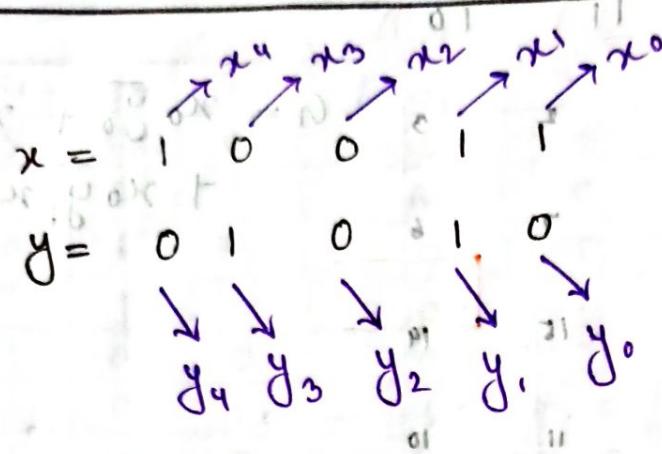
$\neg B < p\bar{x} \Leftrightarrow [B = p\bar{x}]$

$\neg B < c\bar{x} \Leftrightarrow [B = c\bar{x}]$

$\neg B < p\bar{x} \Leftrightarrow [B = p\bar{x}]$

$\neg B < p\bar{x} \Leftrightarrow [A : sA : \bar{s}A : p\bar{A}]$

## \* Five bit magnitude comparator \*



	00	01	10	11
00				
10				
11				

$$[x = y] \Rightarrow x_4 = y_4$$

$$[\bar{x}_4 \bar{y}_4 + x_4 y_4] \rightarrow A_4$$

$$x_3 = y_3$$

$$[\bar{x}_3 \bar{y}_3 + x_3 y_3] \rightarrow A_3$$

$$x_2 = y_2$$

$$[\bar{x}_2 \bar{y}_2 + x_2 y_2] \rightarrow A_2$$

$$x_1 = y_1$$

$$[\bar{x}_1 \bar{y}_1 + x_1 y_1] \rightarrow A_1$$

$$x_0 = y_0$$

$$[\bar{x}_0 \bar{y}_0 + x_0 y_0] \rightarrow A_0$$

$$[x > y] \Rightarrow x_4 > y_4$$

$$[x_4 \bar{y}_4]$$

$$x_4 = y_4, x_3 > y_3 \quad [A_4 \cdot x_3 \bar{y}_3]$$

$$x_4 = y_4, x_3 = y_3, x_2 > y_2$$

$$[A_4 \cdot A_3 \cdot x_2 \bar{y}_2]$$

$$x_4 = y_4, x_3 = y_3, x_2 = y_2, x_1 > y_1$$

$$[A_4 \cdot A_3 \cdot A_2 \cdot x_1 \bar{y}_1]$$

$$x_4 = y_4, x_3 = y_3, x_2 = y_2, x_1 = y_1, x_0 > y_0$$

$$[A_4 \cdot A_3 \cdot A_2 \cdot A_1 \cdot x_0 \bar{y}_0]$$

$$[x < y] \Rightarrow x_4 < y_4, [\bar{x}_4 y_4]$$

$$x_4 = y_4, x_3 < y_3 [A_4 \cdot \bar{x}_3 y_3]$$

$$x_4 = y_4, x_3 = y_3, x_2 < y_2 [A_4 \cdot A_3 \cdot \bar{x}_2 y_2]$$

$$x_4 = y_4, x_3 = y_3, x_2 = y_2, x_1 < y_1 [A_4 \cdot A_3 \cdot A_2 \cdot \bar{x}_1 y_1]$$

$$x_4 = y_4, x_3 = y_3, x_2 = y_2, x_1 = y_1, x_0 < y_0$$

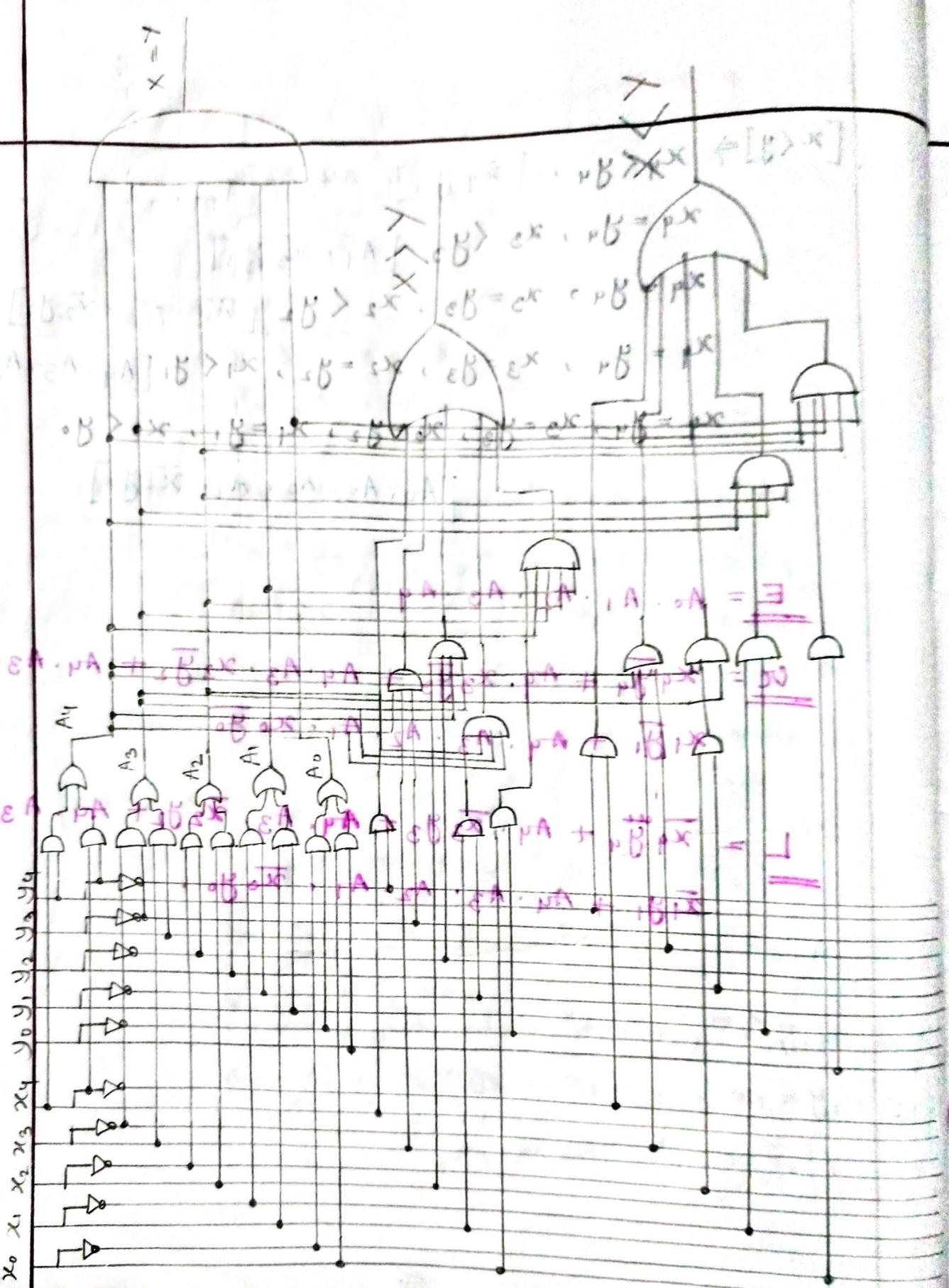
$$[A_4 \cdot A_3 \cdot A_2 \cdot A_1 \cdot \bar{x}_0 y_0]$$

$$\underline{E} = A_0 \cdot A_1 \cdot A_2 \cdot A_3 \cdot A_4$$

$$\underline{\underline{C}} = x_4 \bar{y}_4 + A_4 \cdot x_3 \bar{y}_3 + A_4 \cdot A_3 \cdot x_2 \bar{y}_2 + A_4 \cdot A_3 \cdot A_2 \\ x_1 \bar{y}_1 + A_4 \cdot A_3 \cdot A_2 \cdot A_1 \cdot x_0 \bar{y}_0$$

$$\underline{\underline{L}} = \bar{x}_4 \bar{y}_4 + A_4 \cdot \bar{x}_3 y_3 + A_4 \cdot A_3 \cdot \bar{x}_2 y_2 + A_4 \cdot A_3 \cdot A_2$$

$$\bar{x}_1 y_1 + A_4 \cdot A_3 \cdot A_2 \cdot A_1 \cdot \bar{x}_0 y_0.$$



~~Binary Rotating Subtraction~~ ~~to~~ ~~id~~  
 Carry Look Ahead Adder

$$S = A_0 \oplus B_0 \oplus C_0$$

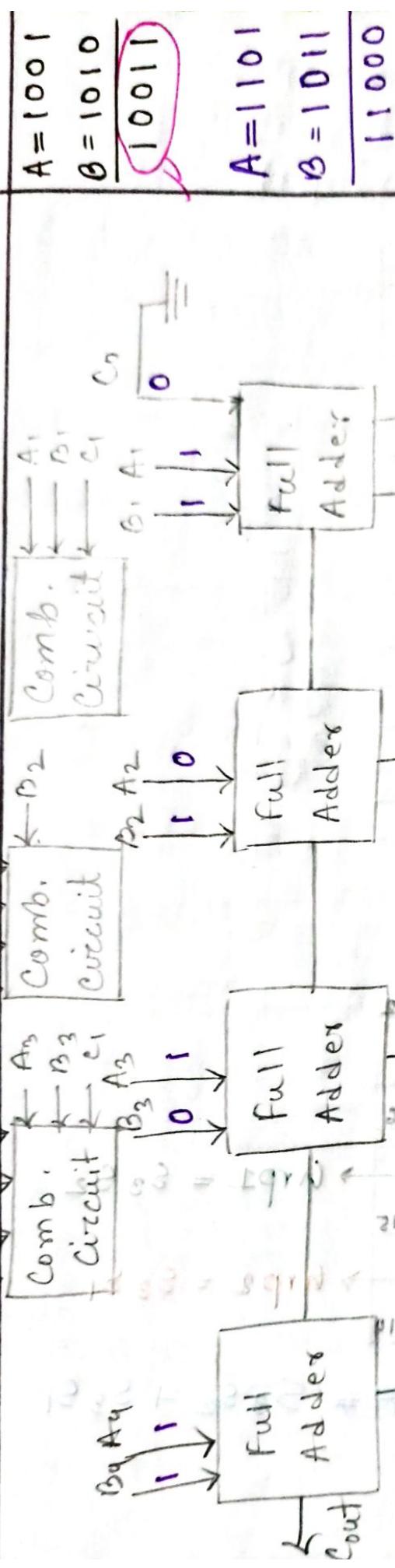
$$\begin{aligned} C_1 &= \frac{A_0 B_0}{G_0} + B_0 C_0 + A_0 C_0 \\ &= G_0 + C_0 \left( \frac{A_0 + B_0}{P_0} \right) \\ &= G_0 + [C_0 P_0] \end{aligned}$$

$$\begin{aligned} C_2 &= G_1 + C_1 P_1 \\ &= G_1 + (G_0 + C_0 P_0) P_1 \\ &= G_1 + G_0 P_1 + P_0 P_1 [C_0] \end{aligned}$$

$$\begin{aligned} C_3 &= G_2 + C_2 P_2 \\ &= G_2 + (G_1 + G_0 P_1 + P_0 P_1 C_0) P_2 \\ &= G_2 + G_1 P_2 + G_0 P_1 P_2 + P_0 P_1 P_2 [C_0] \end{aligned}$$

$$\begin{aligned} C_4 &= G_3 + C_3 P_3 \\ &= G_3 + (G_2 + G_1 P_2 + G_0 P_1 P_2 + P_0 P_1 P_2 C_0) P_3 \\ &= G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + P_0 P_1 P_2 P_3 [C] \end{aligned}$$

$$A_1 \ B_1 \ A_2 \ B_2$$



result: 11000

result

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

11000

## BCD Adder

0000	1101	0110	1011	0011 = A
0001	1110	0101	1001	0111 = B

$S_3$	$S_2$	$S_1$	$S_0$	Error
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1



$$hrp1 = S_3 S_2 \quad \text{(highlighted in green)}$$

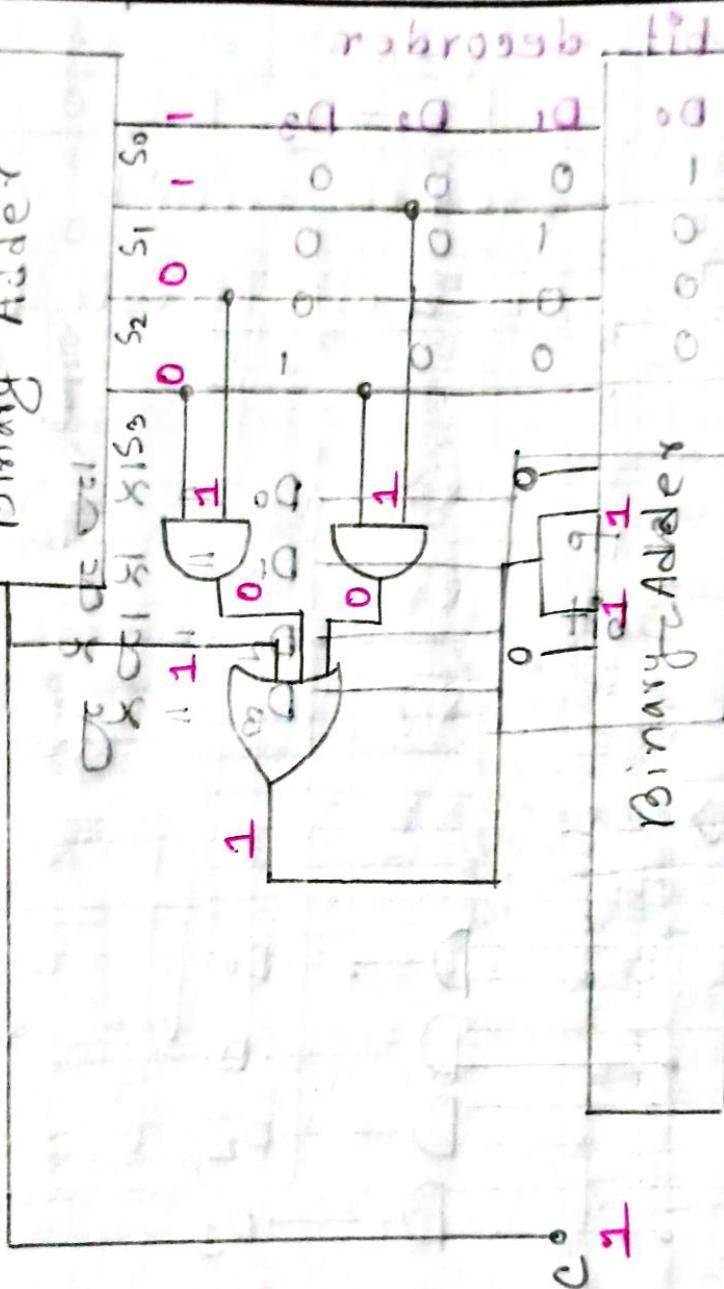
$$hrp2 = S_3 S_1 \quad \text{(highlighted in orange)}$$

$$\Theta - F = S_3 S_2 + S_3 S_1 \quad \text{(highlighted in pink)}$$

$$\begin{array}{r}
 A_3 A_2 A_1 A_0 \\
 \text{---} \\
 B_3 B_2 B_1 B_0 \\
 \text{---} \\
 1 \ 0 \ 0 \ 1 \\
 - \ 0 \ 0 \ 1 \\
 \hline
 1 \ 0 \ 0 \ 1
 \end{array}$$

Binary Adder

T



Binary Adder

T

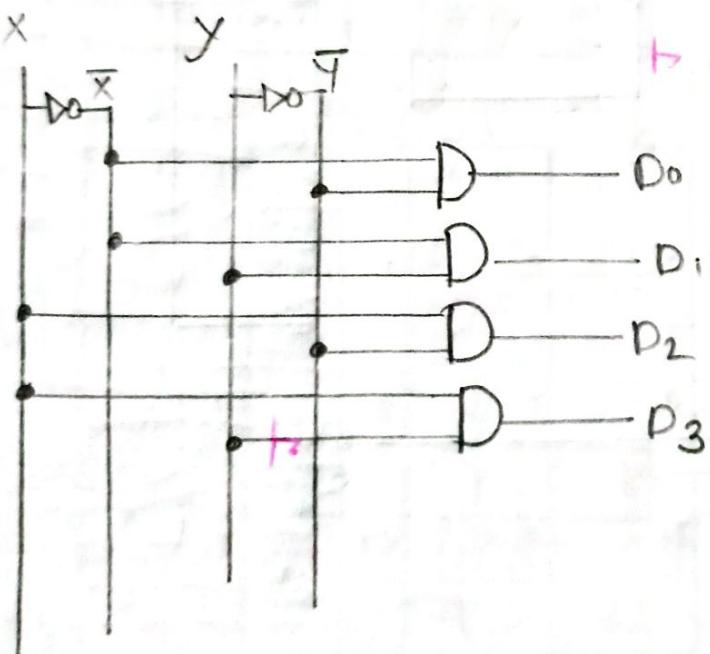
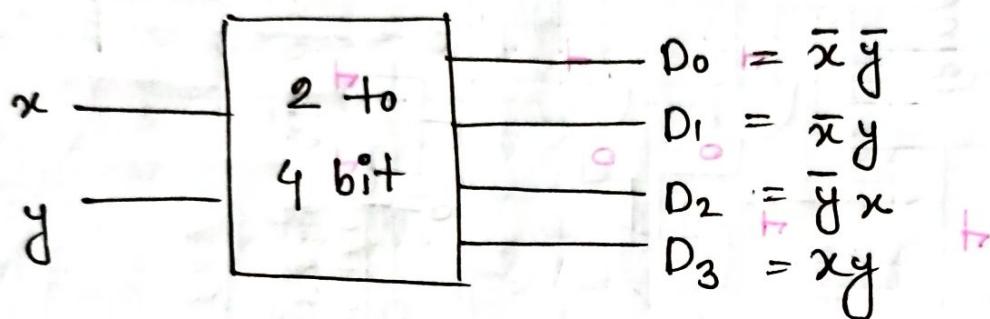
$$\begin{array}{r}
 S_0 \quad S_1 \quad S_2 \quad S_3 \\
 \text{---} \\
 T_0 \quad T_1 \quad T_2 \quad T_3 \\
 \text{---} \\
 0 \ 0 \ 0 \ 0 \\
 - \ 0 \ 0 \ 1 \\
 \hline
 1 \ 0 \ 0 \ 0
 \end{array}$$

$$\begin{array}{r}
 A = 1 \ 0 \ 0 \ 1 \\
 B = 1 \ 0 \ 0 \ 1 \\
 \hline
 1 \ 1 \ 0 \ 0 \ 0
 \end{array}$$

Decoder ( $2^n$ )

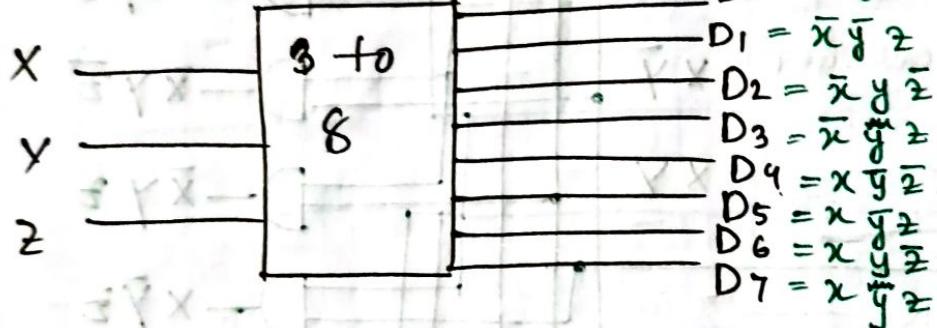
2 to 4 bit decoder

x	y	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



3 to 8 bit decoder

X	Y	Z	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



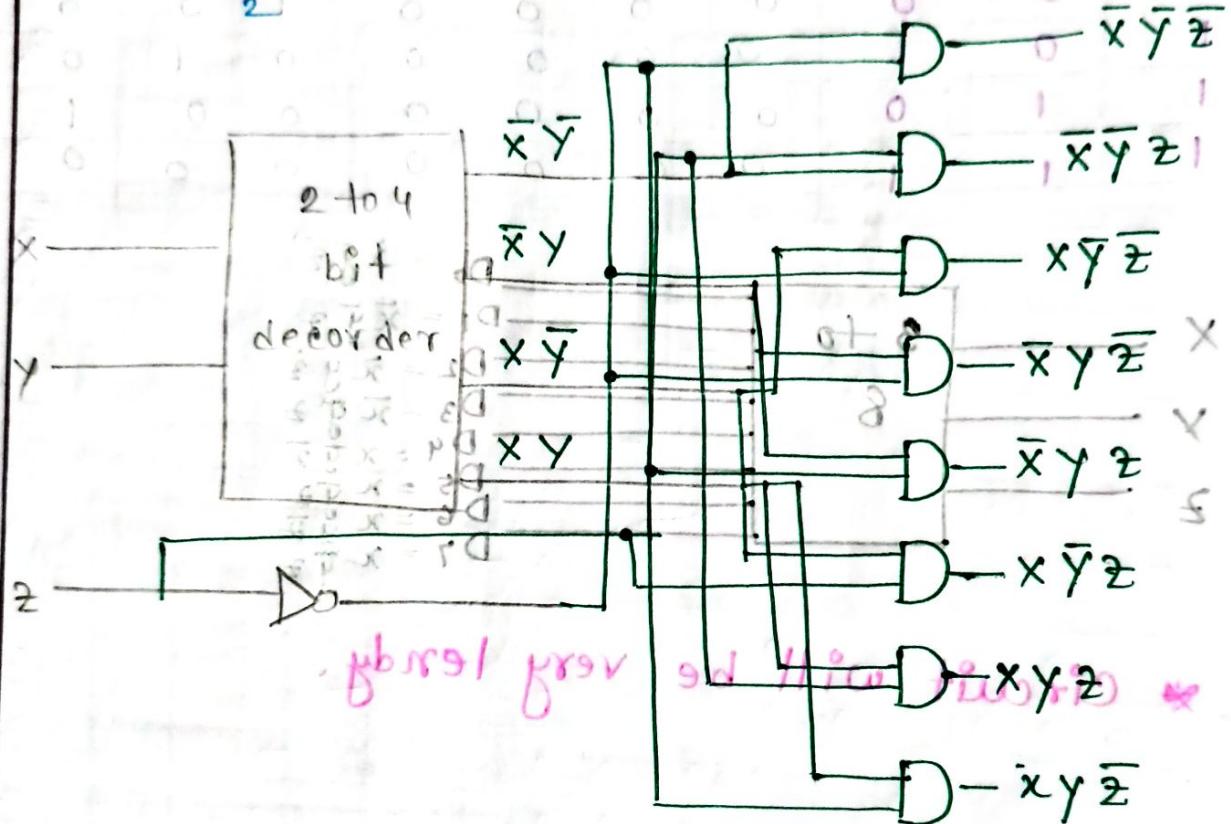
\* Circuit will be very lousy.

## different type of Decoder Circuit

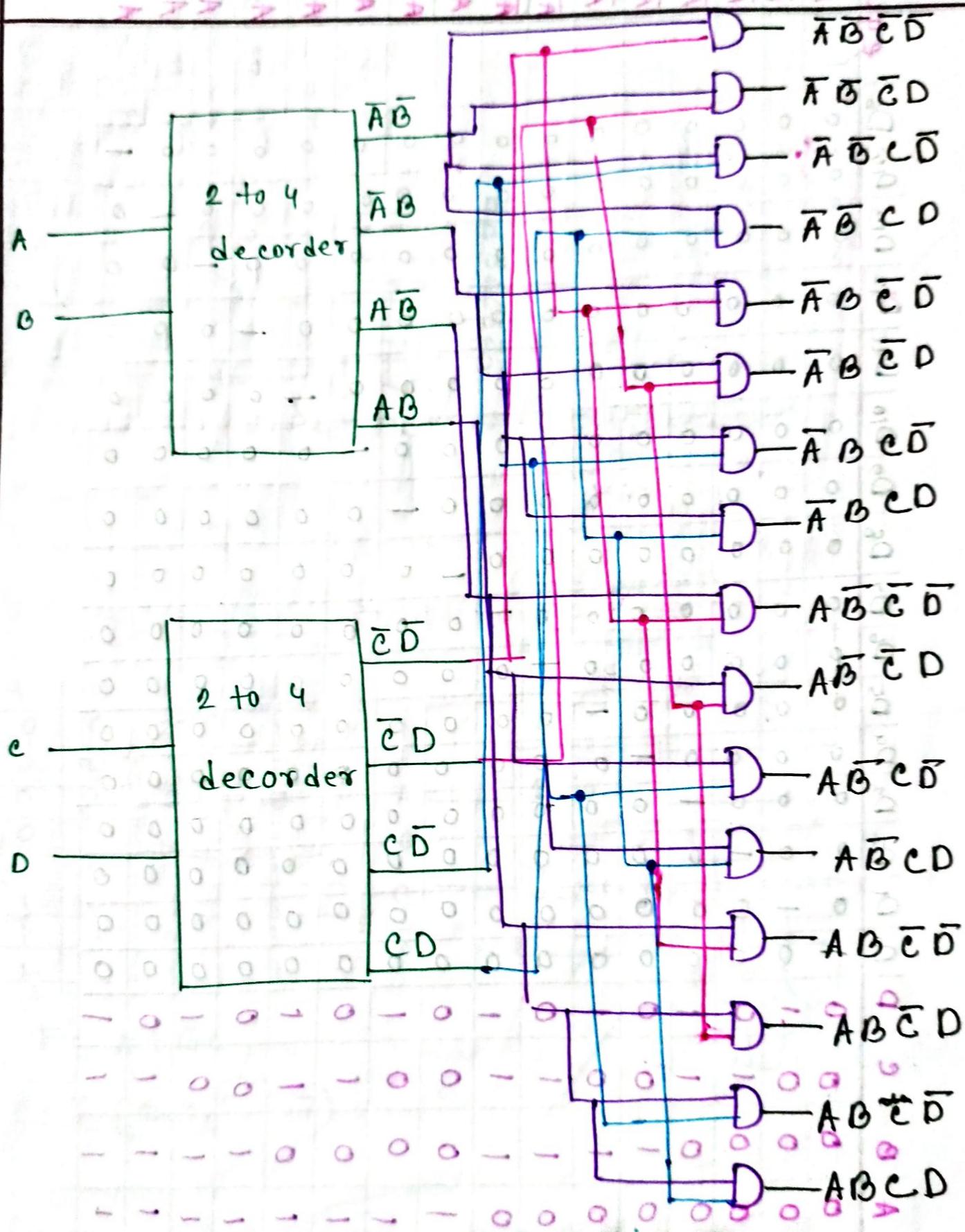
Formula:  $\frac{n+1}{2}, \frac{n-1}{2}$

$\rightarrow \frac{3+1}{2} = 2$  to 4 bit circuit

$\rightarrow \frac{3-1}{2} = 1$  to 2 bit circuit [NOT gate]



16 Decoder Circuit

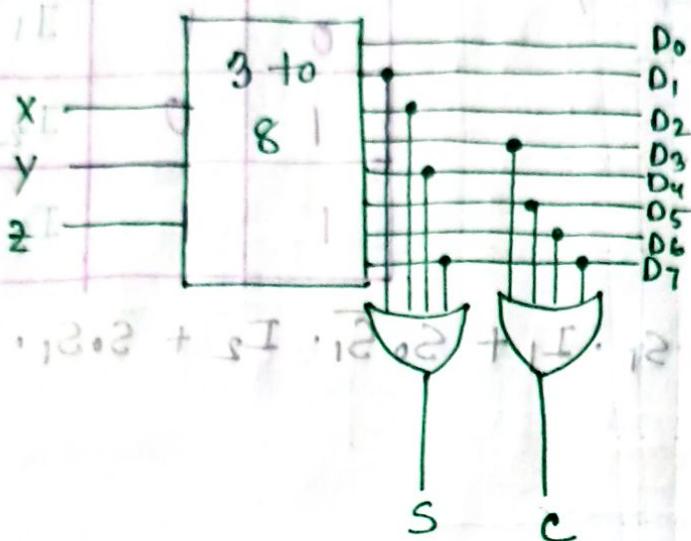




## Designing a decoder with full adder

$$S = \sum_{1,4} (1, 3, 4, 7)$$

$$C = \sum_{0,2} (3, 5, 6, 7)$$



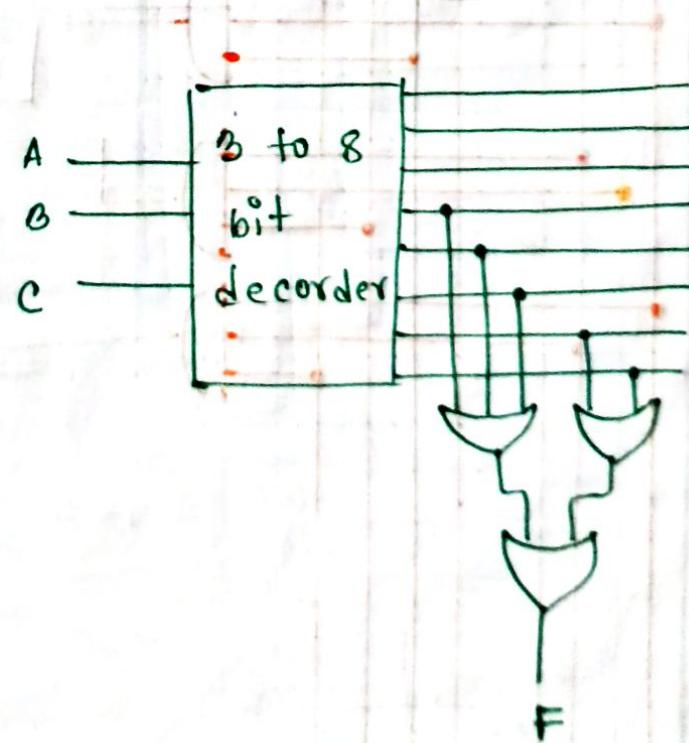
$$S = I_1 \cdot 1 \cdot 3 + S_I \cdot 1 \cdot 2 + I_L \cdot 1 \cdot 2 \cdot 3 + O_I \cdot 1 \cdot 2 \cdot 3 = 3$$

## Designing a circuit with function

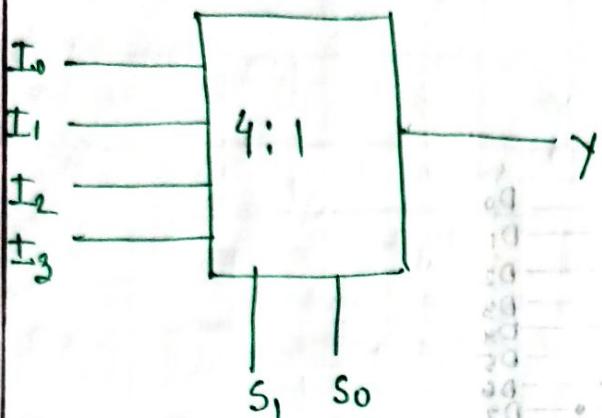
A	B	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$F = A + Bc /$$

$$F = \sum (3, 4, 5, 6, 7)$$

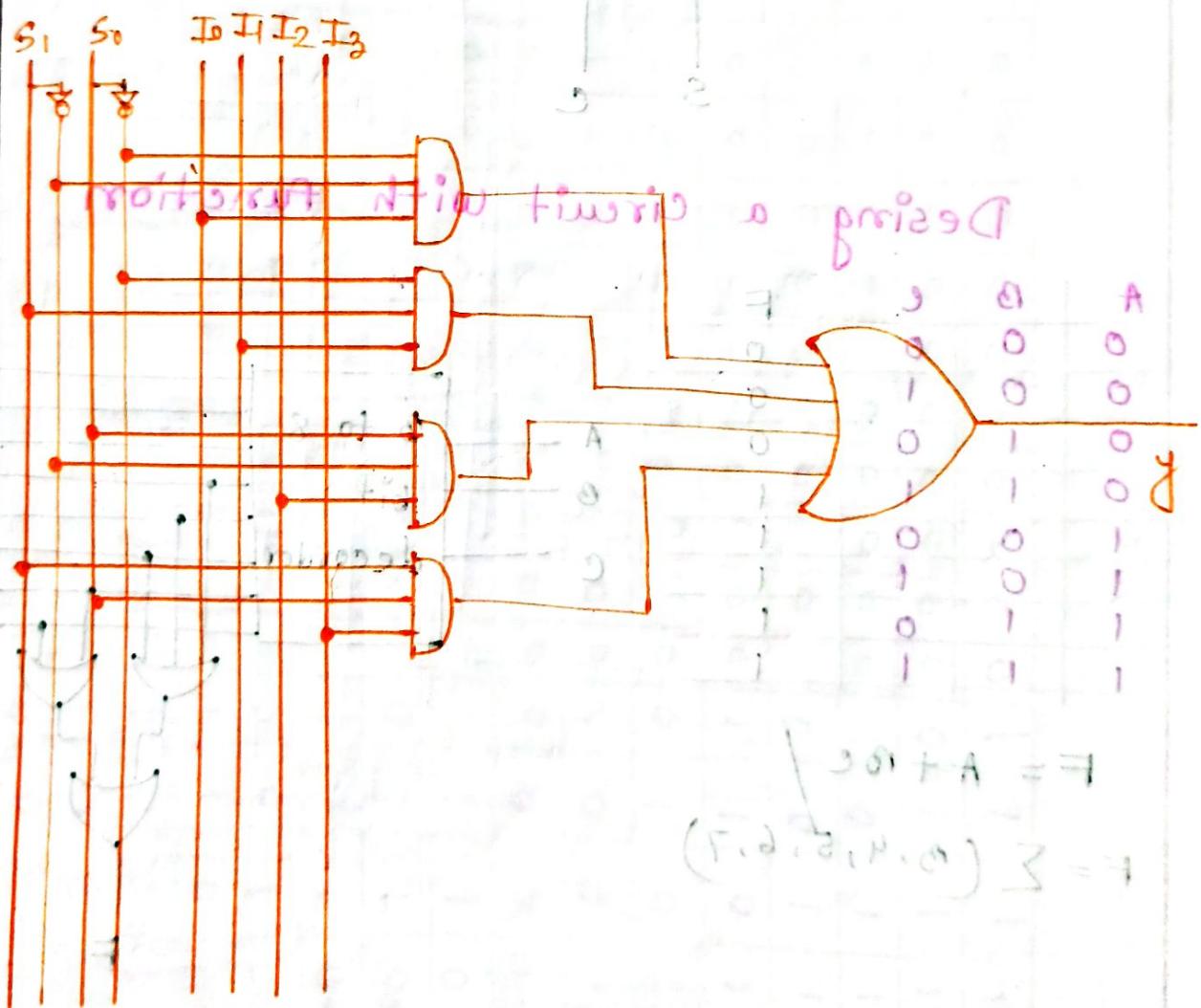


re�ka llnk Nlini rob rabb → projekti  
Multiplexer



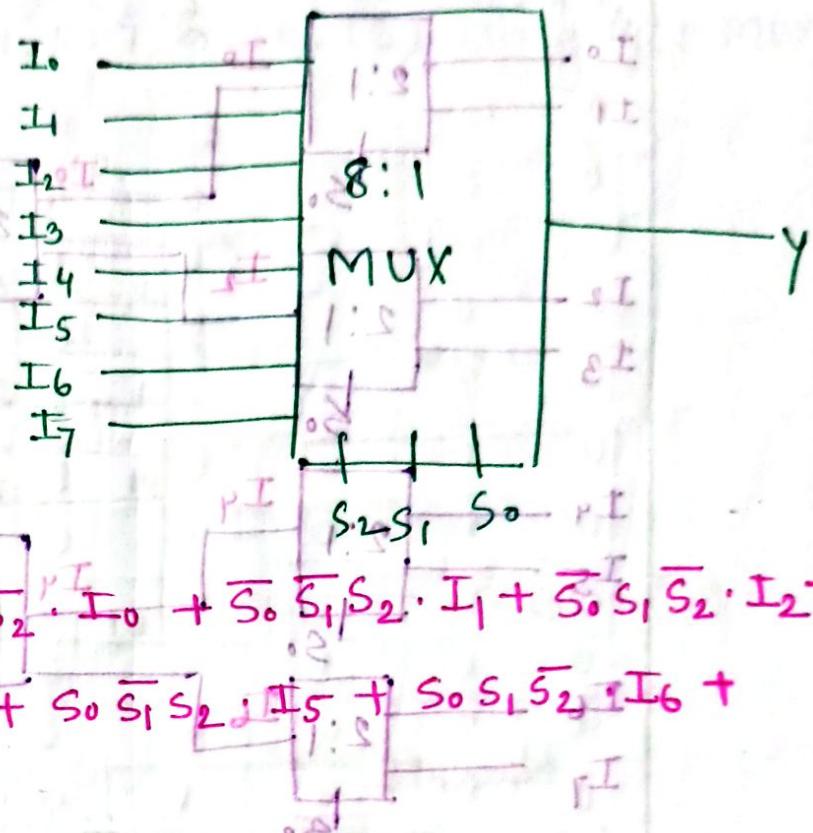
S0	S1	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

$$Y = \overline{S_0} \overline{S_1} \cdot I_0 + \overline{S_0} S_1 \cdot I_1 + S_0 \overline{S_1} \cdot I_2 + S_0 S_1 \cdot I_3$$



(XUM 1:2 prioru XUM 1:8) resultum

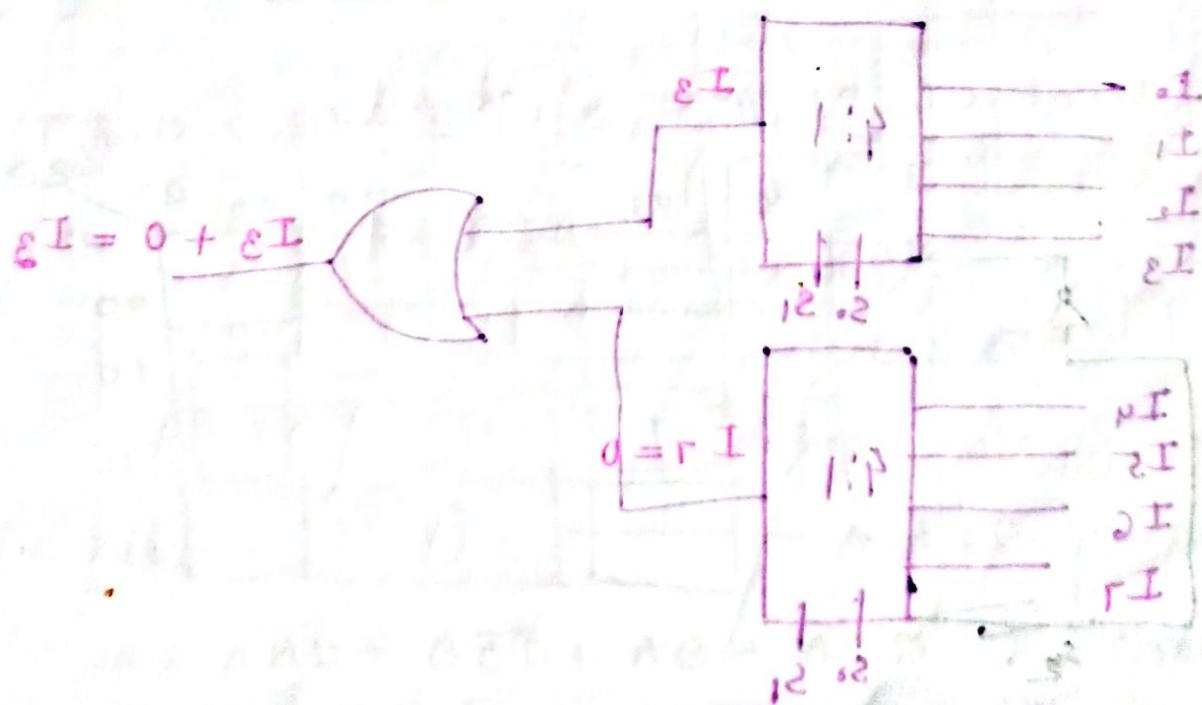
$S_0$	$S_1$	$S_2$	$y$
0	0	0	$I_0$
0	0	1	$I_1$
0	1	0	$I_2$
0	1	1	$I_3$
1	0	0	$I_4$
1	0	1	$I_5$
1	1	0	$I_6$
1	1	1	$I_7$



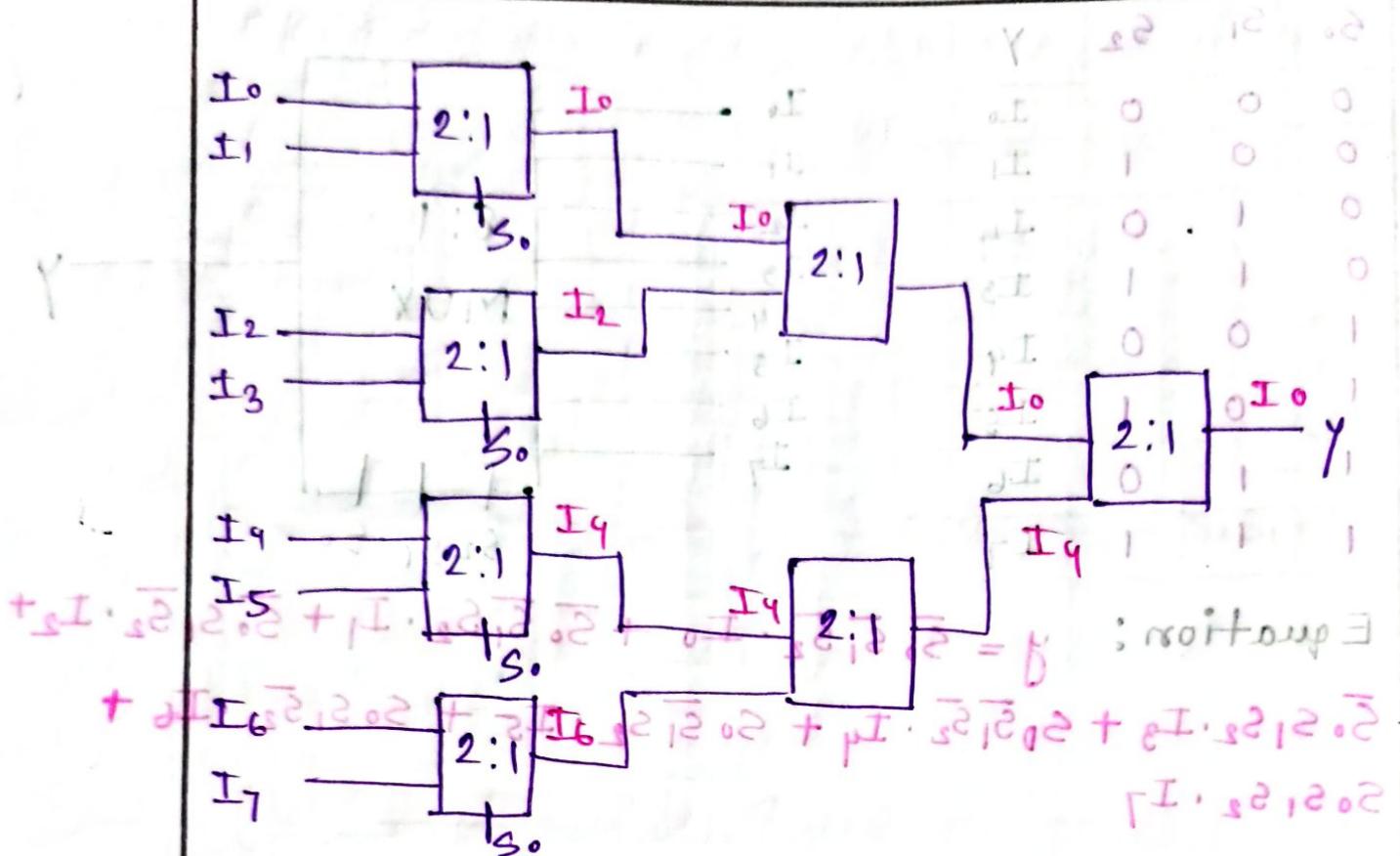
Equation:

$$y = \bar{S}_0 \bar{S}_1 \bar{S}_2 \cdot I_0 + \bar{S}_0 \bar{S}_1 S_2 \cdot I_1 + \bar{S}_0 S_1 \bar{S}_2 \cdot I_2 + \bar{S}_0 S_1 S_2 \cdot I_3 + S_0 \bar{S}_1 \bar{S}_2 \cdot I_4 + S_0 \bar{S}_1 S_2 \cdot I_5 + S_0 S_1 \bar{S}_2 \cdot I_6 + S_0 S_1 S_2 \cdot I_7$$

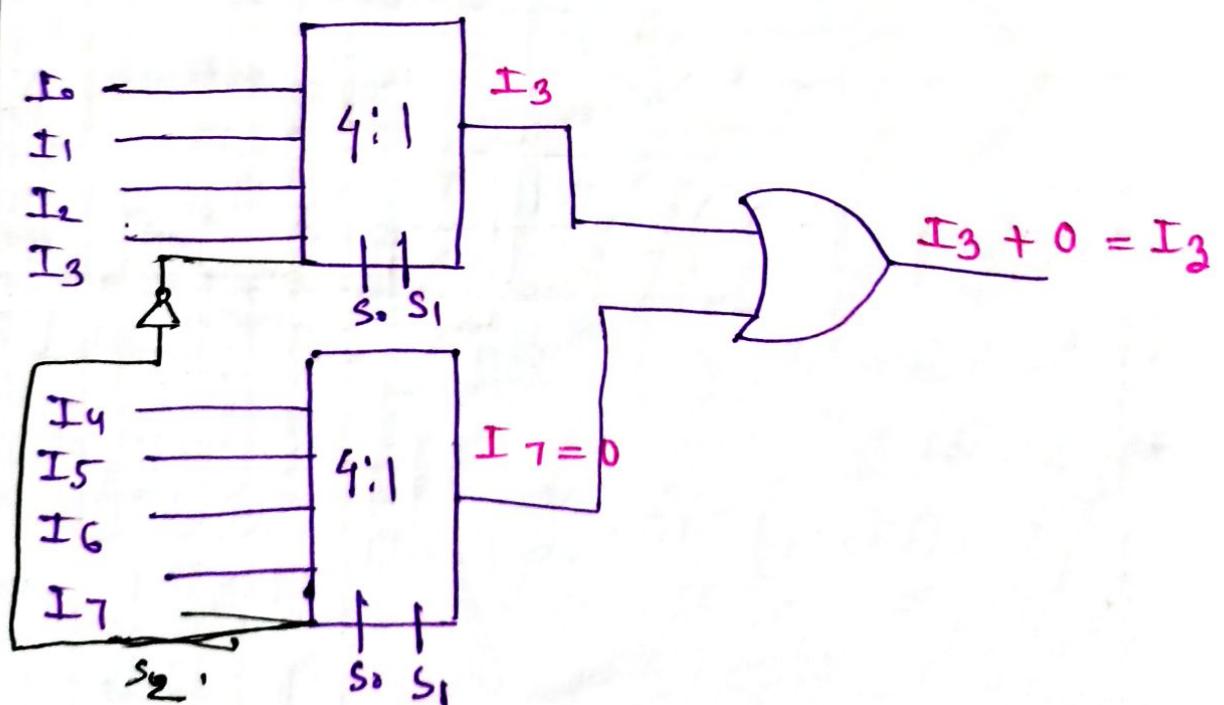
### XUM 1:8 CIRCUIT



## Multiplexer (8:1 MUX using 2:1 MUX)



## 8:1 MUX using 4:1 MUX



XOR gate realization

## Implementation Boolean Function

$$F(A, B, C, D) = \sum m(1, 4, 5, 7, 9, 12, 13) \text{ using } (4:1 \text{ mux})$$

S<sub>0</sub> S<sub>1</sub>

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

$$A \quad 0 \quad 1 \quad 0 \quad A$$

$$\bar{C}D \quad 0 \quad 0 \quad 0 \quad 0$$

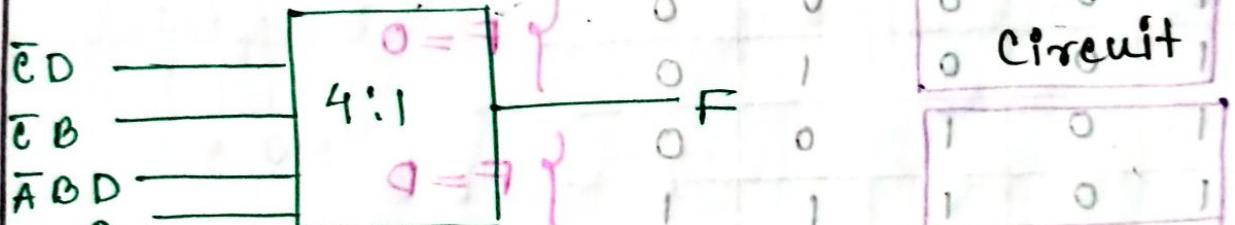
$$Grp2 = \bar{A}BD \quad 1 \quad 0 \quad 0 \quad 0$$

$$Grp3 = \bar{C}B \quad 0 \quad 1 \quad 0 \quad 0$$

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



Circuit



$$F(A, B, C, D) = \sum m(1, 3, 4, 11, 12, 13, 14, 15)$$

S<sub>0</sub> S<sub>1</sub>

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

$$\bar{A}BD \quad 1 \quad 1 \quad 1$$

$$B\bar{C}\bar{D} \quad 0 \quad 1 \quad 1$$

$$Grp1 : AB \quad 1 \quad 0 \quad 1$$

$$ACD \quad 1 \quad 1 \quad 0$$

$$F = \bar{A}BD + B\bar{C}\bar{D} + AB + ACD \Rightarrow \text{circuit}$$

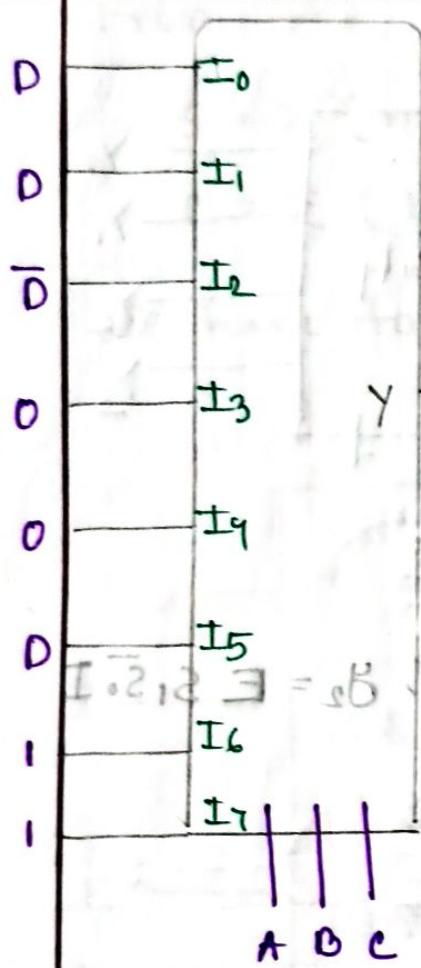
## Implementation Using MUX

$$F(A, B, C, D) = \sum (1, 3, 4, 11, 12, 13, 14, 15)$$

(XOR 1:1) prev (81, 81, 0, 1, 2, 14) m3 = (4, 5, 0, 1, A), 13, 02

$$\text{firms} \leftarrow \text{firms} + \bar{A}G + \bar{A}D + \bar{A}\bar{G} + \bar{A}\bar{D}$$

## DE-Multibitex



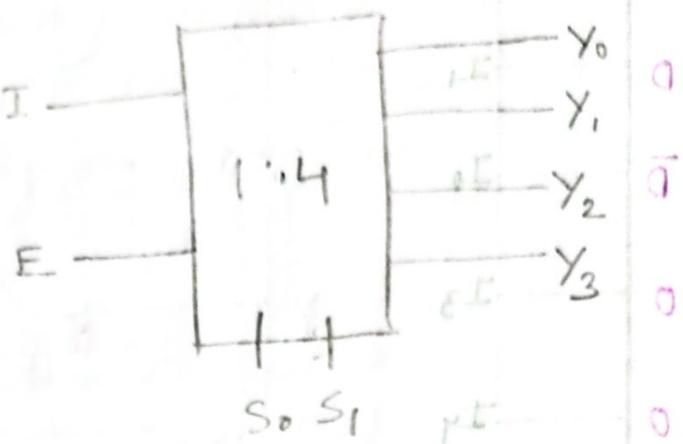
	Y	$\bar{Y}$	$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	X	X	0	2	2	3
	0	0	0	0	0	0	0	0	0	0	X	X	0	2	2	3
	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	1	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1

$$I_{0,2,2} \bar{E} = eB \quad I_{0,2,2} E = 1B \quad I_{0,2,2} \bar{E} = 0B$$

$$I_{0,2,2} E = eB$$

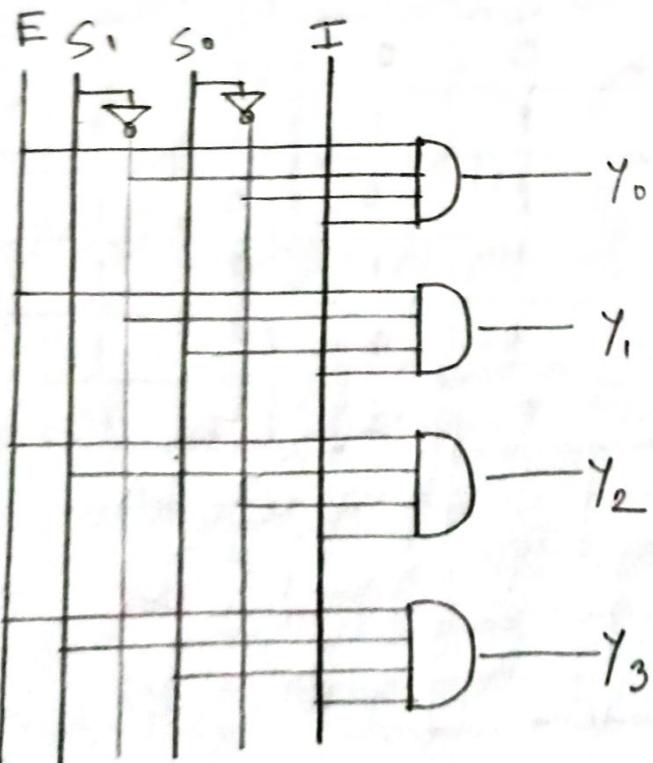
## DE-multiplexer

E	S <sub>1</sub>	S <sub>0</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
0	X	X	0	0	0	0
1	0	0	I	0	0	0
1	0	1	0	I	0	0
1	1	0	0	0	I	0
1	1	1	0	0	0	I



$$y_0 = E \bar{S}_1 \bar{S}_0 I, \quad y_1 = E \bar{S}_1 S_0 I, \quad y_2 = E S_1 \bar{S}_0 I$$

$$y_3 = E S_1 S_0 I$$



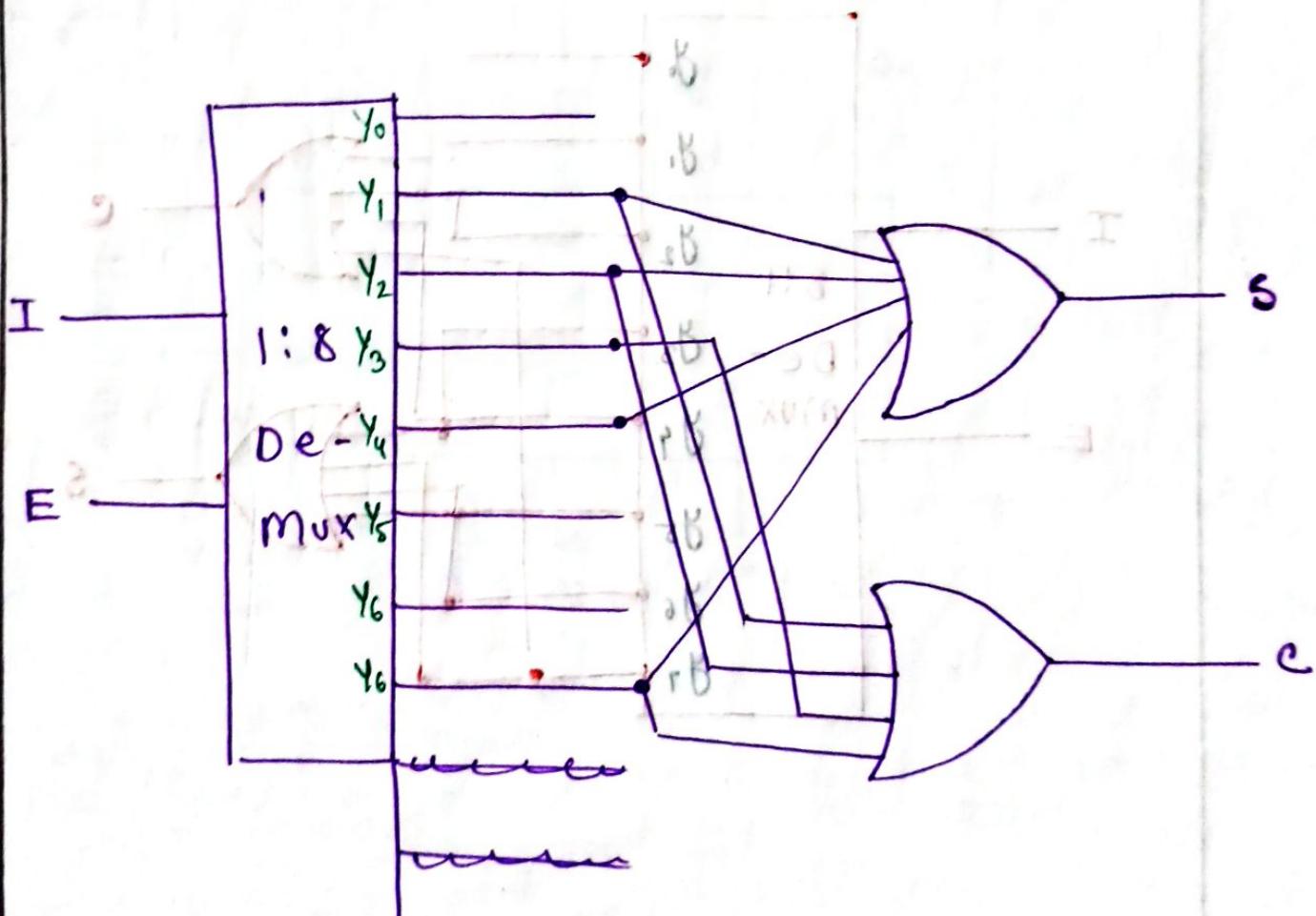
## Full subtractor Using de-multiplexer

From the full subtractor

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

$$C = \Sigma (1, 2, 3, 7)$$

we have to use (1:8 De-multiplexer)



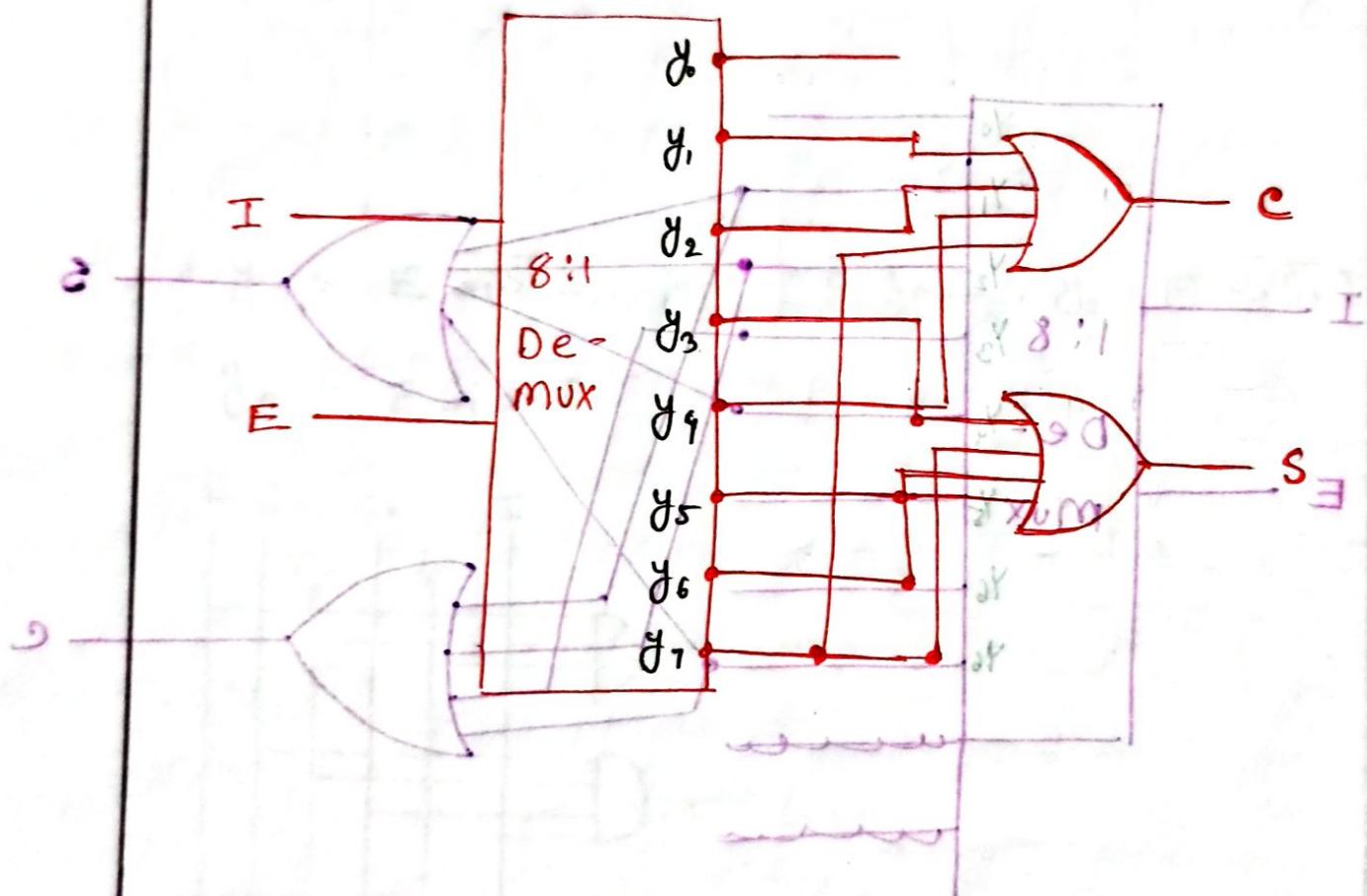
## Full Adder Using demultiplexer

From the full Adder

$$S = \sum (3, 5, 6, 7) \quad (F, P, S, U) Z = 2$$

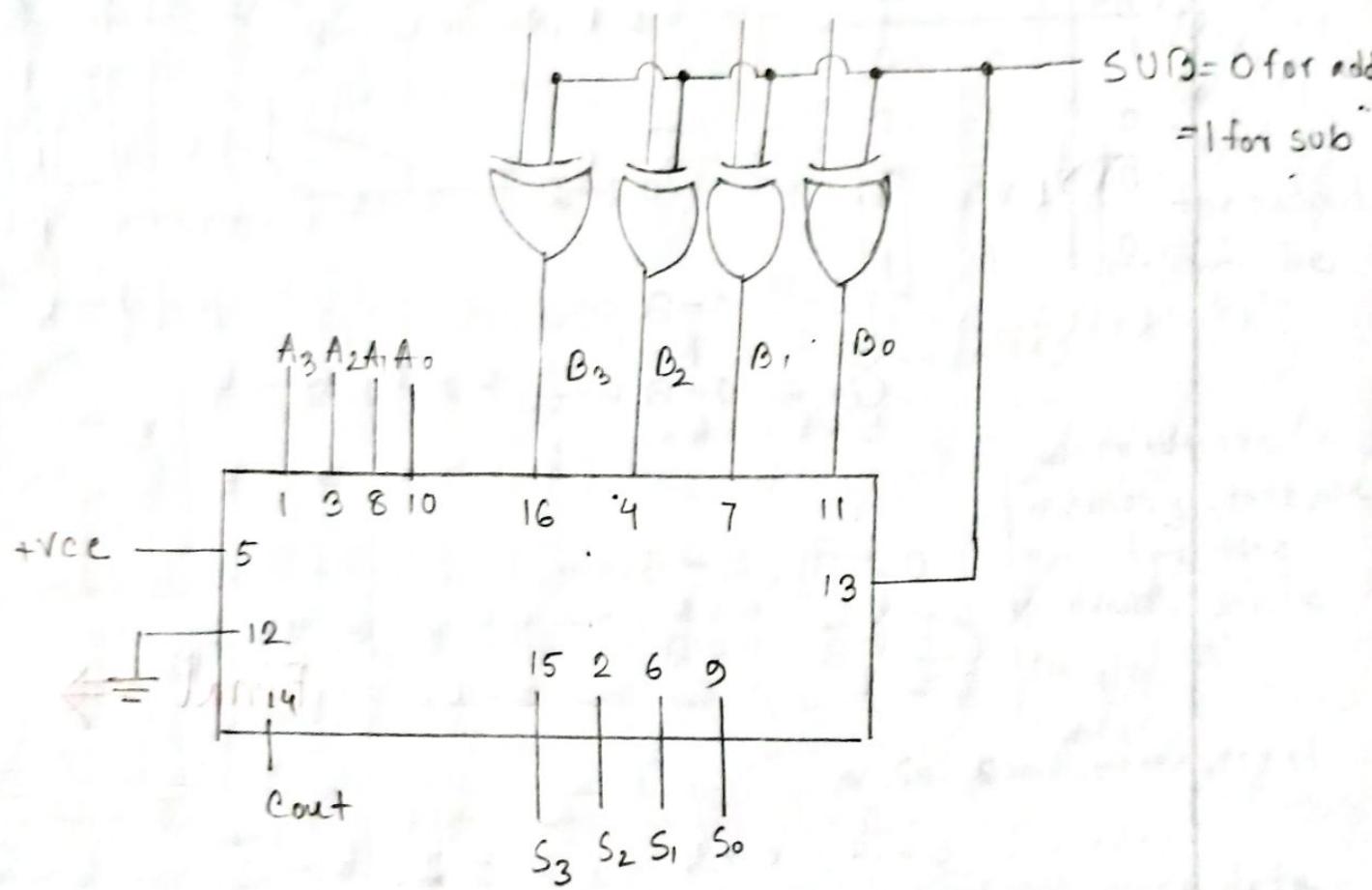
$$C = \sum (1, 2, 4, 7) \quad (F, S, S, U) Z = 3$$

(Multiplexer 8:1) see or even see



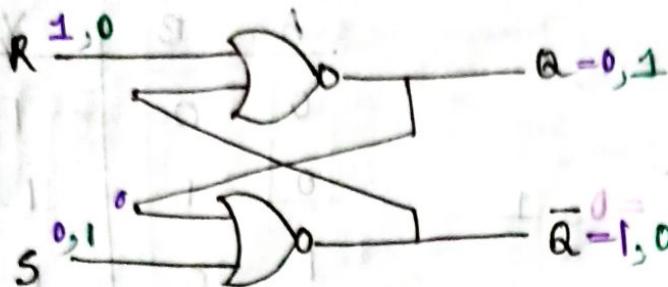
## Binary Adder Subtractor

4 bit



## SR Latch using (NOR)

truth-table for NOR Gate



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

if any input '1' the output will be '0'

case 1:  $S=0, R=1 \Rightarrow Q=0, \bar{Q}=1$ ,  
 $\Rightarrow S=0, R=0 \Rightarrow Q=0, \bar{Q}=1$

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

when we are in memory state, we consider the previous state output '1'

case 3:  $S=1, R=1 \Rightarrow Q=0, \bar{Q}=1$

$Q \rightarrow S=0, R=0 \Rightarrow Q=0, \bar{Q}=1$  but not store data

$\bar{Q} \rightarrow S=0, R=0 \Rightarrow Q=1, \bar{Q}=0$  are not same  
definitely not memory state

not use in latch and

flip flops

bit by bit

0 0

1 0

0 1

1 1

S	R	Q	$\bar{Q}$
0	0	0	1
0	1	0	1
1	0	1	0

(memory)

Truth Table for (NOR) SR Latch  $\Rightarrow$

S	R	Q	$\bar{Q}$
0	0	0	1
0	1	0	1
1	0	1	0

0 1

1 0

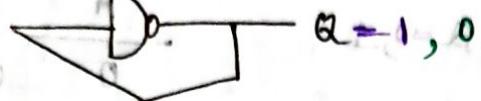
1 1

(invalid)

as per  
know it store  
data is so,  
data is already  
stored so, if it is  
memory state

## SR latch using (NAND) gate

$S \underline{0,1,0}$



$$Q = 1, 0$$

$R \underline{1,0,0}$

$$\bar{Q} = 0, 1$$

Truth Table for NAND gate

A	B	$\bar{Y}$	Y
0	0	1	1
0	1	0	0
1	0	0	1

input  
output

case 1:  $S = 0, R = 1 \Rightarrow Q = 1, \bar{Q} = 0$

memory state  $\Rightarrow S = 1, R = 1 \Rightarrow Q = 1, \bar{Q} = 0$

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

memory state  $\Rightarrow S = 1, R = 1 \Rightarrow Q = 0, \bar{Q} = 1$

case 3:  $S = 0, R = 0 \Rightarrow Q = 1, \bar{Q} = 1 \Rightarrow Q = \bar{Q}$

$Q \Rightarrow S = 1, R = 1 \Rightarrow Q = 1, \bar{Q} = 0$  (invalid)

$\bar{Q} \Rightarrow S = 1, R = 1 \Rightarrow \bar{Q} = 1, Q = 0$  memory state are not same

$\bar{Q}$	Q	S	R	Q	$\bar{Q}$
0	1	0	0	1	0

Truth table for

(NAND) SR-latch

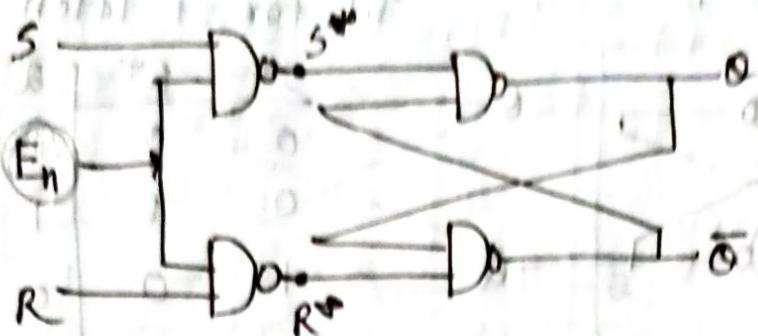
$\bar{Q}$	Q	S	R	Q	$\bar{Q}$
1	0	0	0	0	1
0	1	0	0	1	0
(bilavni)	1	1	1	1	1

invalid

bilavni

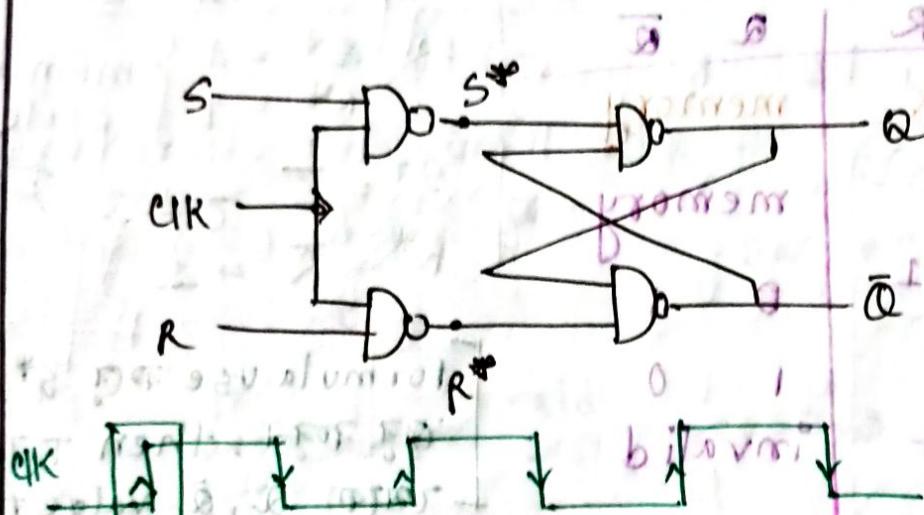
memory

## Difference between Latch and flip flop



this circuit will act like a latch when  $[En = 1]$

- \* this circuit will operate when Enable is high.
- \* But when we change the enable as low then  $S=0, R=0$ , in SR latch of (NAND) this is invalid state.



When we change the enable as clk then this circuit will act like SR flip flop

edge triggering

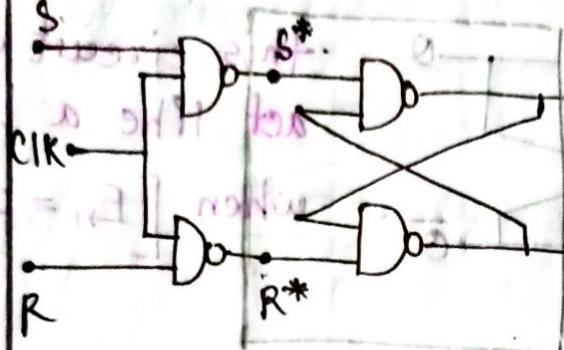
- \* this circuit will operate when clk is high to low or low to high.

## SR Flip-Flop (NAND)

T.T. for SR latch (NAND)

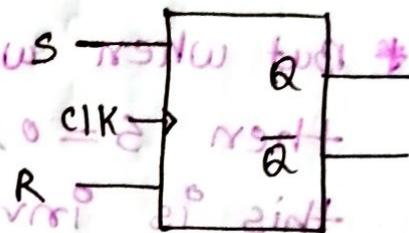
$S^*$	$R^*$	$Q$	$\bar{Q}$
0	0	invalid	
0	1	1	0
1	0	0	1

Memory



$$S^* = \overline{(S \cdot \overline{Q})} = \overline{S} + \overline{\overline{Q}} = \overline{S} + Q$$

$$R^* = \overline{(R \cdot \overline{Q})} = \overline{R} + \overline{\overline{Q}} = \overline{R} + Q$$



Truthtable for SR flip-flop

CLK	S	R	$Q$	$\bar{Q}$
0	X	X	memory	
1	0	0	memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	invalid	

$$\begin{cases} \text{CLK}=0, \overline{\text{CLK}}=1 \\ \text{So, } S^* = 1 \\ R^* = 1 \end{cases} \left\{ \begin{array}{l} \text{memory} \\ \text{state} \end{array} \right.$$

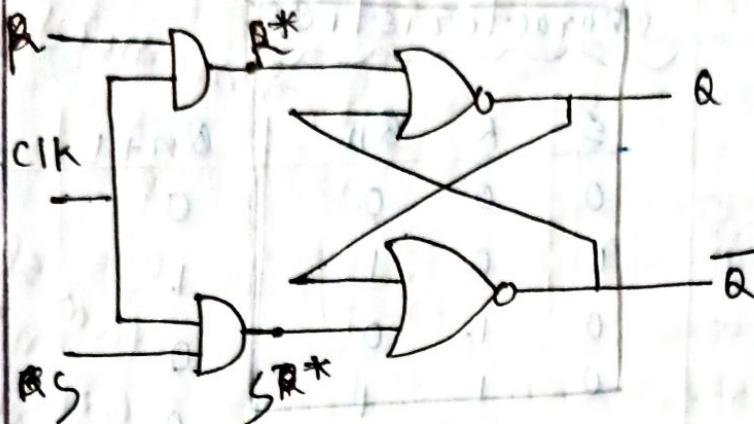
$$\begin{cases} S^* = \overline{S} = 1 \\ R^* = \overline{R} = 1 \end{cases} \left\{ \begin{array}{l} J \\ T \end{array} \right.$$

formula use করে  $S^*, R^*$   
কে বস্তুত, then T.T  
যথেক  $Q, \bar{Q}$  value ফল

[Define the 5 case of  
SR flip flop]

বিনামূলক  
input of we

## SR Flip-Flop (NOR)



$$S^* = S \cdot \text{CLK}$$

$$R^* = R \cdot \text{CLK}$$

Truth Table SR Latch (NOR)

$S^*$	$R^*$	$Q$
1	0	0
0	0	0
0	1	0
1	0	1
1	1	1

Memory      0      0  
                0      1  
                1      0  
                1      1  
invalid

Truth table for SR flip-flop

CLK	S	R	$Q_{(n+1)}$
0	X	X	$Q_n$
1	0	0	memory
1	0	1	0
1	1	0	1
1	1	1	invalid

case 1: CLK=0,  $S^*=0, R^*=0$   
So, in NOR SR latch  
it is memory state  
 $\Rightarrow Q_{(n+1)} = Q_n$

case 2: CLK=1,  $S^*=0, R^*=0$   
[same reason]

case 3: CLK=1,  $S^*=0, R^*=1$   
in NOR SR-latch  
 $Q=0, \text{ so, } Q_{n+1}=0$

SR latch flip flop (TT)  
are same for (NOR + NAND)

\* But NOR SR Latch  
and NAND SR Latch  
are different (TT).

[case 4, 5 as same as  
previous]

## Characteristic and excitation for SR flip-flop

### SR flip flop (T)

S	R	Q <sub>n+1</sub>
0	0	memory
0	1	0
1	0	1
1	1	invalid

### characteristics

S'	R	Q <sub>n</sub>	Q <sub>n+1</sub>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	X
1	1	0	X
1	1	1	X

### Excitation table

Q <sub>n</sub>	Q <sub>n+1</sub>	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

### Kmap for characteristic

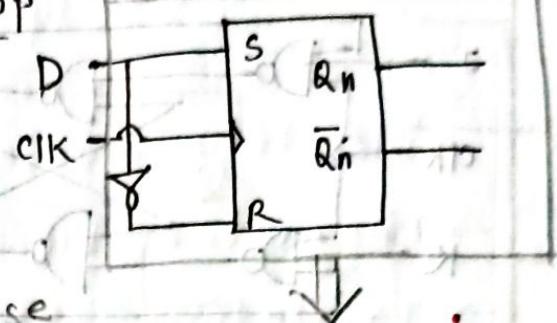
SR	Q <sub>n</sub>	0	1
00	0	0	1
01	1	1	1
10	2	1	1
11	3	X	X

$$F = S + \bar{R}Q_n$$

Data  
'D' Flip Flop (only for storing data)

Truth table for SR Flip-flop

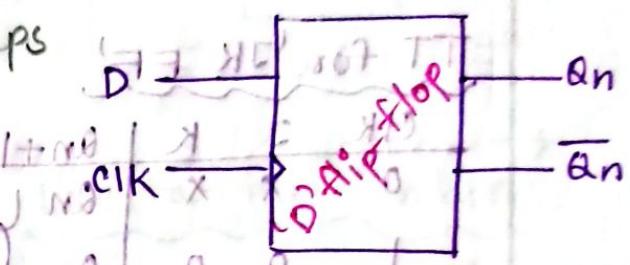
CLK	S	R	Q <sub>n+1</sub>
0	X	X	Q <sub>n</sub>
1	0	0	Q <sub>n</sub>
0	0	1	0
1	1	0	1
1	1	1	invalid



truth table for D flip flop

CLK      D      Q<sub>n+1</sub>

CLK	D	Q <sub>n+1</sub>
0	X	Q <sub>n</sub>
1	0	0
1	1	1



characteristic table and Excitation table

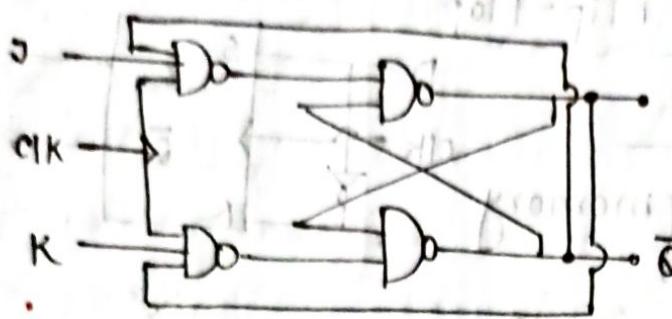
Q <sub>n</sub>	D	Q <sub>n+1</sub>
0	0	0
0	1	1
1	0	0
1	1	1

Q <sub>n</sub>	Q <sub>n+1</sub>	D
0	0	0
0	1	1
1	0	0
1	1	1

\* So, in here characteristic table and excitation table are same

(JK flip-flop)

We use it to define the invalid state of SR FF



TT of SR FF

CLK	S	R	$Q_{n+1}$
0	X	X	$Q_n$
1	0	0	$Q_n$
1	0	1	0
1	1	0	1
1	1	1	invalid

TT for JK FF

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

(toggle)

only change this state  
বাকি সব as same as  
(SR FF)

when  $CLK = 1, J = 1, K = 1$   
then whatever we input  
value  $Q = 0, 1, 0, 1, \dots$   
 $\bar{Q} = 1, 0, 1, 0, \dots$

characteristic

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

Excitation table

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

$$Q_{n+1} = \overline{Q_n}J + Q_nK$$

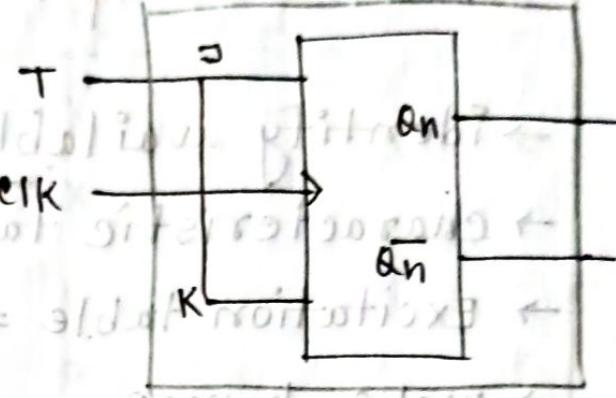
$$J = \overline{Q_n} + 1 \quad \text{From Kmap}$$

## (T' FLIP FLOP)

To define the toggle state of (JK FF)

### Truth table of T FF

$C_{IK}$	T	$Q_{n+1}$
0	X	$Q_n$
1	0	$\bar{Q}_n$
1	1	$\bar{Q}_n$ (toggle)



### Characteristic table → Excitation Table

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

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

$$Q_{n+1} = Q_n \oplus T \quad [\text{because here is odd '1'}]$$

(initial state of flip-flop)

(final state of flip-flop)

## \*\* Flip-Flop Conversion \*\*

- identify available and require flip-flop
- characteristic table  $\Rightarrow$  required FF
- Excitation table  $\Rightarrow$  Available FF
- make K-map
- \* → draw the circuit

T	1+R	M	T	1+R	M
0	0	0	0	0	0
1	1	0	1	1	0
1	0	1	1	0	1
0	1	1	0	1	1

$$B = P + Q \cdot T \cdot M \quad T \oplus M = 1 + R$$

available  
JK to D FlipFlop

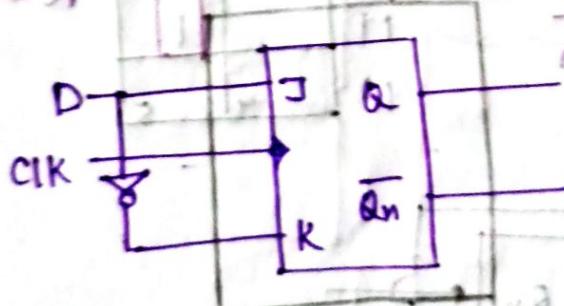
$Q_n$	D	$Q_{n+1}$	J	K	A	B	C
0	0 X	0	0	X	0	0	0
0	1 J	1	1	X	1	0	0
1	0 X	0	0	1	0	1	0
1	1 J	1	X	0	0	0	1
					1	0	1
					0	1	1
					1	0	0

k-map for J

$Q_n$	D	0	1
0	0	1	X
1	X	X	1

$$J = D$$

$$\text{and } X = X$$



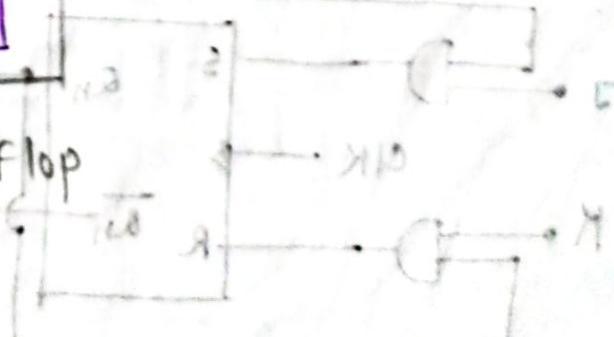
k-map for K

$Q_n$	D	0	1
0	X	X	1
1	1	0	0

$$K = \overline{D}$$

$Q_n$	D	0	1
0	X	0	0
1	1	0	1

diagram: D FlipFlop



~~SR to JK~~

J	K	$Q_n$	$Q_{n+1}$	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

K-map for S

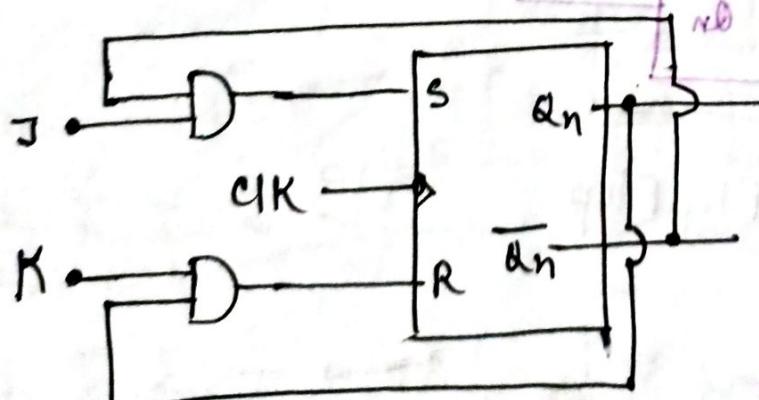
JK	$Q_n$	0	1
00	0	X <sub>1</sub>	
01	1		X <sub>3</sub>
11	1		
10	0	X <sub>4</sub>	X <sub>5</sub>

$$S = \overline{J} Q_n$$

K-map for R

JK	$Q_n$	0	1
00	0	X <sub>0</sub>	
01	1	X <sub>2</sub>	X <sub>3</sub>
11	1		
10	0	X <sub>4</sub>	X <sub>5</sub>

$$R = K Q_n$$



T to D

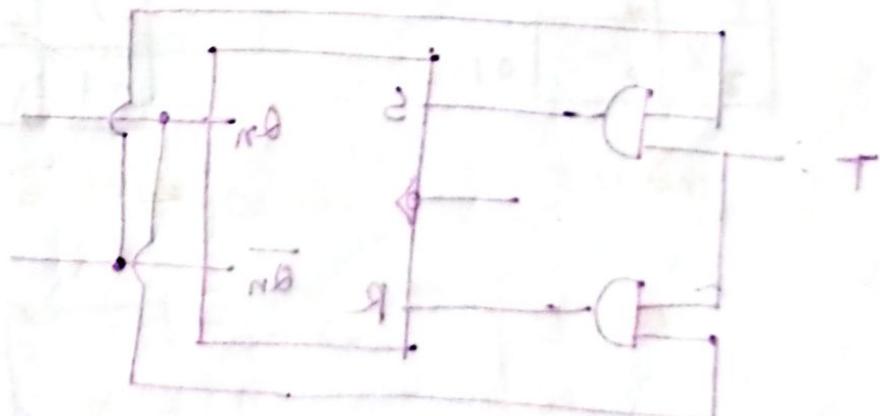
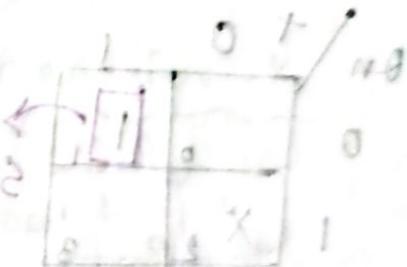
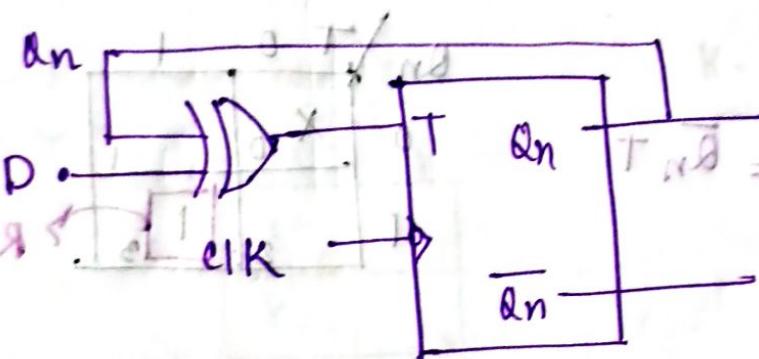
$Q_n$	D	$Q_{n+1}$	T	$T_{\text{odd}}$	$T_{\text{even}}$
0	0	0	0	0	0
0	1	1	1	1	0
1	0	0	1	0	1
1	1	1	0	0	0

$T_{\text{odd}} = T \oplus Q_n$

$T_{\text{even}} = T \oplus \bar{Q}_n$

\* no need K-map

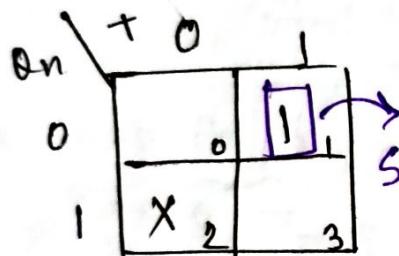
$T_{\text{odd}} = T \oplus Q_n$  when the input has odd number of 1's



SR to T

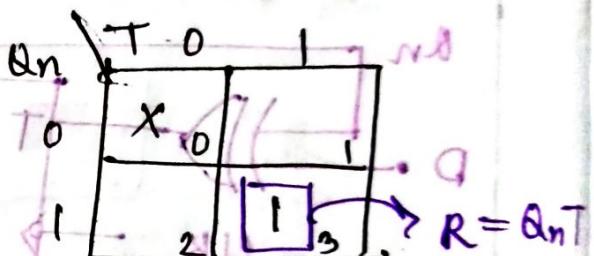
$Q_n$	T	$Q_{n+1}$	S	R	Q <sub>n+1</sub>
0	0	0	0	0	X
0	1	1	1	0	1
1	0	1	X	0	0
1	1	0	0	1	1

K-map for S

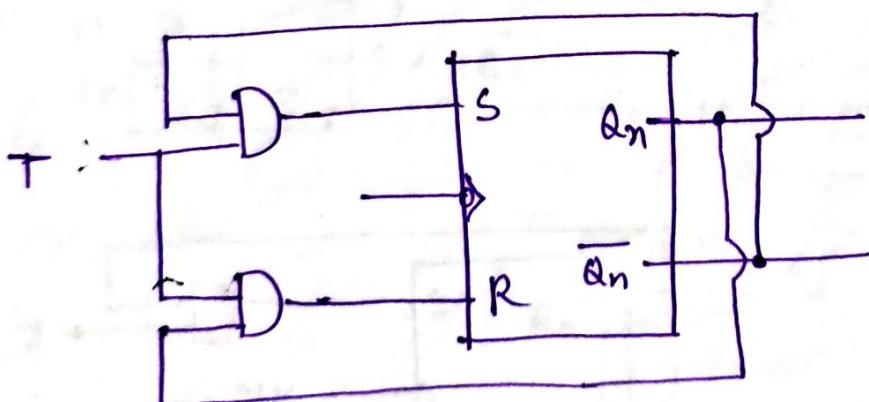


$$S = \bar{Q}_n T$$

K-map for R



$$R = Q_n T$$



JK to SR

S	R	$Q_n$	$Q_{n+1}$	J	K	S	R
0	0	0	0	X	1	0	X
0	0	1	1	1	X	X	0
0	1	0	0	X	0	0	X
0	1	1	0	0	X	0	0
1	0	0	1	X	0	X	1
1	0	1	1	1	X	1	X
1	1	0	X	X	X	X	0
1	1	1	X	X	X	X	X

K-map for J

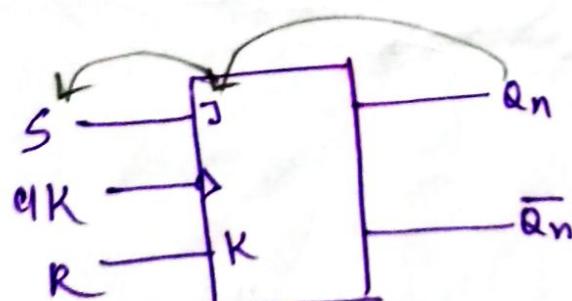
SR \ $Q_n$		0	1
0	0	X	1
0	1	X	1
1	0	X	1
1	1	1	X

$$Q_n = Q_n$$

K-map for K

SR \ $Q_n$		0	1
0	0	1	X
0	1	1	1
1	0	X	1
1	1	1	X

$$K = \bar{Q}_n$$



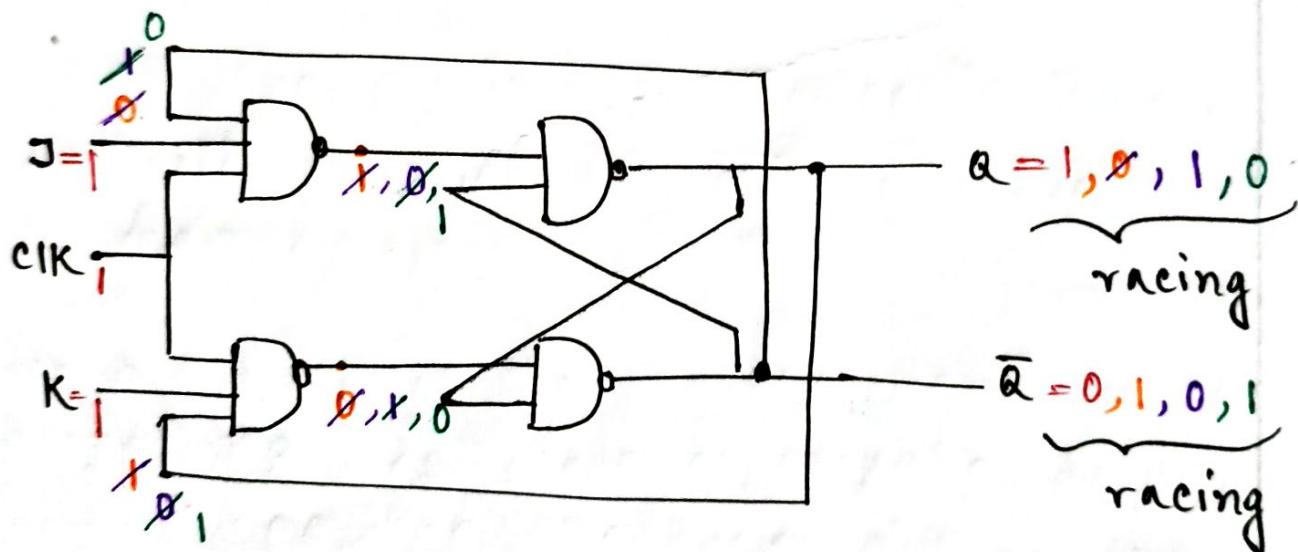
$$\begin{cases} J = Q_n \\ S_0, S = J \\ K = \bar{Q}_n \\ S_0, R = K \end{cases}$$

## Race around Condition

We use this condition for JK flipflop because in there we see in that state when  $\text{CLK} = 1, J = 1, K = 1$  then the output will,  $Q = 1, 0, 1, 0 \dots$  } (racing)  
 $\bar{Q} = 0, 1, 0, 1 \dots$  } (racing)

"racing and toggling are not same"  
 because we control ~~racing~~-toggling but racing will not to control. So, we have to make this output toggle."

we assume that the previous state output  $\Rightarrow Q = 1, \bar{Q} = 0$



<u>CLK</u>	<u>J</u>	<u>K</u>	<u>Q<sub>n+1</sub></u>	<u>Q̄<sub>n+1</sub></u>
0	X	X	Q <sub>n</sub>	Q̄ <sub>n</sub>
1	0	0	Q <sub>n</sub>	Q̄ <sub>n</sub>
1	0	1	0	1
1	1	0	1	0
1	1	1	?	?

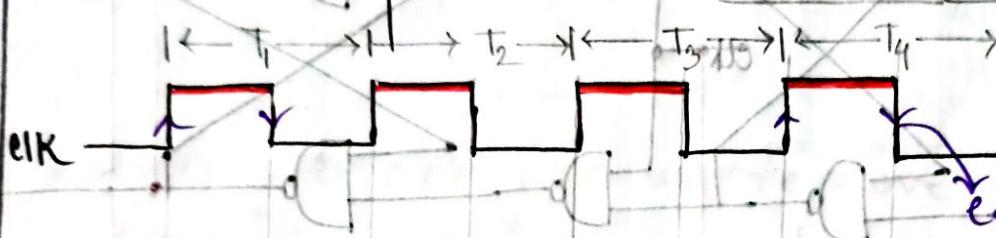
initial state

from SRJK flip flop \*

state after  $T/2$  delay

delay  $\Rightarrow$  output change of system  
মাত্র স্থিতি পরিবর্তন

System এর নির্ভুল  
ব্যবহার করতে হবে  
সময় লাগে,  
মেরে এক  $clk = \text{High}$   
moment থেকে  
আপরা output  
গাই, তাই  $T/2$   
delay  $\Rightarrow$  হবে



edge triggering

J=1

K=1

output racing

condition to overcome Racing:

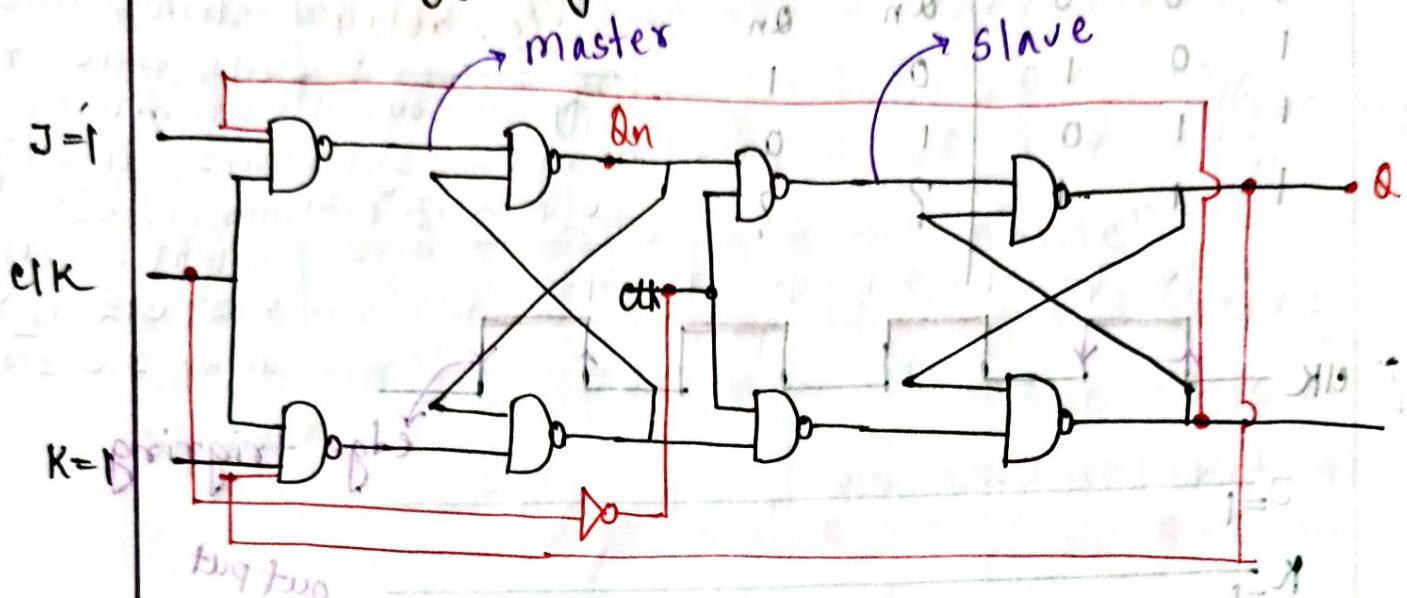
i.  $T/2 <$  propagation delay of FF

ii. edge triggering [positive; Negative]

iii Master slave

## Master Slave

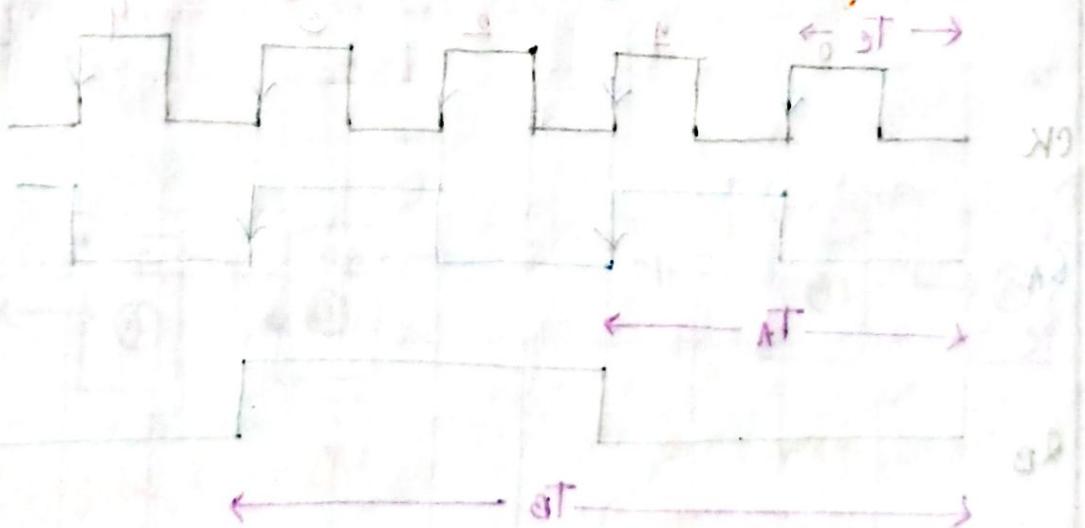
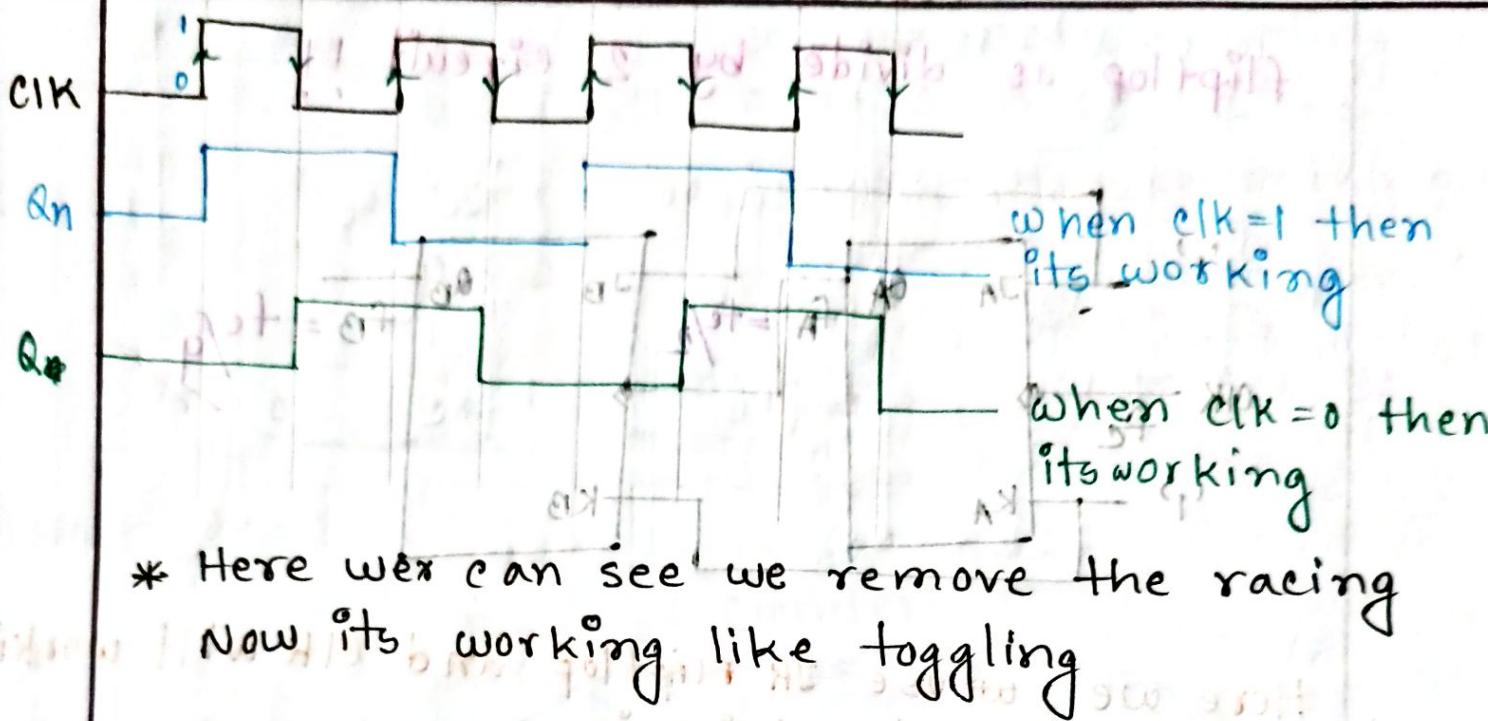
\* Master slave operation is same as negative edge triggering.



\* When the clk is high Master (FF) operating as Slave has given the  $\overline{\text{clk}}$  so slave work as memory state.

\* When clk is low for master then this can't operating, as slave as operate as well and we see the feedback will go the master but as master will not operating so, there is no need to feed back

We see in racing output will change 0 to 1 but for reason-2, output will change once. that is called toggling.



$$ST^S = \overline{AT} \Leftrightarrow ST^S = AT$$

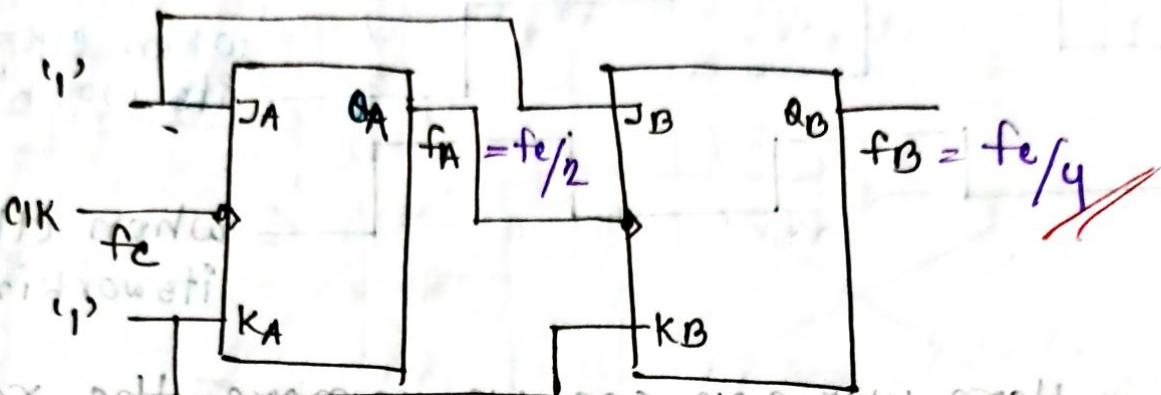
$$ST = \overline{AT} \Leftrightarrow ST = AT$$

$$AT^S = ST^S = NS = \overline{S}$$

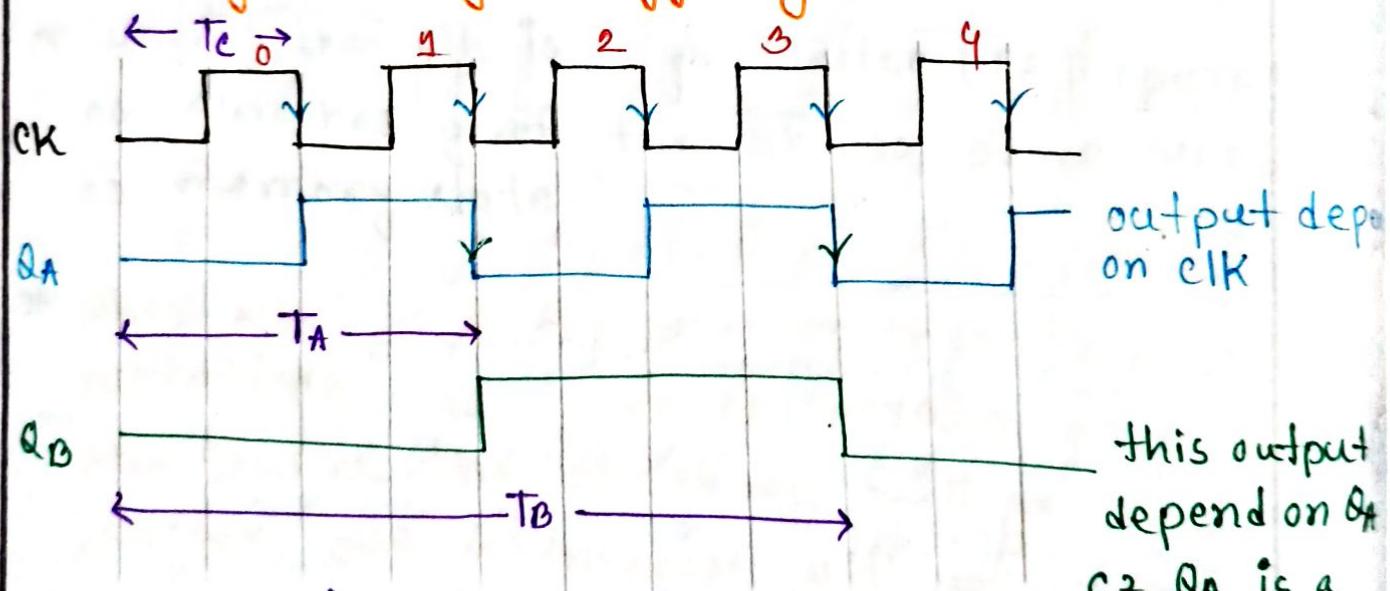
$$P \setminus ST = S \setminus AT = ST$$

## Counter

flip flop as divide by 2 circuit !!



Here we use JK flip flop and CIK will work on negative edge triggering



$$T_A = 2T_C \Rightarrow \frac{1}{f_A} = 2/f_C \\ \therefore f_A = f_C/2$$

$$T_B = 2T_A = \frac{1}{f_B} = 2/f_A$$

$$\therefore f_B = f_A/2 = f_C/4$$

N of flip flop

N = No of flip flop

$$2^N = 2^2 = 4$$

[that means frequency will divided by 4]

this output depend on Q\_A  
cz Q\_A is a CIK for second JK flip flop

CLK	$Q_0$	$Q_1$
0	0	0 (0)
1	0	1 (1)
2	1	0 (2)
3	1	1 (3)
4	0	0 [repeating]

counter means counting  
 $0 \text{ to } 3 \Rightarrow 2^N = 2^2 = 4$

[means after counting 3 it repeats its value]

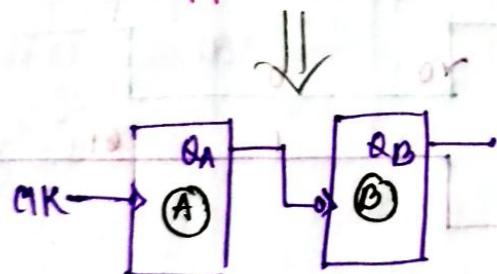
$0 \text{ to } 15 \Rightarrow 2^4 = 16$

[after counting 16 it repeats its value]

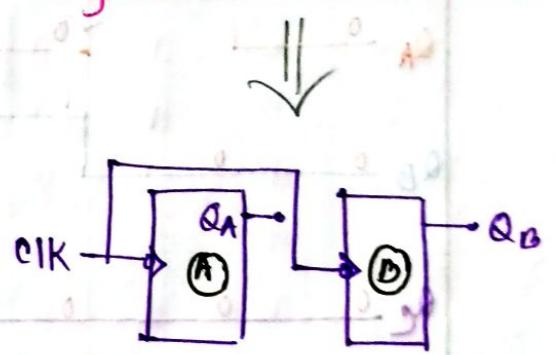
### Types of Counters

Asynchronous counters  
 (Ripple Counter)

Synchronous counters



- \* Simple Circuit
- \* speed is slow



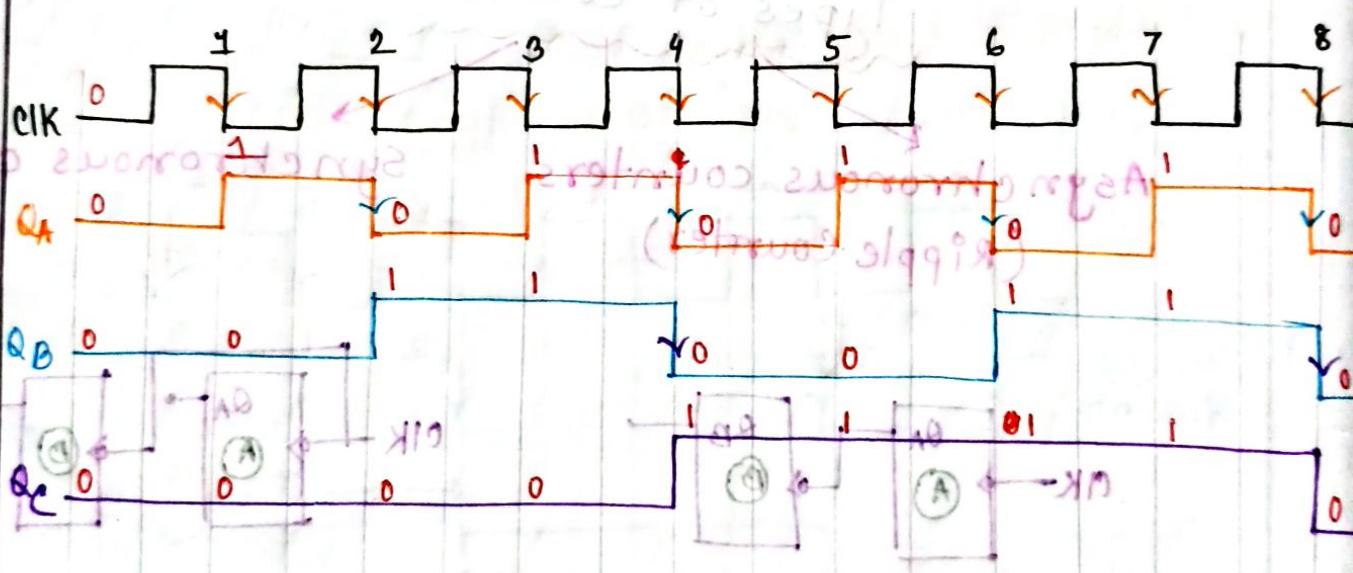
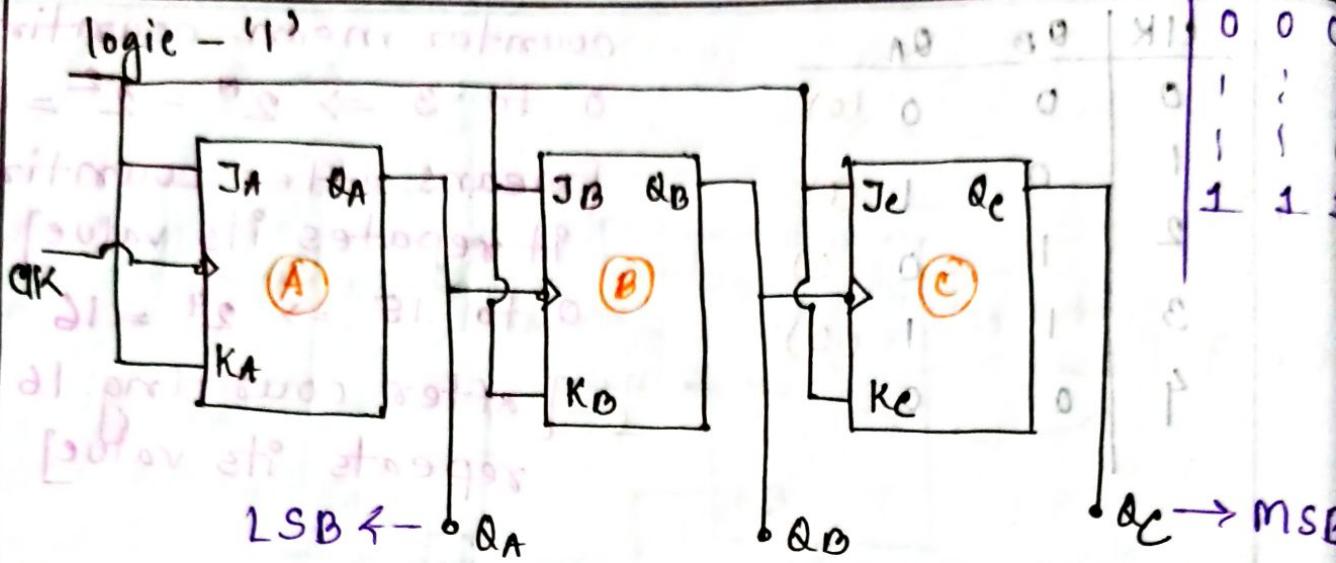
- \* complicated
- \* high speed.

counters →

- up counter ( $0, 1, 2, 3 \dots$ )
- down counter ( $7, 6, 5, 4 \dots 0$ )
- up/down counter (combination)

UP

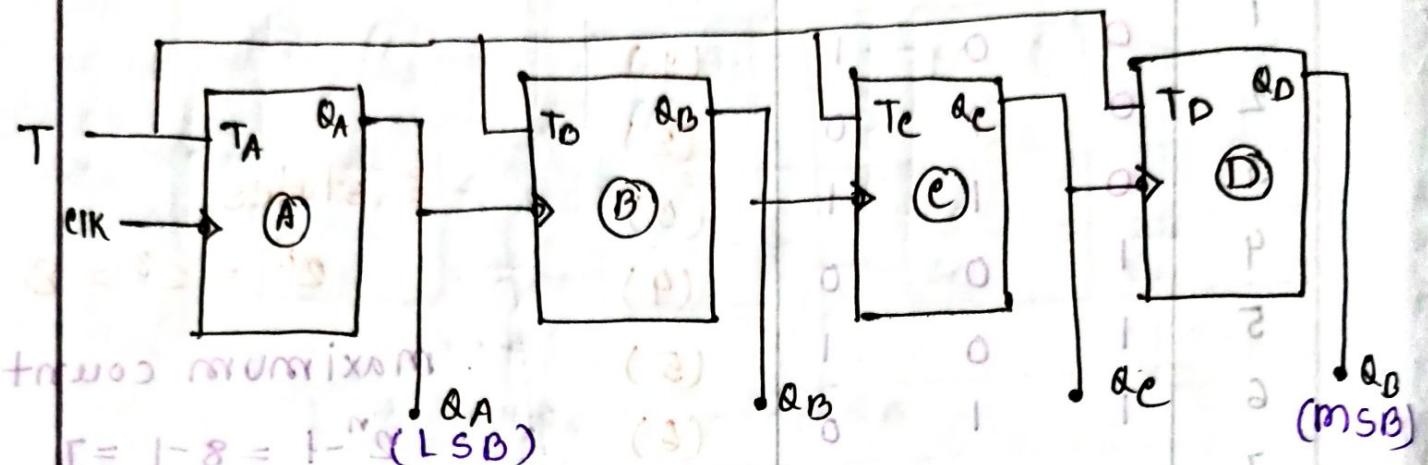
## '3 bit Asynchronous Counter' (JK)



CLK	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	Decimal
0	0	0	0	(0)
1	0	0	1	(1)
2	0	1	0	(2)
3	0	1	1	(3)
4	1	0	0	(4)
5	1	0	1	(5)
6	1	1	0	(6)
7	1	1	1	(7)
8	0	0	0	(0) $\leftarrow$ repeating
9	0	0	0	(1)
10	0	1	0	(2)
11	1	0	0	(3)
12	1	0	1	(4)
13	0	1	0	(5)
14	1	1	0	(6)
15	0	0	1	(7)
16	0	0	0	(0)
17	0	0	0	(1)
18	0	1	0	(2)
19	1	0	0	(3)
20	1	0	1	(4)
21	0	1	0	(5)
22	1	1	0	(6)
23	0	0	1	(7)
24	0	0	0	(0)
25	0	0	0	(1)
26	0	1	0	(2)
27	1	0	0	(3)
28	1	0	1	(4)
29	0	1	0	(5)
30	1	1	0	(6)
31	0	0	1	(7)
32	0	0	0	(0)
33	0	0	0	(1)
34	0	1	0	(2)
35	1	0	0	(3)
36	1	0	1	(4)
37	0	1	0	(5)
38	1	1	0	(6)
39	0	0	1	(7)
40	0	0	0	(0)
41	0	0	0	(1)
42	0	1	0	(2)
43	1	0	0	(3)
44	1	0	1	(4)
45	0	1	0	(5)
46	1	1	0	(6)
47	0	0	1	(7)
48	0	0	0	(0)
49	0	0	0	(1)
50	0	1	0	(2)
51	1	0	0	(3)
52	1	0	1	(4)
53	0	1	0	(5)
54	1	1	0	(6)
55	0	0	1	(7)
56	0	0	0	(0)
57	0	0	0	(1)
58	0	1	0	(2)
59	1	0	0	(3)
60	1	0	1	(4)
61	0	1	0	(5)
62	1	1	0	(6)
63	0	0	1	(7)
64	0	0	0	(0)
65	0	0	0	(1)
66	0	1	0	(2)
67	1	0	0	(3)
68	1	0	1	(4)
69	0	1	0	(5)
70	1	1	0	(6)
71	0	0	1	(7)
72	0	0	0	(0)
73	0	0	0	(1)
74	0	1	0	(2)
75	1	0	0	(3)
76	1	0	1	(4)
77	0	1	0	(5)
78	1	1	0	(6)
79	0	0	1	(7)
80	0	0	0	(0)
81	0	0	0	(1)
82	0	1	0	(2)
83	1	0	0	(3)
84	1	0	1	(4)
85	0	1	0	(5)
86	1	1	0	(6)
87	0	0	1	(7)
88	0	0	0	(0)
89	0	0	0	(1)
90	0	1	0	(2)
91	1	0	0	(3)
92	1	0	1	(4)
93	0	1	0	(5)
94	1	1	0	(6)
95	0	0	1	(7)
96	0	0	0	(0)
97	0	0	0	(1)
98	0	1	0	(2)
99	1	0	0	(3)
100	1	0	1	(4)
101	0	1	0	(5)
102	1	1	0	(6)
103	0	0	1	(7)
104	0	0	0	(0)
105	0	0	0	(1)
106	0	1	0	(2)
107	1	0	0	(3)
108	1	0	1	(4)
109	0	1	0	(5)
110	1	1	0	(6)
111	0	0	1	(7)
112	0	0	0	(0)
113	0	0	0	(1)
114	0	1	0	(2)
115	1	0	0	(3)
116	1	0	1	(4)
117	0	1	0	(5)
118	1	1	0	(6)
119	0	0	1	(7)
120	0	0	0	(0)
121	0	0	0	(1)
122	0	1	0	(2)
123	1	0	0	(3)
124	1	0	1	(4)
125	0	1	0	(5)
126	1	1	0	(6)
127	0	0	1	(7)
128	0	0	0	(0)
129	0	0	0	(1)
130	0	1	0	(2)
131	1	0	0	(3)
132	1	0	1	(4)
133	0	1	0	(5)
134	1	1	0	(6)
135	0	0	1	(7)
136	0	0	0	(0)
137	0	0	0	(1)
138	0	1	0	(2)
139	1	0	0	(3)
140	1	0	1	(4)
141	0	1	0	(5)
142	1	1	0	(6)
143	0	0	1	(7)
144	0	0	0	(0)
145	0	0	0	(1)
146	0	1	0	(2)
147	1	0	0	(3)
148	1	0	1	(4)
149	0	1	0	(5)
150	1	1	0	(6)
151	0	0	1	(7)
152	0	0	0	(0)
153	0	0	0	(1)
154	0	1	0	(2)
155	1	0	0	(3)
156	1	0	1	(4)
157	0	1	0	(5)
158	1	1	0	(6)
159	0	0	1	(7)
160	0	0	0	(0)
161	0	0	0	(1)
162	0	1	0	(2)
163	1	0	0	(3)
164	1	0	1	(4)
165	0	1	0	(5)
166	1	1	0	(6)
167	0	0	1	(7)
168	0	0	0	(0)
169	0	0	0	(1)
170	0	1	0	(2)
171	1	0	0	(3)
172	1	0	1	(4)
173	0	1	0	(5)
174	1	1	0	(6)
175	0	0	1	(7)
176	0	0	0	(0)
177	0	0	0	(1)
178	0	1	0	(2)
179	1	0	0	(3)
180	1	0	1	(4)
181	0	1	0	(5)
182	1	1	0	(6)
183	0	0	1	(7)
184	0	0	0	(0)
185	0	0	0	(1)
186	0	1	0	(2)
187	1	0	0	(3)
188	1	0	1	(4)
189	0	1	0	(5)
190	1	1	0	(6)
191	0	0	1	(7)
192	0	0	0	(0)
193	0	0	0	(1)
194	0	1	0	(2)
195	1	0	0	(3)
196	1	0	1	(4)
197	0	1	0	(5)
198	1	1	0	(6)
199	0	0	1	(7)
200	0	0	0	(0)
201	0	0	0	(1)
202	0	1	0	(2)
203	1	0	0	(3)
204	1	0	1	(4)
205	0	1	0	(5)
206	1	1	0	(6)
207	0	0	1	(7)
208	0	0	0	(0)
209	0	0	0	(1)
210	0	1	0	(2)
211	1	0	0	(3)
212	1	0	1	(4)
213	0	1	0	(5)
214	1	1	0	(6)
215	0	0	1	(7)
216	0	0	0	(0)
217	0	0	0	(1)
218	0	1	0	(2)
219	1	0	0	(3)
220	1	0	1	(4)
221	0	1	0	(5)
222	1	1	0	(6)
223	0	0	1	(7)
224	0	0	0	(0)
225	0	0	0	(1)
226	0	1	0	(2)
227	1	0	0	(3)
228	1	0	1	(4)
229	0	1	0	(5)
230	1	1	0	(6)
231	0	0	1	(7)
232	0	0	0	(0)
233	0	0	0	(1)
234	0	1	0	(2)
235	1	0	0	(3)
236	1	0	1	(4)
237	0	1	0	(5)
238	1	1	0	(6)
239	0	0	1	(7)
240	0	0	0	(0)
241	0	0	0	(1)
242	0	1	0	(2)
243	1	0	0	(3)
244	1	0	1	(4)
245	0	1	0	(5)
246	1	1	0	(6)
247	0	0	1	(7)
248	0	0	0	(0)
249	0	0	0	(1)
250	0	1	0	(2)
251	1	0	0	(3)
252	1	0	1	(4)
253	0	1	0	(5)
254	1	1	0	(6)
255	0	0	1	(7)
256	0	0	0	(0)
257	0	0	0	(1)
258	0	1	0	(2)
259	1	0	0	(3)
260	1	0	1	(4)
261	0	1	0	(5)
262	1	1	0	(6)
263	0	0	1	(7)
264	0	0	0	(0)
265	0	0	0	(1)
266	0	1	0	(2)
267	1	0	0	(3)
268	1	0	1	(4)
269	0	1	0	(5)
270	1	1	0	(6)
271	0	0	1	(7)
272	0	0	0	(0)
273	0	0	0	(1)
274	0	1	0	(2)
275	1	0	0	(3)
276	1	0	1	(4)
277	0	1	0	(5)
278	1	1	0	(6)
279	0	0	1	(7)
280	0	0	0	(0)
281	0	0	0	(1)
282	0	1	0	(2)
283	1	0	0	(3)
284	1	0	1	(4)
285	0	1	0	(5)
286	1	1	0	(6)
287	0	0	1	(7)
288	0	0	0	(0)
289	0	0	0	(1)
290	0	1	0	(2)
291	1	0	0	(3)
292	1	0	1	(4)
293	0	1	0	(5)
294	1	1	0	(6)
295	0	0	1	(7)
296	0	0	0	(0)
297	0	0	0	(1)
298	0	1	0	(2)
299	1	0	0	(3)
300	1	0	1	(4)
301	0	1	0	(5)
302	1	1	0	(6)
303	0	0	1	(7)
304	0	0</		

## 4 Bit Asynchronous Up Counter (T)

"We use T flip Flop because T stands for toggling"



CLK	QD	Qc	QB	QA	Decimal
0	0	0	0	0	(0)
1	0	0	0	1	(1)
2	0	0	1	0	(2)
3	0	0	1	1	(3)
4	0	1	0	0	(4)
5	0	1	0	1	(5)
6	0	1	1	0	(6)
7	0	1	1	1	(7)
8	1	0	0	0	(8)
9	1	0	0	1	(9)
10	1	0	1	0	(10)
11	1	0	1	1	(11)
12	1	1	0	0	(12)
13	1	1	0	1	(13)
14	1	1	1	0	(14)
15	1	1	1	1	(15)

16 states

$$\{2^4 = 16\}$$

maximum count

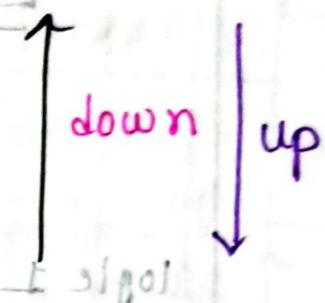
$$2^n - 1 = 2^4 - 1 = 15$$

## "State Diagram"

2 bit Up - Counter

$$\begin{aligned} \text{Maximum count} &= 2^n - 1 \\ &= 2^2 - 1 = 3 \end{aligned}$$

$Q_A$	$Q_B$
0	0
0	1
1	0
1	1



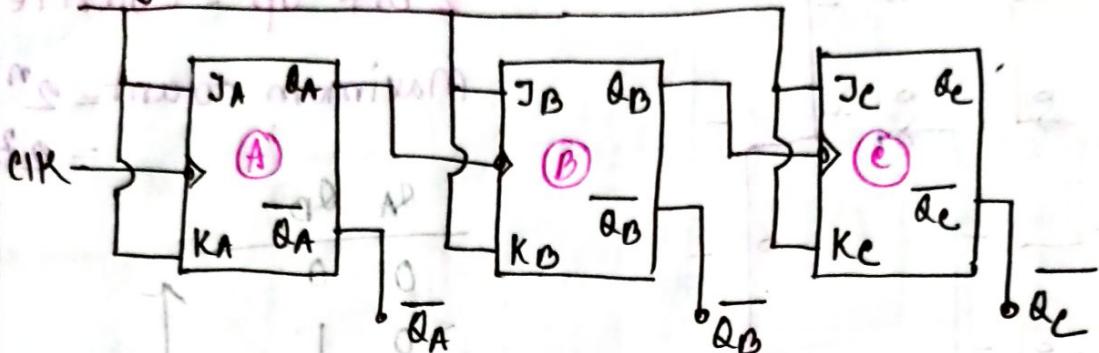
2-bit UP Counter

"B-bit up-down Counter"

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

## 3 bit up-down Counter (Asynchronous)

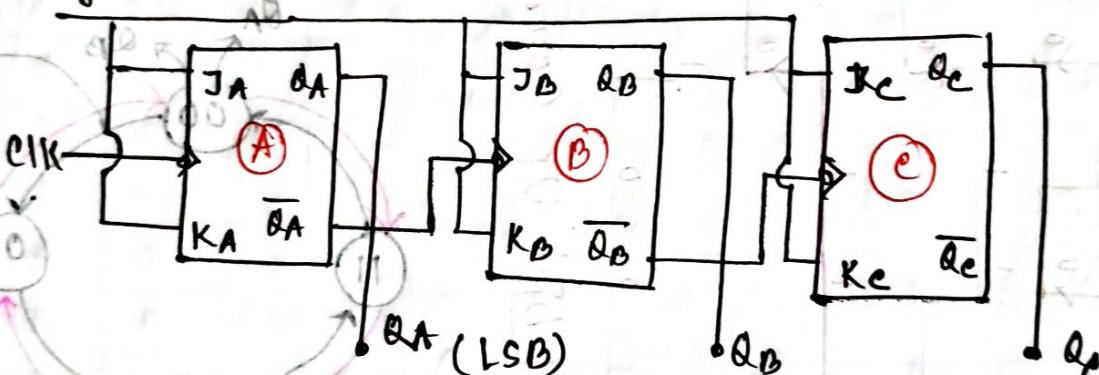
Logic - 1



(LSB)

(UP Counter)

Logic 1

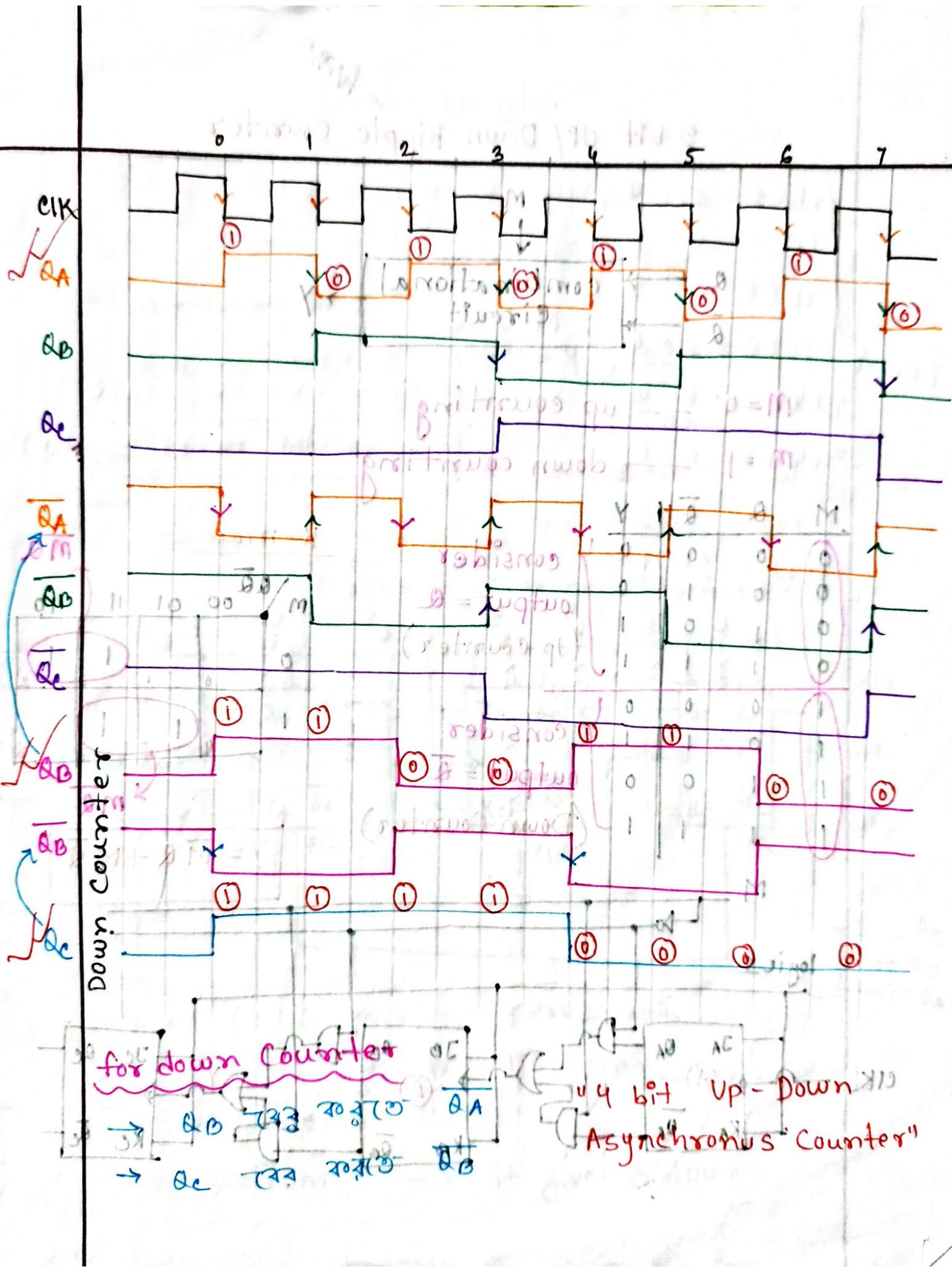


(Down Counter)

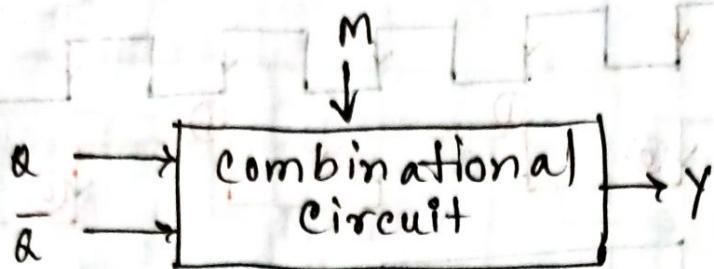
up Counter      Down Counter

CLK	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
0	0	0	0	1	1	1
1	0	0	1	1	1	0
2	0	1	0	1	0	1
3	0	1	1	1	0	0
4	1	0	0	0	1	1
5	1	0	1	0	1	0
6	1	1	0	0	0	1
7	1	1	1	0	0	0

down  
counter



## 3 bit UP/ Down Ripple Counter



$M=0$  -- up counting

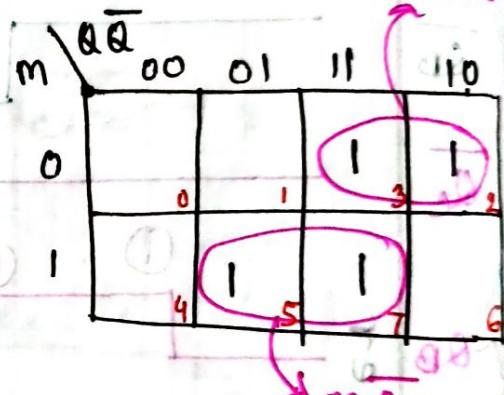
$M=1$  -- down counting

M	Q	$\bar{Q}$	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
<hr/>			
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

consider  
output = Q  
(Up Counter)

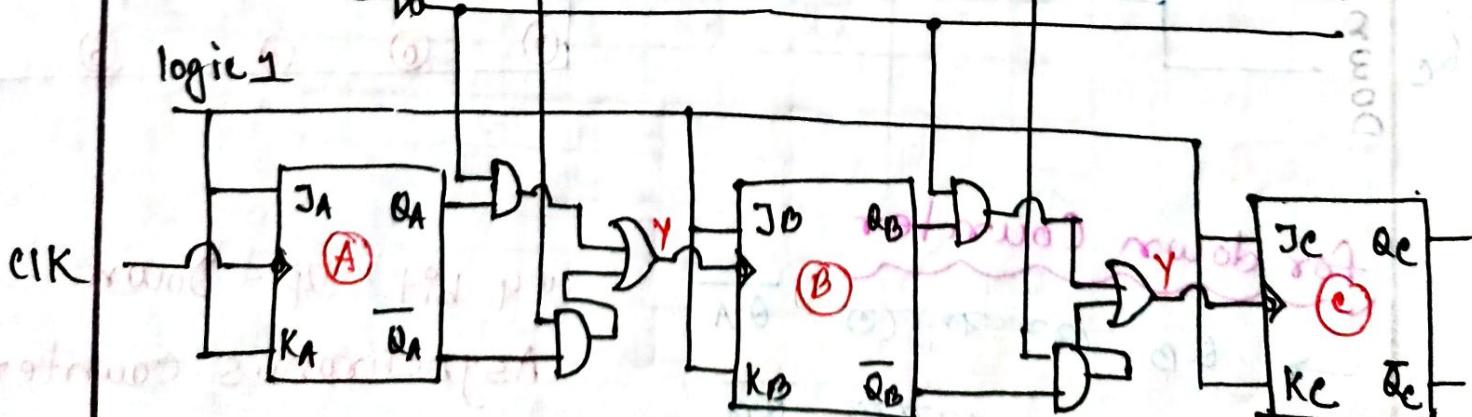
consider  
output =  $\bar{Q}$   
(Down Counter)

K-map



$$Y = \bar{M}Q + M\bar{Q}$$

logic 1



## Modulus Counter

→ 2 bit ripple counter  $\Rightarrow$  MOD-4 counter

→ 3 bits  $\Rightarrow$  MOD-8

$n = \text{no of bits}$

$$\text{MOD number} = 2^n = 2^2 = 4, 2^3 = 8$$

**MOD = State**

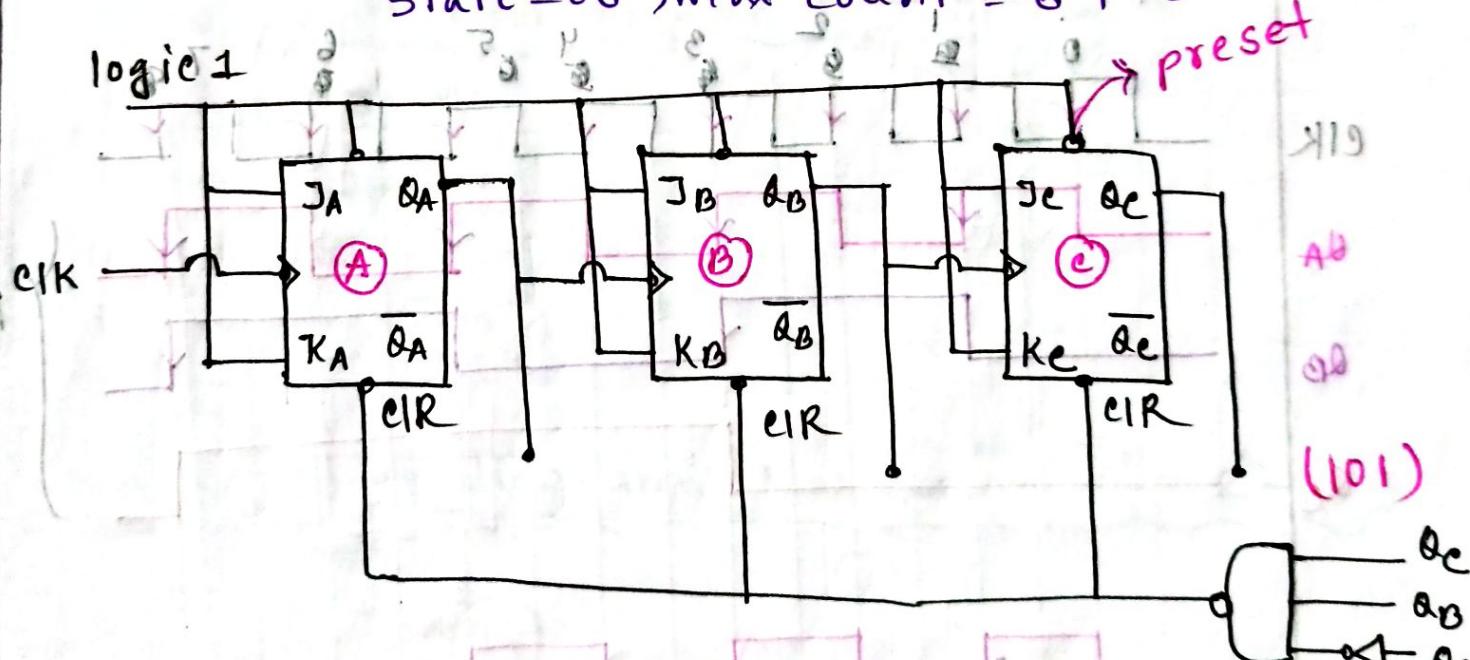
preset = 0,  $Q=1$

$\text{CLR} = 0, Q = 0$

always  
post = 1

\* Designing MOD-6 using MOD-8

State = 06, max count =  $6-1 = 5$



When (110) arrive then NAND gate give 0, so, system will clear and again consider (000).

because from 0 - 5 it gives output



## Synchronous Counters

\* Designing 2-bit synchronous up counter

Steps:

→ number of flipflop

→ Excitation table of flipflop

→ draw state diagram and circuit excitation table

→ K-map

→ logic diagram

Step-2: Excitation table of flipflop

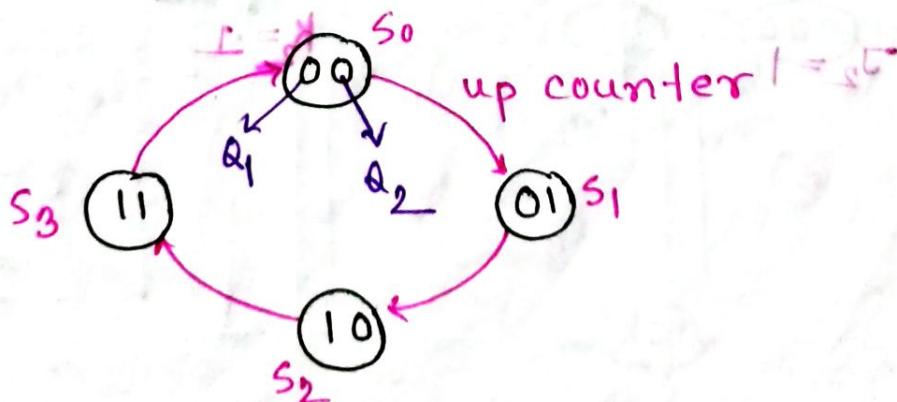
$Q_n$	$Q_{n+1}$	$J$	$K$
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$s_0$	$s_1$	$s_2$	$s_3$
X	X	.	1
1	.	.	0
.	.	.	0
0	0	1	0

$$s_0 = 15$$

$s_0$	$s_1$	$s_2$	$s_3$
X	1	.	0
1	.	.	0
.	.	.	0
0	0	1	0

Step-3: State diagram, Max count,  $2^n = 2^2 - 1 = 3$



## circuit excitation table

$Q_1$	$Q_2$	$Q_1^*$	$Q_2^*$	$J_1$	$K_1$	$J_2$	$K_2$
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	1	0	0	X	1	X	1

Step: 4 K-map

$Q_1$	$Q_2$	0	1
0	1	X	1
1	X	X	1

$$J_1 = Q_2$$

$Q_1$	$Q_2$	0	1
0	1	X	
1	1	X	

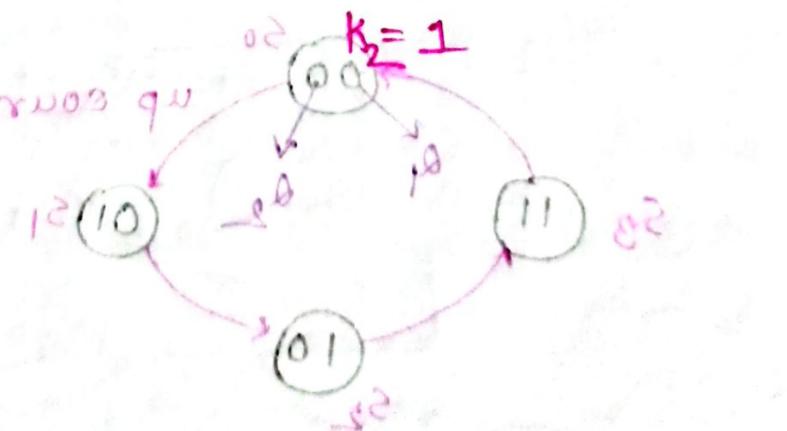
$Q_1$	$Q_2$	0	1
0	X	X	1
1	X	0	0

$$K_1 = Q_2$$

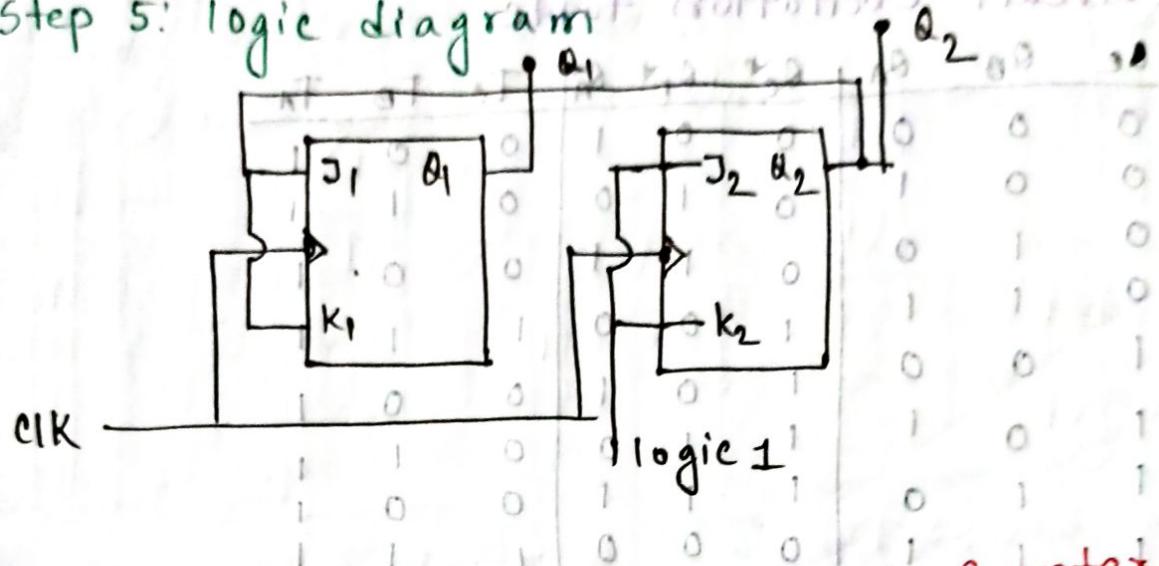
$Q_1$	$Q_2$	0	1
0	X	1	
1	X	1	

$$K_2 = 1$$

$$J_2 = 1 \text{ returns qu}$$



### Step 5: logic diagram

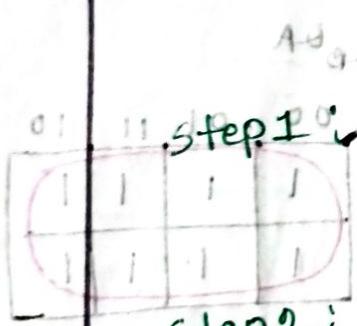


3 bit Synchronous Up-Counter

Step 1 ✓

No. of flip flop = 3

Type of flip flop = T flip flop

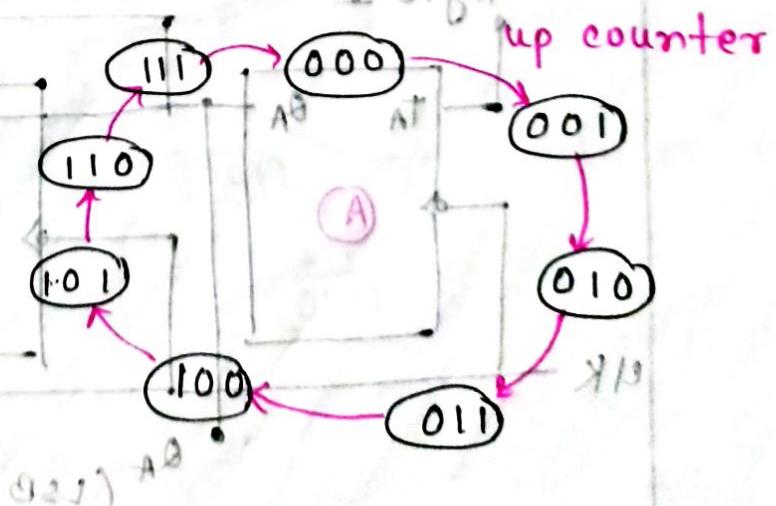


Step 2: excitation table of T FF  
 $AB = ST$

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

Step 3: state diagram

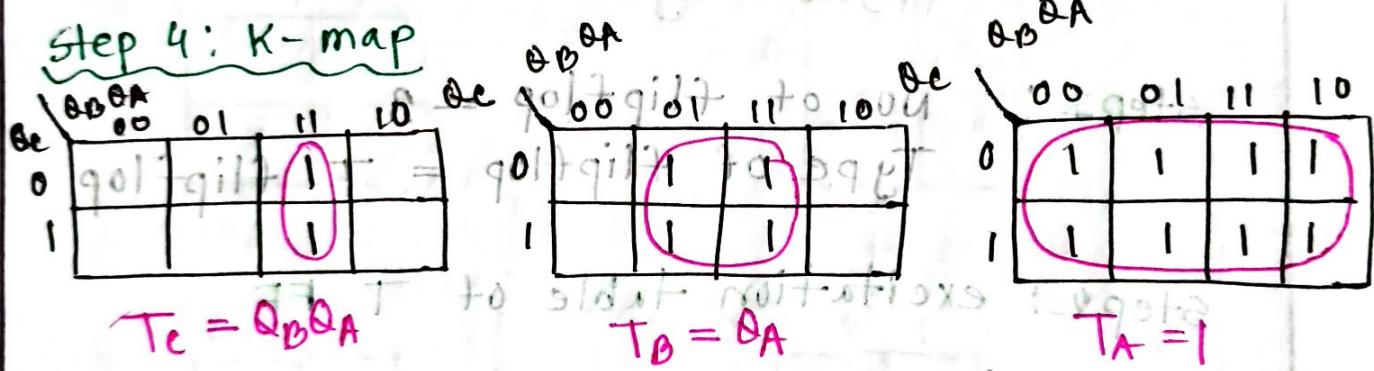
Max count,  $2^n - 1 = 2^3 - 1 = 7$



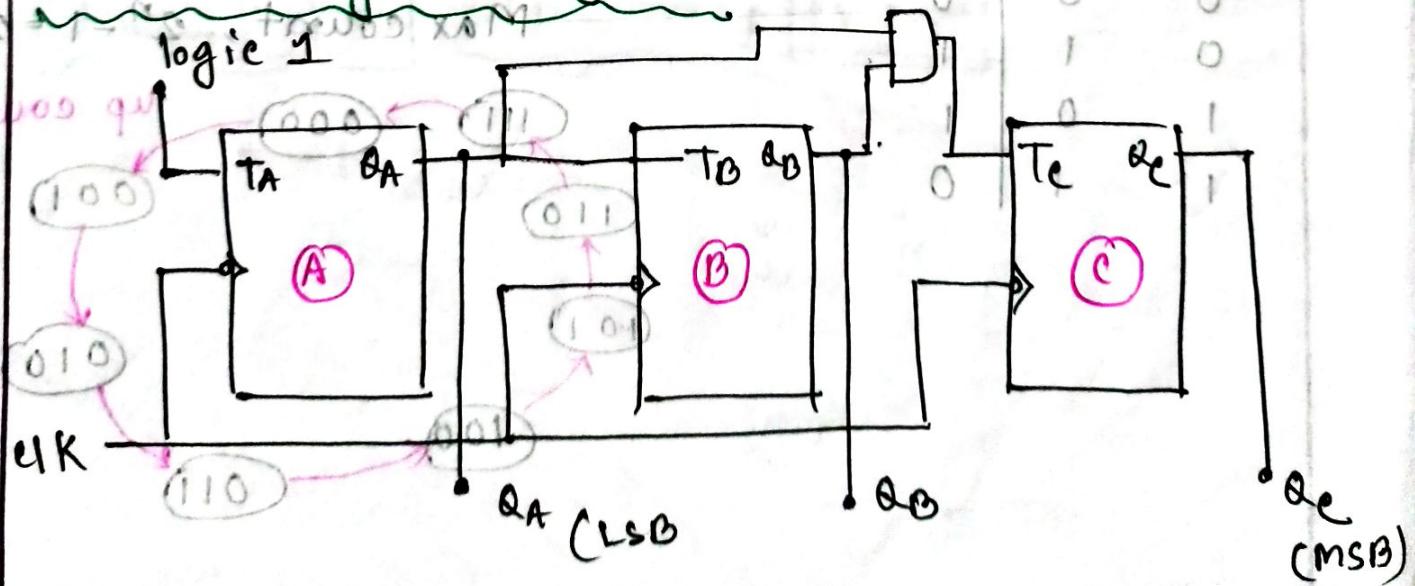
## circuit excitation table

$Q_c$	$Q_B$	$Q_A$	$Q_c^*$	$Q_B^*$	$Q_A^*$	$T_C$	$T_B$	$T_A$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	0	0	0	1	1
1	1	0	1	1	0	0	0	1
1	1	1	0	0	0	1	1	1

## Step 4: K-map



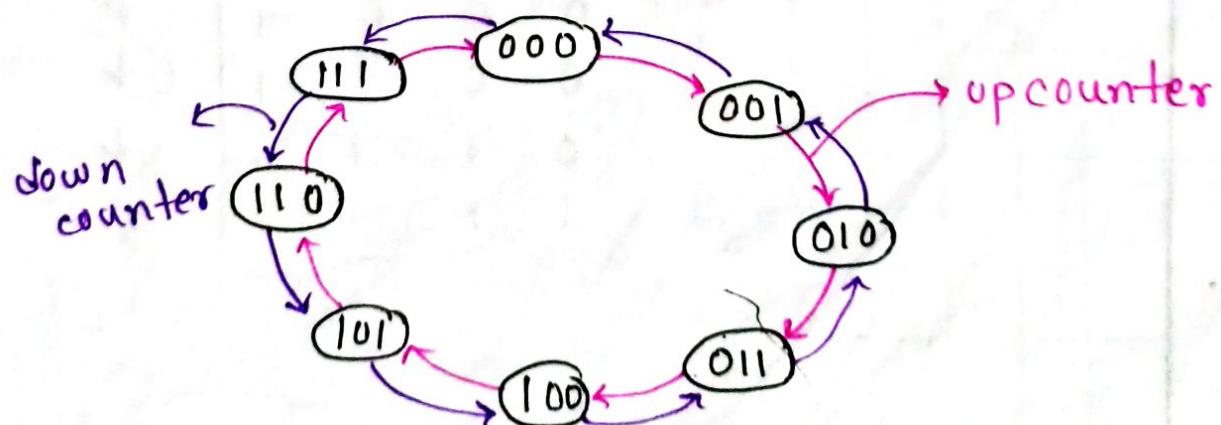
## Step 5: logic diagram



next state  
from excitation table

### 3 bit Up/down Synchronous Counter

M	$\Delta_e$	$Q_B$	$Q_A$	$Q_B^*$	$Q_B^*$	$Q_A^*$	$\Delta_e$	$T_B$	$T_A$	M=0 up counter
0	0	0	0	0	1	0	0	0	1	M=1 down counter
0	0	0	1	0	1	0	0	1	1	up counter
0	0	0	1	0	1	1	0	0	1	up counter
0	0	1	1	1	0	0	1	1	1	up counter
0	1	0	0	1	0	1	1	0	1	up counter
0	1	0	1	1	1	0	0	1	1	up counter
0	1	0	1	1	1	1	0	0	1	up counter
0	1	1	1	1	1	1	0	0	1	up counter
1	1	1	1	1	1	1	1	1	1	up counter
1	0	0	0	1	0	1	1	0	1	down counter
1	0	0	1	0	0	0	0	0	1	down counter
1	0	1	0	0	0	1	0	1	1	down counter
1	0	1	1	0	1	0	1	0	1	down counter
1	1	0	0	0	1	1	1	1	1	down counter
1	1	0	0	1	1	1	1	1	1	down counter
1	1	1	0	1	1	0	0	0	1	down counter
1	1	1	1	1	1	0	0	0	1	down counter



		BB QA	00	01	11	10
		QD QA	00	01	11	10
moe		00	0	1	1	1
00		0	0	1	1	1
01		0	1	0	1	1
11		1	1	0	0	1
10		1	0	1	0	1

		BB QA	00	01	11	10
		QD QA	00	01	11	10
moe		00	0	1	1	1
00		0	0	1	1	1
01		0	1	0	1	1
11		1	1	0	0	1
10		1	0	1	0	1

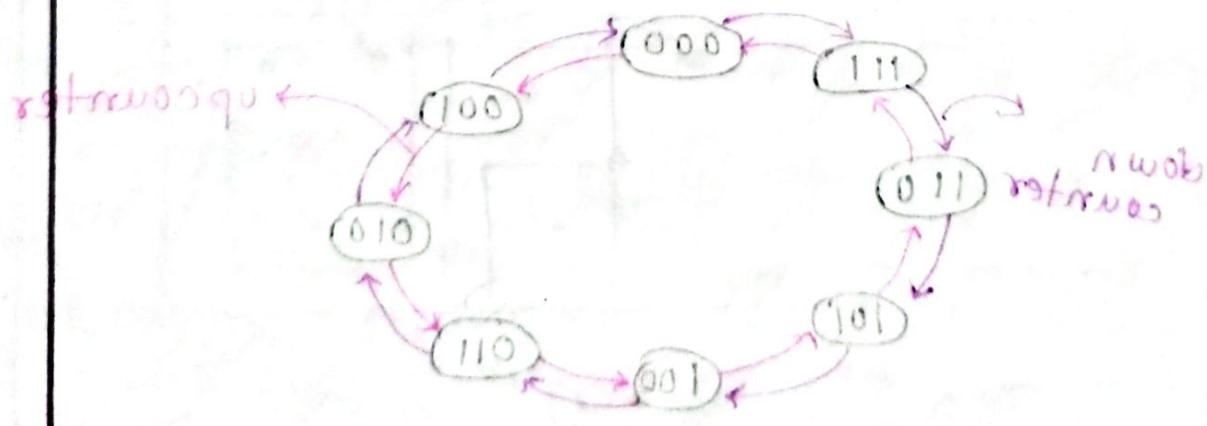
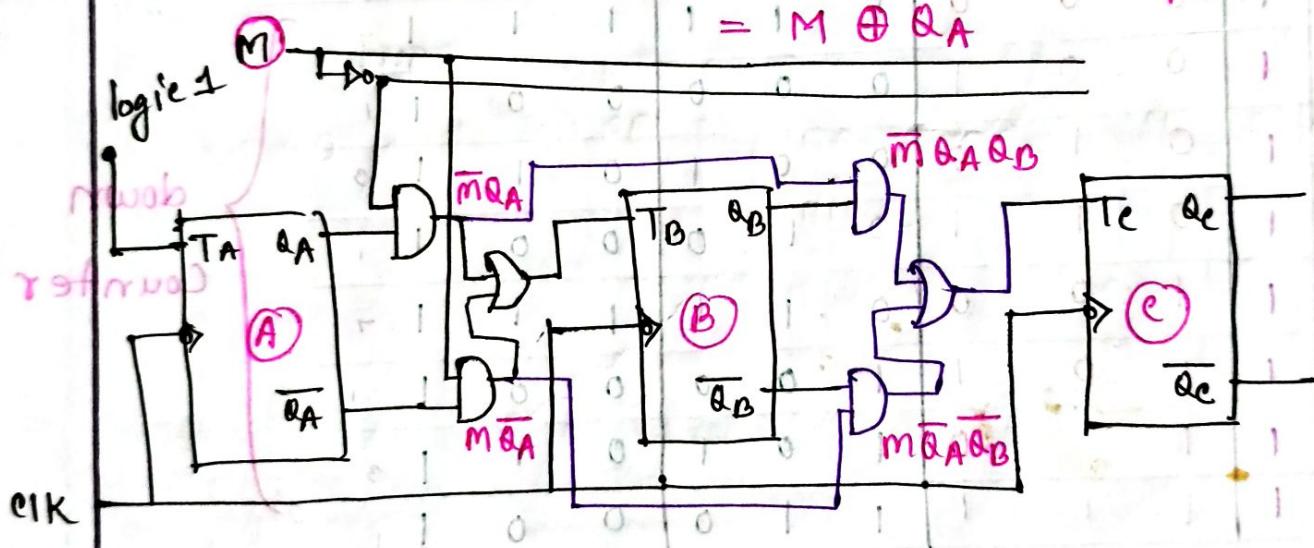
		BB QA	00	01	11	10
		QD QA	00	01	11	10
moe		00	0	1	1	1
00		0	0	1	1	1
01		0	1	0	1	1
11		1	1	0	0	1
10		1	0	1	0	1

$$T_C = \bar{M} \& B \& A + M \bar{B} \& \bar{A}$$

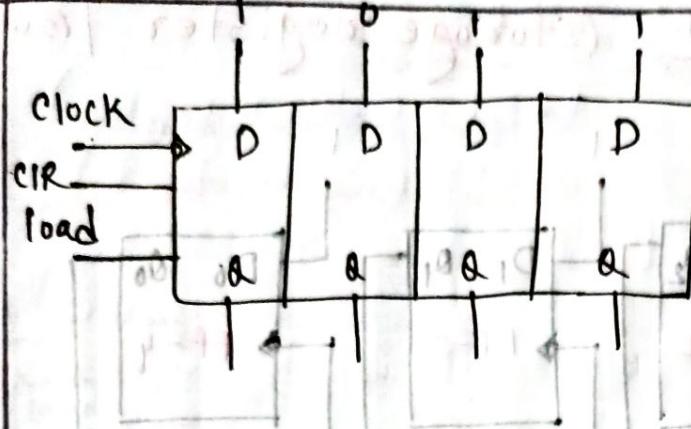
$$T_B = \bar{M} Q_A + \bar{B} A \bar{M}$$

$$= M \oplus Q_A$$

$$T_A = 1$$



## Introduction of Register



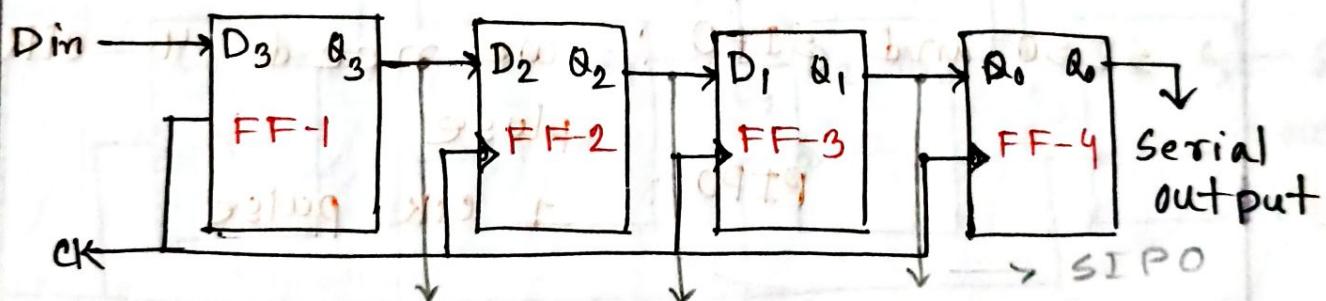
Synch : clock ↑ and load ↑  
Asynch : only load ↑

### Shift Register

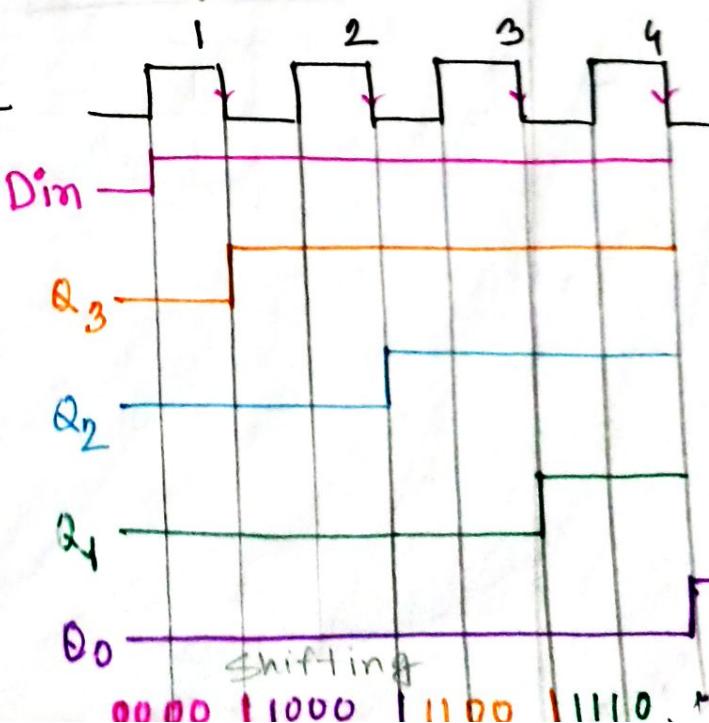
- (SISO) [shift right]

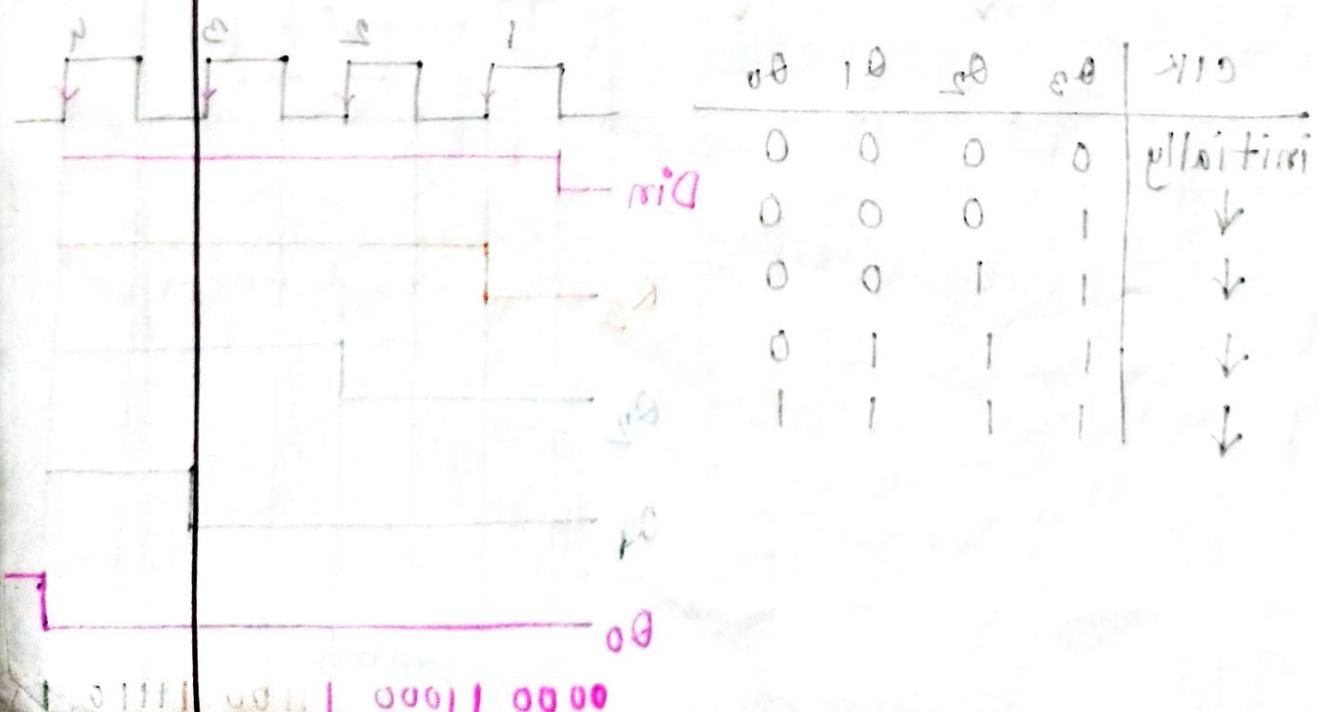
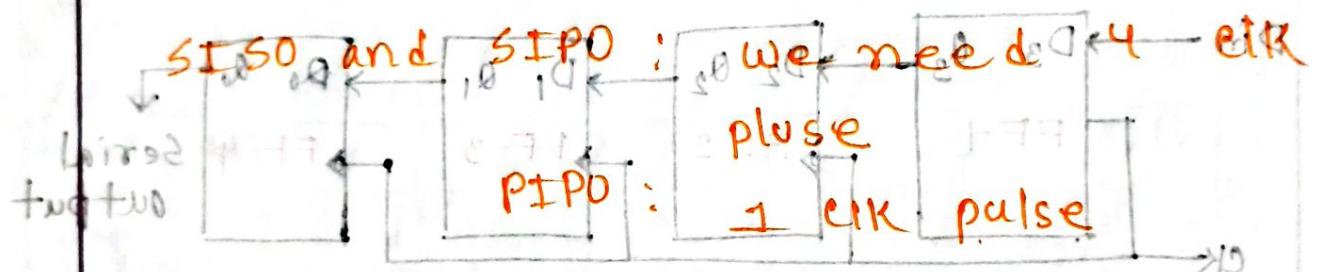
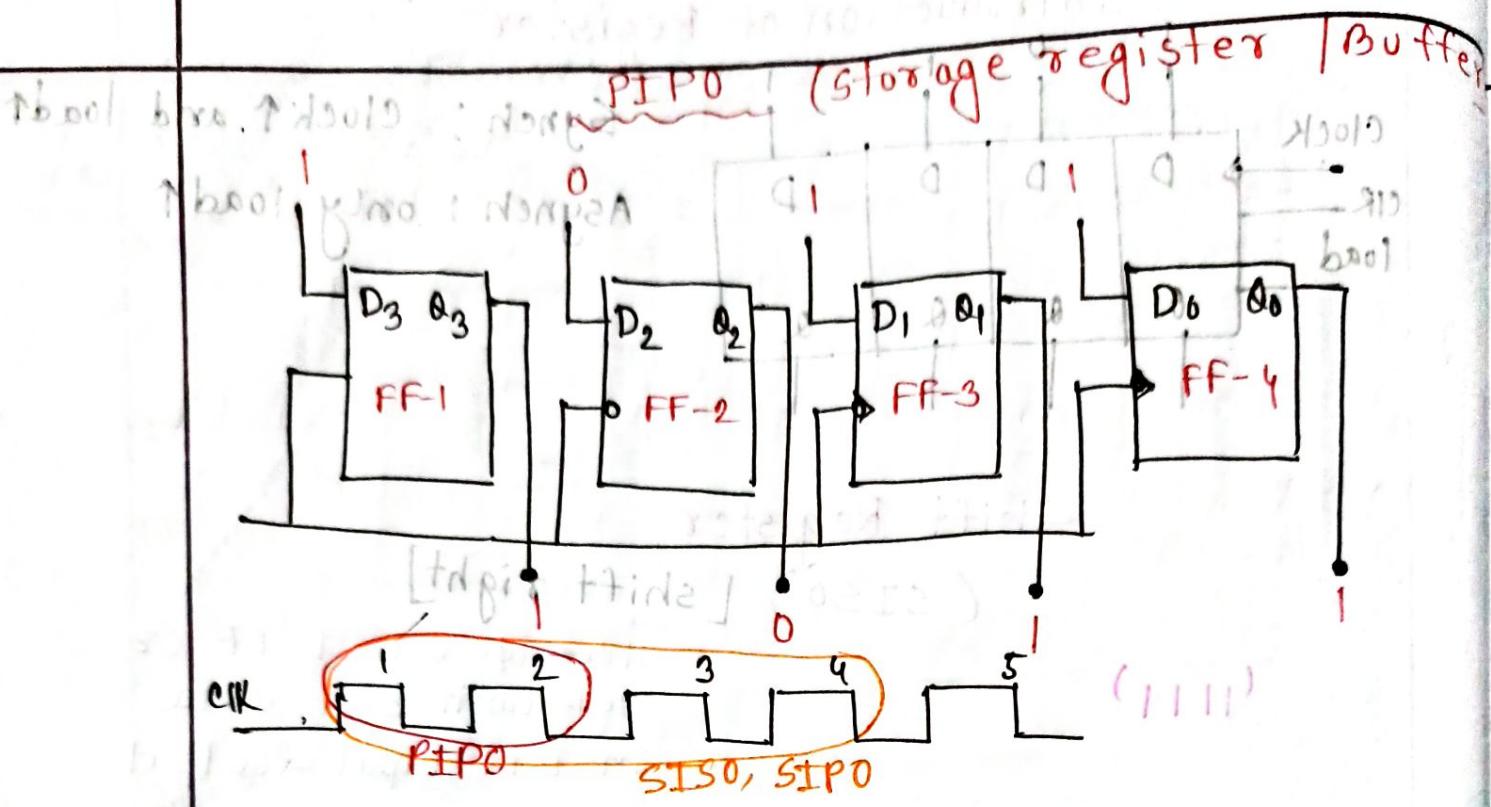
(1111)

Here we use D FF c2  
we know it starts data  
in D FF input = output



C1K	$Q_3$	$Q_2$	$Q_1$	$Q_0$
initially	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

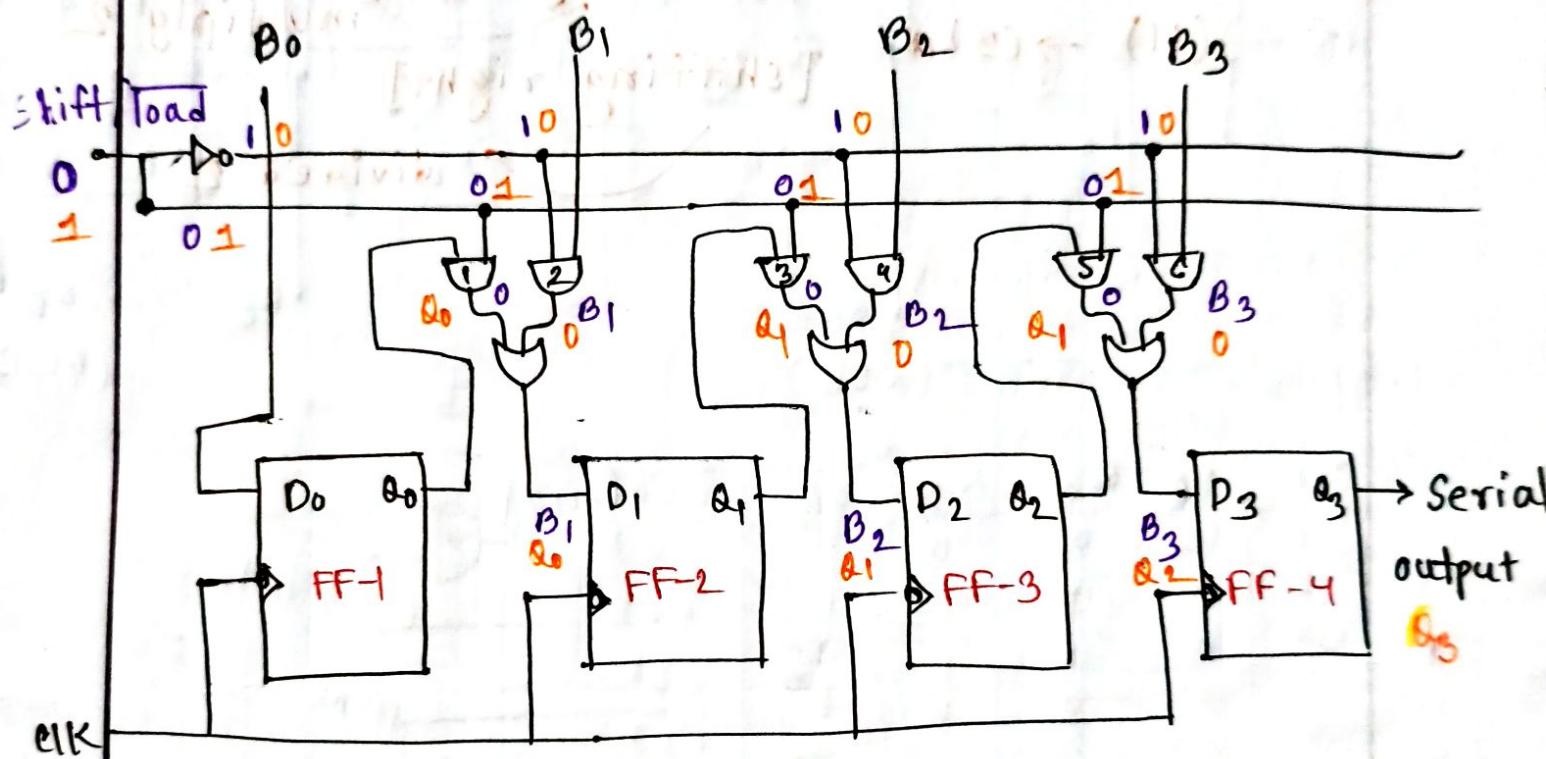




PISO

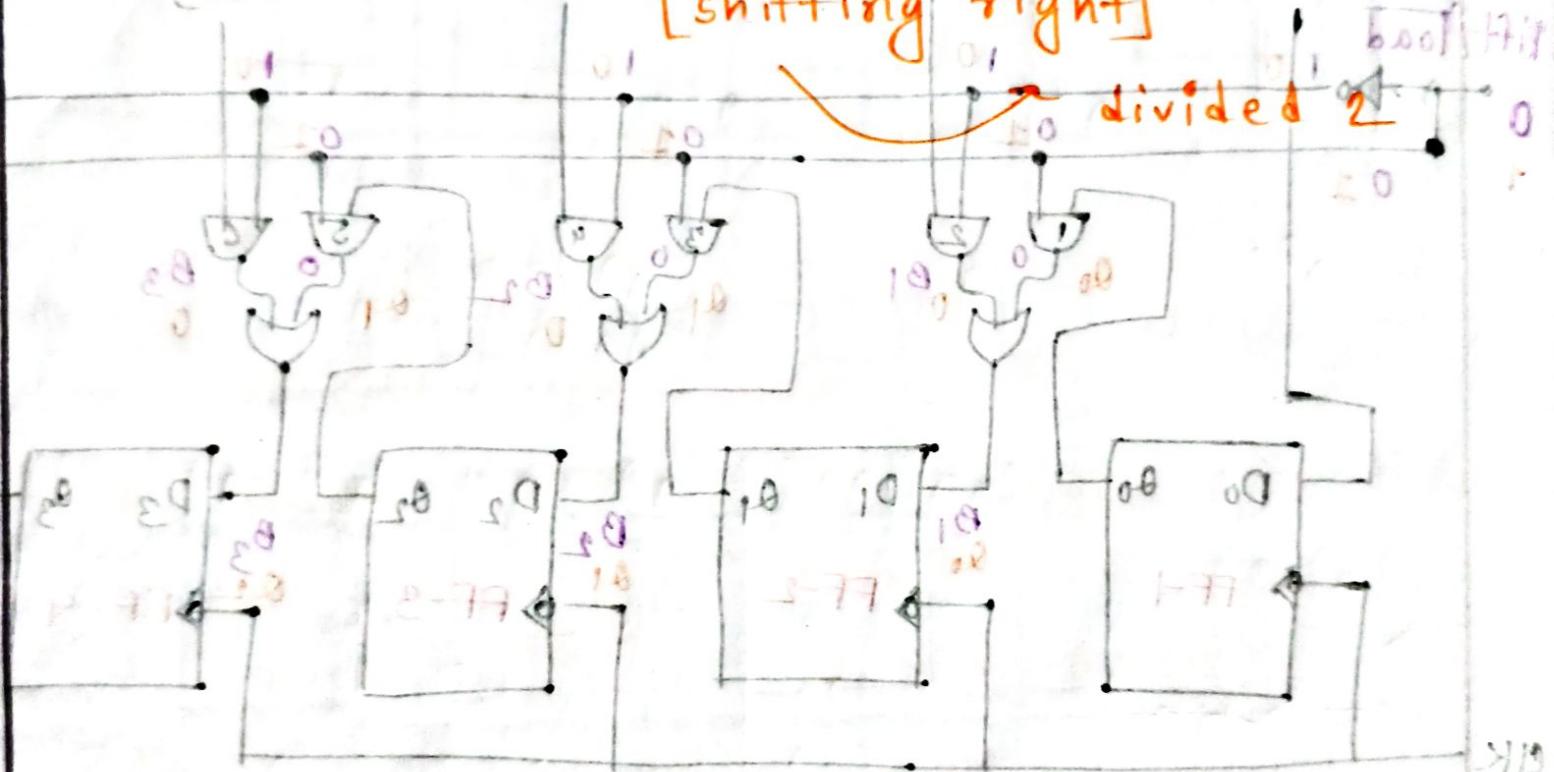
1. Load mode  $\Rightarrow 0$

2. Shift mode  $\Rightarrow 1$

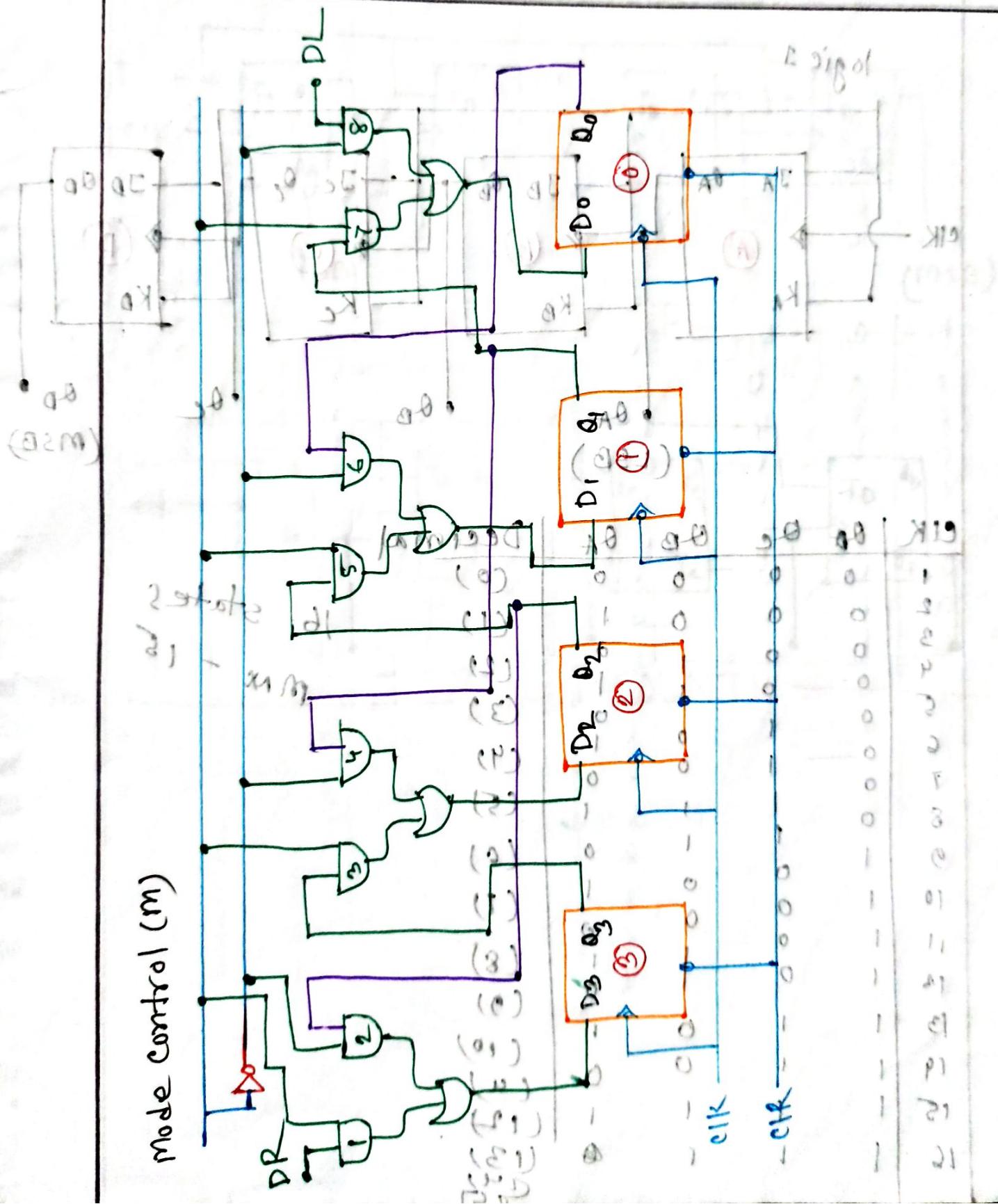


## Bidirectional Shift register

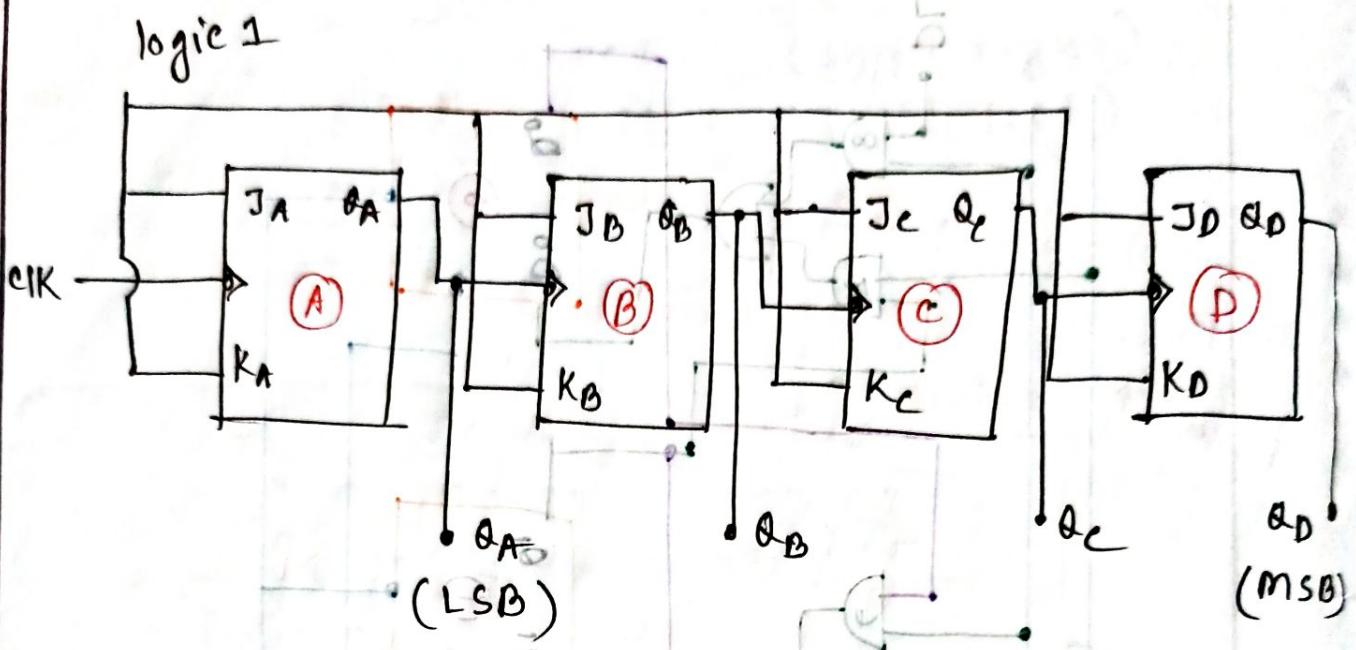
- \*  $(110)_2 \rightarrow (6)_10$  [shifting right] multiply 2
- $M = 1$  (shift right),  
 $M = 0$  (shift left)
- $(11)_2 \rightarrow (3)_10$  [shifting right]



(15) n/a



## 4-bit Asynchronous Up Counter (JK)



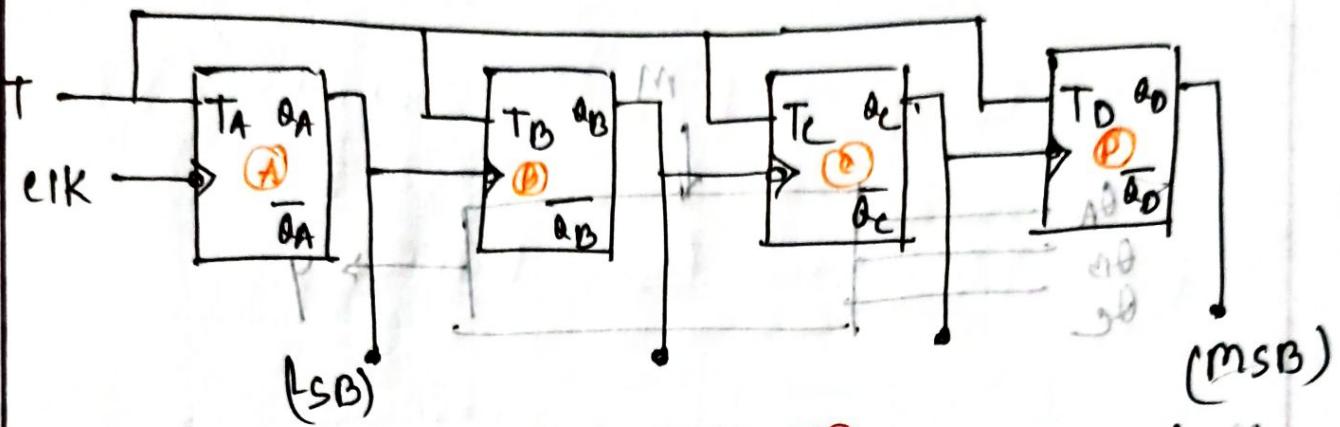
CLK	$Q_A$	$Q_C$	$Q_B$	$Q_A$	Decimal
1	0	0	0	0	(0)
2	0	0	0	1	(1)
3	0	0	1	0	(2)
4	0	0	1	1	(3)
5	0	0	1	0	(4)
6	0	1	0	1	(5)
7	0	1	0	0	(6)
8	0	1	1	1	(7)
9	1	0	1	0	(8)
10	1	0	0	1	(9)
11	1	0	0	0	(10)
12	1	0	1	0	(11)
13	1	1	0	1	(12)
14	1	1	0	0	(13)
15	1	1	1	1	(14)
16	1	1	1	0	(15)

16 states

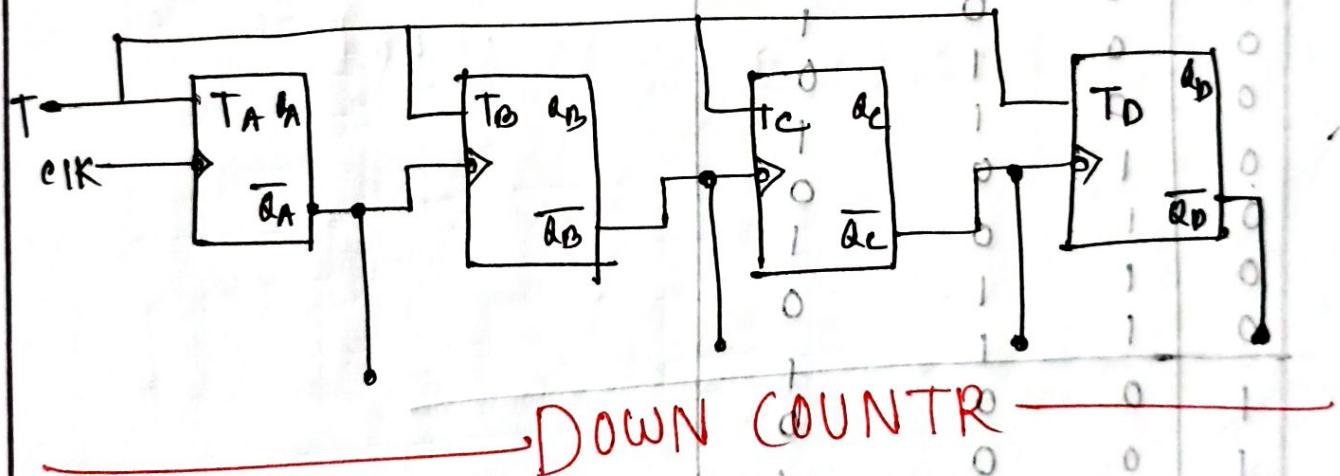
$2^4 = 16$

long count (w)

## 4 Bit Up down Asynchronous (T)

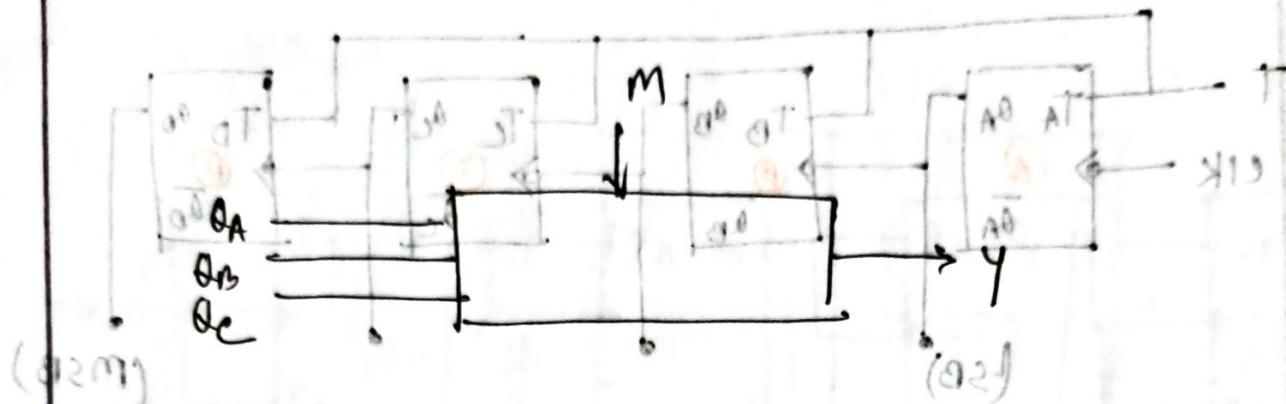


UP COUNTER



DOWN COUNTER

## 4 Bit Up / Down Ripple Counter



M	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Y
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	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	