

# Number System

- ① Binary  $(\quad)_2$
- ② Octal  $(\quad)_8$
- ③ Decimal  $(\quad)_{10}$
- ④ Hexadecimal  $(\quad)_{16}$

- ① From decimal to others  $\rightarrow$  Binary  
 $\rightarrow$  Octal  
 $\rightarrow$  Hexadecimal

ⓐ  $(19.35)_{10} \rightarrow (\quad)_2$

2	19	/
2	9	1
2	8	1
2	4	0
2	2	0
	0	

2	19	/
2	9	1
2	4	1
2	2	0
2	1	0
	0	1

$$\begin{aligned} 0.35 \times 2 &= 0.70 = 0 \\ 0.70 \times 2 &= 1.40 = 1 \\ 1.40 \times 2 &= 0.80 = 0 \\ 0.80 \times 2 &= 1.60 = 1 \end{aligned}$$

$$\Rightarrow (10011.010)_2$$

ⓑ  $(19.35)_{10} \rightarrow (\quad)_8$

8	19	/
8	2	3
8	0	2

$$\begin{aligned} 0.35 \times 8 &= 2.8 = 2 \\ 0.80 \times 8 &= 6.4 = 6 \\ 0.40 \times 8 &= 3.2 = 3 \\ 0.20 \times 8 &= 1.60 = 1 \\ 0.60 \times 8 &= 4.80 = 4 \end{aligned}$$

$$\Rightarrow (23.2631)_8$$

$$\textcircled{C} \quad (19.35)_{10} = (?)_{16}$$

$$\begin{array}{r} 16 \\ \hline 16 \Big| 1 \quad 3 \\ \hline 0 \quad 1 \end{array}$$

$$0.35 \times 16 = 5.6 = 5$$

$$0.60 \times 16 = 9.6 = 9$$

$$0.60 \times 16 = 9.6 = 9$$

$$\Rightarrow (13.599)_{16}$$

\textcircled{D} From Octal to Decimal

$$\textcircled{E} \quad (1101.01)_2 = (?)_{10}$$

$$= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 \cdot 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$= 1 + 0 + 4 + 8 \cdot .25$$

$$\Rightarrow (13.25)_{10}$$

$$\frac{3}{16} + \frac{1}{8} + \frac{1}{2}$$

$$\textcircled{F} \quad (17.4)_8 = (?)_{10}$$

$$7 \times 8^0 + 1 \times 8^1 \cdot 8^{-1} \times 4$$

$$7 + 8 \cdot \frac{1}{8} \times 4$$

$$\Rightarrow (15.5)_{10}$$

$$\textcircled{G} \quad (A7.C)_{16} = (?)_{10}$$

$$7 \times 16^0 + 10 \times 16^1 \cdot 16^{-1} \times 12$$

$$7 + 160 \cdot .75$$

$$\Rightarrow (167.75)_{10}$$

Binary  $\leftrightarrow$  Octal

Octal  $\leftrightarrow$  Hexadecimal

Binary  $\leftrightarrow$  Hexadecimal

Ⓐ  $(11011.110)_2 = (?)_8$

$\begin{array}{r} 011011.110 \\ \fbox{3} \quad \fbox{3} \quad \fbox{6} \end{array}$

$\Rightarrow (33.6)_8$

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

$\begin{array}{r} 000 \\ 8,2,1 \end{array}$

Ⓑ  $(33.6)_8 \rightarrow (?)_2$

$\Rightarrow (011011.110)_2$

Ⓒ  $(237.61)_8 = (?)_{16}$

$\begin{array}{r} 0000100111.11000100 \\ \fbox{9} \quad \fbox{15} \quad \fbox{12} \quad \fbox{4} \end{array}$

$\Rightarrow (9F.C4)_{16}$

Ⓓ  $(9F.C4)_{16} \rightarrow (?)_8$

$\begin{array}{r} 8421 \\ 0000 \end{array}$

$\begin{array}{r} 10011111.11000100 \\ \fbox{11} \quad \fbox{11} \quad \fbox{10} \end{array}$

$\Rightarrow (237.61)_8$

$$\textcircled{2} \quad (110100.0110)_2 \rightarrow (\ )_{16}$$

1	1
0	0
1	0
0	1

$\underbrace{0 \ 1 \ 1 \ 0}_{6} \quad \underbrace{1 \ 0 \ 1 \ 0}_{A} \cdot \underbrace{0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0}_{6 \ 8} \quad AB$

$$\Rightarrow (6A \cdot 68)_{16}$$

Binary To Gray Code & Vice versa

↓  
010111

↓  
011100

XOR Truth table

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

$$(11001)_B = (\ )_6$$

↓  
10101

$$\Rightarrow (10101)_6$$

$$(10101)_6 = (\ )_B$$

↓  
10110

↓  
1

$$\Rightarrow (11001)_B$$

Min Term: Given by SOP {sum of products}

$$\text{i.e } f(A, B) \rightarrow AB + \bar{B}$$

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

SSOP ~~canon~~ form of SOP  $\rightarrow A\bar{B} + \bar{A}B$

$\times A\bar{B} + \bar{B}$  {not canonically}

Every term of SSOP is called min term.

Decimal	A	B	C	Min Term
0	0	0	0	$\bar{A}\bar{B}\bar{C}$
1	0	0	1	$\bar{A}\bar{B}C$
2	0	1	0	$\bar{A}B\bar{C}$
3	0	1	1	$\bar{A}BC$
4	1	0	0	$A\bar{B}\bar{C}$
5	1	0	1	$A\bar{B}C$
6	1	1	0	$AB\bar{C}$
7	1	1	1	$ABC$

Min terms are denoted  
as 1 & further  
used in Kmaps

$$\sum_m (0, 1, 2)$$

$$T_M (0, 1, 2) \quad (\bar{A} + \bar{B} + C)$$

$$Y = \sum_m (0, 5, 6)$$

$$= m_0 + m_5 + m_6 = \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC$$

SOP Canonical form -

$$\textcircled{1} \quad Y(A, B) = A + B$$

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

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

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

$$\textcircled{2} \quad Y(A, B, C, D) = AB + ACD$$

$$= AB(C + \bar{C})(D + \bar{D}) + ACD(C + \bar{C})$$

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

$$= ABCD + ABC\bar{D} + A\bar{B}CD + ABC\bar{D} + A\bar{B}CD$$

$$= ABCD + ABC\bar{D} + A\bar{B}CD + A\bar{B}\bar{C}D + A\bar{B}CD$$

Maxterm - every term of gpos term is called Maxterm.

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

Decimal	A	B	C	Max Term	
0	0	0	0	$A + B + C$	$M_0$
0	0	0	1	$A + B + \bar{C}$	$M_1$
0	1	0	0	$A + \bar{B} + C$	$M_2$
0	1	0	1	$A + \bar{B} + \bar{C}$	$M_3$
1	0	0	0	$\bar{A} + B + C$	$M_4$
1	0	0	1	$\bar{A} + B + \bar{C}$	$M_5$
1	1	0	0	$\bar{A} + \bar{B} + C$	$M_6$
1	1	0	1	$\bar{A} + \bar{B} + \bar{C}$	$M_7$

list opposite  
to min term

# Canonical Form POS

$$\begin{aligned}
 Y(ABC) &= (A + \bar{B})(B + C)(A + \bar{C}) \\
 &= (A + \bar{B} + C\bar{C})(B + C + A\bar{A})(A + \bar{C} + B\bar{B}) \\
 &= A + \bar{B} + C \quad A + \bar{B} + \bar{C} \quad B + C + A \quad B + C + \bar{A} \quad A + \bar{C} + B \quad A + \bar{C} + \bar{B} \\
 &= (A + \bar{B} + C)(A + \bar{B} + \bar{C})(B + C + \bar{A})(B + C + \bar{A})(A + \bar{C} + B)(A + \bar{C} + \bar{B})
 \end{aligned}$$

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

① Canonical SOP

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

$$\begin{aligned}
 &A(B + \bar{B})(C + \bar{C}) + \bar{B}C(A + \bar{A}) \\
 \Rightarrow &(AB + A\bar{B})(C + \bar{C}) + \bar{B}CA + \bar{B}C\bar{A} \\
 \Rightarrow &\begin{matrix} ABC \\ 111 \end{matrix} + \begin{matrix} AB\bar{C} \\ 110 \end{matrix} + \begin{matrix} A\bar{B}C \\ 101 \end{matrix} + \begin{matrix} A\bar{B}\bar{C} \\ 100 \end{matrix} + \begin{matrix} \bar{B}CA \\ 1001 \end{matrix} + \begin{matrix} \bar{B}C\bar{A} \\ 001 \end{matrix} \\
 &\quad 7 \quad 6 \quad 5 \quad 4 \quad 1 \\
 \Rightarrow &\sum_m (1, 4, 5, 6, 7)
 \end{aligned}$$

②  $Y = A + \bar{B}C$  { This is SOP }

We have converted it into POS

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

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

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

$$\begin{aligned}
 &= (\bar{A} + B + C)(\bar{A} + B + \bar{C}) * (\bar{A} + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C}) \\
 &= (A + B + \bar{C})(\bar{A} + B + \bar{C})
 \end{aligned}$$

$$= \pi(0, 2, 3)$$

Principle of duality -  $0 \leftrightarrow 1$   
 $+ \leftrightarrow \cdot$

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

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

K-map:

2 variable

		A	B
		0	1
B	0	00	10
	1	01	11

A	B
0	0
0	1
1	0
1	1

3 variable

		AB	C		
		00	01	11	10
C	0	000	010	110	100
	1	001	011	111	101

		AB	C		
		0	2	6	4
C	0	0	2	6	4
	1	3	7	11	5

4 variable

		AB'	CD		
		00	01	11	10
CD	00	0000	0100	1100	1000
	01	0001	0101	1101	1001
CD	11	0011	0111	1111	1011
	10	0010	0110	1110	1010

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

D	D	D	D
D	1	D	D
D	D	1	D
D	D	D	1

$$y(A, B, C, D) = \sum_m (1, 3, 5, 7, 8, 9, 12, 13)$$

	AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
CD	00	01	11	10	
CD	00	0	1	12	18
CD	01	1	13	1	9
CD	11	2	1	15	11
CD	10	2	6	14	10

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

$$Q_2 \quad y = ABCD + \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C} + AB$$

$$\sum_m (8, 10, 11, 12, 13, 14, 15)$$

	AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
CD	00	01	11	10	
CD	00	0	1	12	18
CD	01	1	5	13	9
CD	11	3	7	15	11
CD	10	2	6	14	10

$$\rightarrow \bar{A}\bar{D} + AB + AC$$

$$Q_3 \quad Y = \sum_m (7, 9, 10, 11, 12, 13, 14, 15)$$

	AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
CD	00	01	11	10	
CD	00	0	1	12	8
CD	01	1	5	13	9
CD	11	3	7	15	11
CD	10	2	6	14	10

$$\rightarrow AB + AD + AC + BCD$$

$$Q_4 \quad Y = \pi (0, 1, 4, 5, 6, 8, 9, 12, 13, 14)$$

	AB	$\bar{A}+\bar{B}$	$\bar{A}+B$	$A+B$	$A+\bar{B}$
CD	000	0+1	1+1	1+0	
$\bar{C}+\bar{D}$	0+0	0	0	0	0
$\bar{C}+D$	0+1	0	0	1	0
$C+\bar{D}$	1+1	0	1	1	1
$C+\bar{D}$	1+0	1	0	1	0

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

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

sign offsite

Ques  $F(A, B, C, D) = \sum_m (0, 1, 2, 5, 8, 9, 10)$   
 $= \sum_m (3, 4, 6, 7, 11, 12, 13, 14, 15)$

		AB	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
		CD	00	01	11	10
CD	00	00	1	0	1	0
CD	01	1	1	1	1	0
CD	11	3	7	15	11	
CD	10	2	6	14	1	10

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

		AB	$A+\bar{B}$	$\bar{A}+\bar{B}$		
		CD	00	01	11	10
CD	00	0	0	0	0	8
CD	01	1	1	0	1	9
CD	11	0	0	0	1	1
CD	10	2	0	0	0	10

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

Ques Design 4 bit Binary to gray code converter.

4 bit binary

$B_1, B_2, B_3, B_4$

$g_1, g_2, g_3, g_4$

4-bit gray

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

$$G_1 = \{8, 9, 10, 11, 12, 13, 14, 15\}$$

$$G_2 = \sum_m \{4, 5, 6, 7, 8, 9, 10, 11\}$$

$$G_3 = \sum_m \{2, 3, 4, 5, 10, 11, 12, 13\}$$

$$G_4 = \sum_m \{1, 2, 5, 6, 9, 10, 13, 14\}$$

$$\begin{array}{l} AB + B\bar{A} \\ A \oplus B \end{array}$$

for  $G_1$

		$B_1B_2$	$\cancel{B_3B_4}$	$00$	$01$	$11$	$10$
		$B_3B_4$		00	01	11	10
$B_3B_4$	00	0	4	12	1	8	
	01	1	5	13	1	9	
	11	3	7	15	1	11	
	10	2	6	14	1	10	

$\Rightarrow$

$$G_1 = B_1$$

$$\begin{array}{l} AB + \bar{A}\bar{B} \\ A \odot B \end{array}$$

for  $G_2$

		$B_1B_2$	$\cancel{B_3B_4}$	$00$	$01$	$11$	$10$
		$B_3B_4$		00	01	11	10
$B_3B_4$	00	0	1	12	1	8	
	01	1	2	5	3	13	9
	11	3	1	7	15	2	11
	10	2	1	14	1	10	

$$\Rightarrow G_2 = \bar{B}_1 B_2 + B_1 \bar{B}_2$$

$$G_2 = B_1 \oplus B_2$$

for  $G_3$

		$B_1B_2$	$\cancel{B_3B_4}$	$00$	$01$	$11$	$10$
		$B_3B_4$		00	01	11	10
$\bar{B}_3\bar{B}_4$	00	0	1	4	12	1	8
	01	1	2	5	3	13	9
	11	3	1	7	15	2	11
	10	2	1	14	1	10	

$$G_3 = B_2 \bar{B}_3 + \bar{B}_2 B_3$$

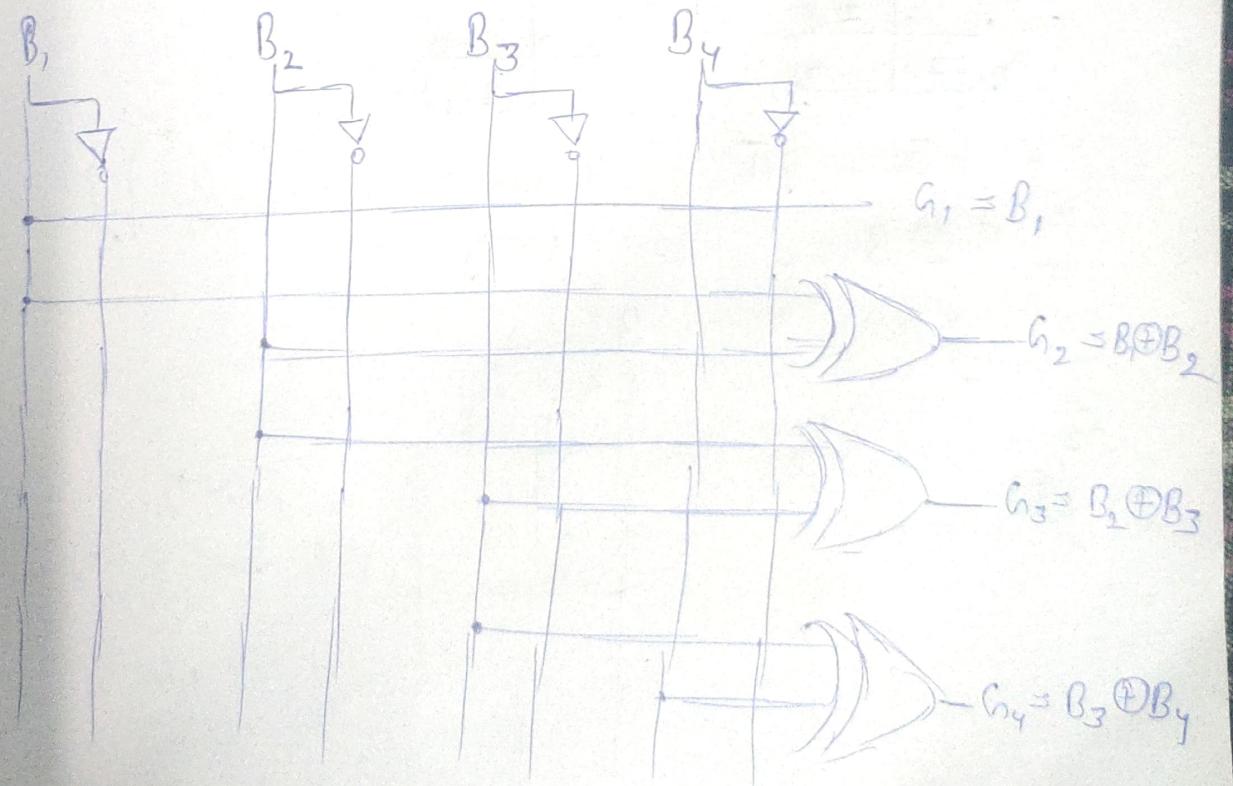
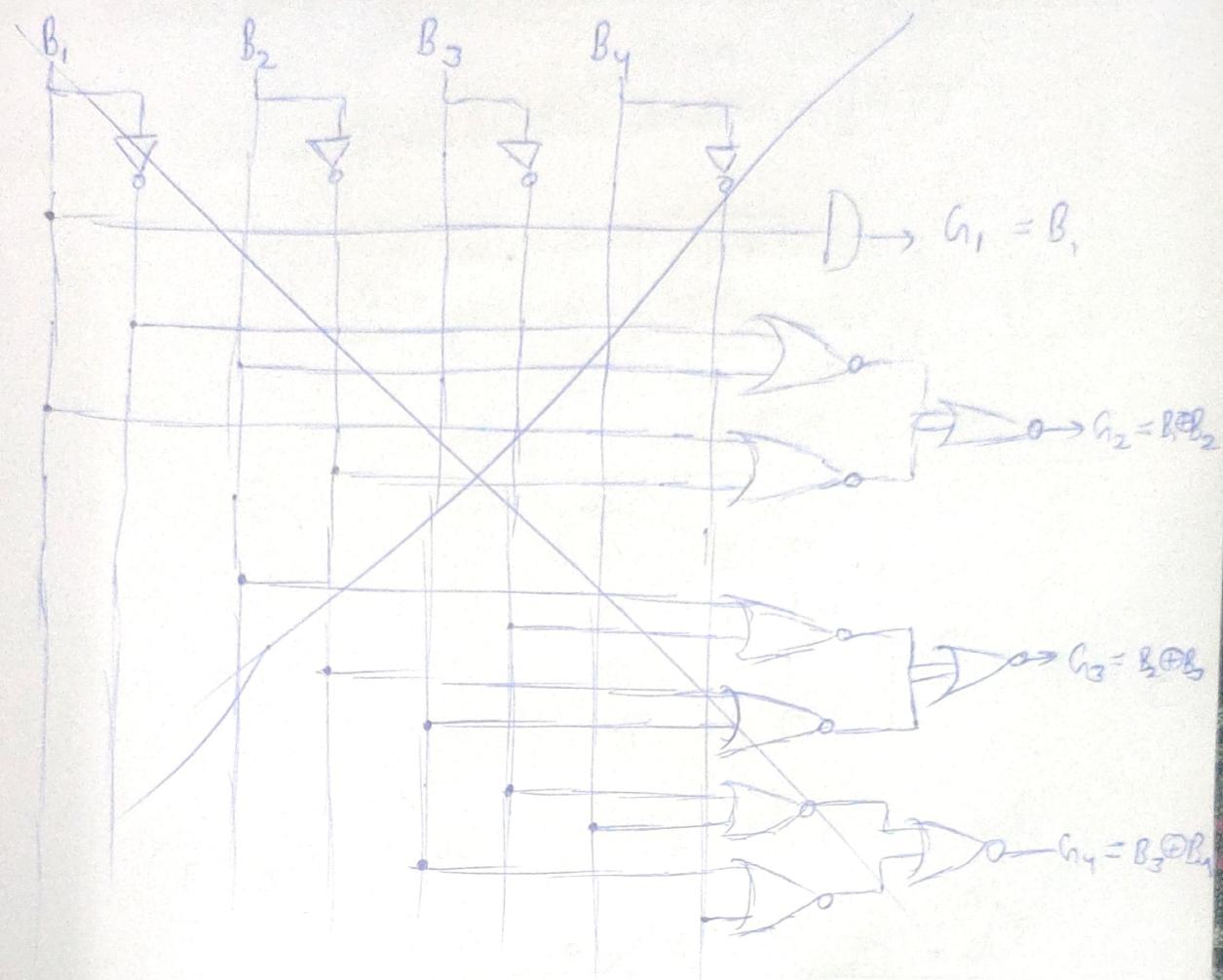
$$G_3 = B_2 \oplus B_3$$

for  $G_4$

		$B_1B_2$	$\cancel{B_3B_4}$	$00$	$01$	$11$	$10$
		$B_3B_4$		00	01	11	10
$B_3B_4$	00	0	4	12	1	8	
	01	1	2	5	3	13	9
	11	3	1	7	15	2	11
	10	2	1	14	1	10	

$$\Rightarrow G_4 = \bar{B}_3 B_4 + B_3 \bar{B}_4$$

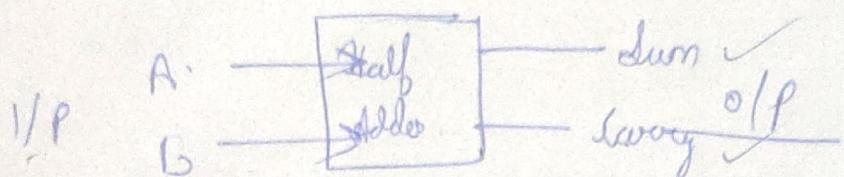
$$G_4 = B_3 \oplus B_4$$



Half Adder Combinational circuit which performs addition of numbers.

① Used in ALU ✓

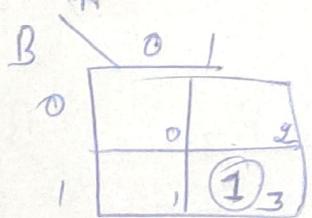
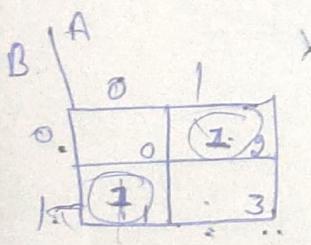
② Perform arithmetic addition of 2 one bit numbers.



Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

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

$$\text{Carry} = \sum m(3)$$



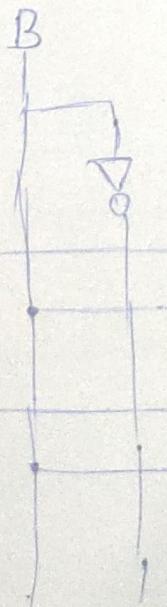
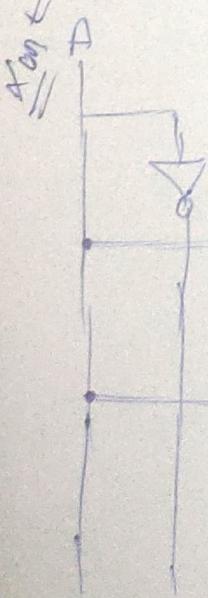
$$\text{Sum} = \bar{A}\bar{B} + \bar{B}\bar{A}$$

$$= A \oplus B$$

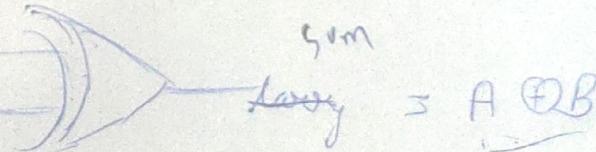
$$\text{Sum} = \overline{AB}$$

D

$$\begin{matrix} & \oplus \\ A & \oplus B \end{matrix}$$

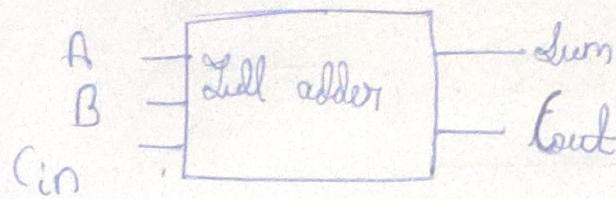


$$\text{Carry} = AB$$



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

Full Adder: ① combinational logic circuit that performs arithmetic sum of three inputs bits.



$$\text{sum} = \sum_{m=1}^7 1, 2, 4, 7$$

$$b_{out} = \sum_m \{ 3, 5, 6, 7 \}$$

$C_{in}$	$AB$	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	$AB$
$\bar{C}_{in}$	0	0	1	0	0
$\bar{C}_{in}$	0	1	1	0	1
$C_{in}$	1	1	0	1	1
$C_{in}$	1	0	0	0	0

$$\Rightarrow \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}^*C_{in} + A\bar{B}\bar{C}_{in}$$

$$= \bar{A}(\bar{B}C_{in} + B\bar{C}_{in}) + A(BC_{in} + \bar{B}\bar{C}_{in})$$

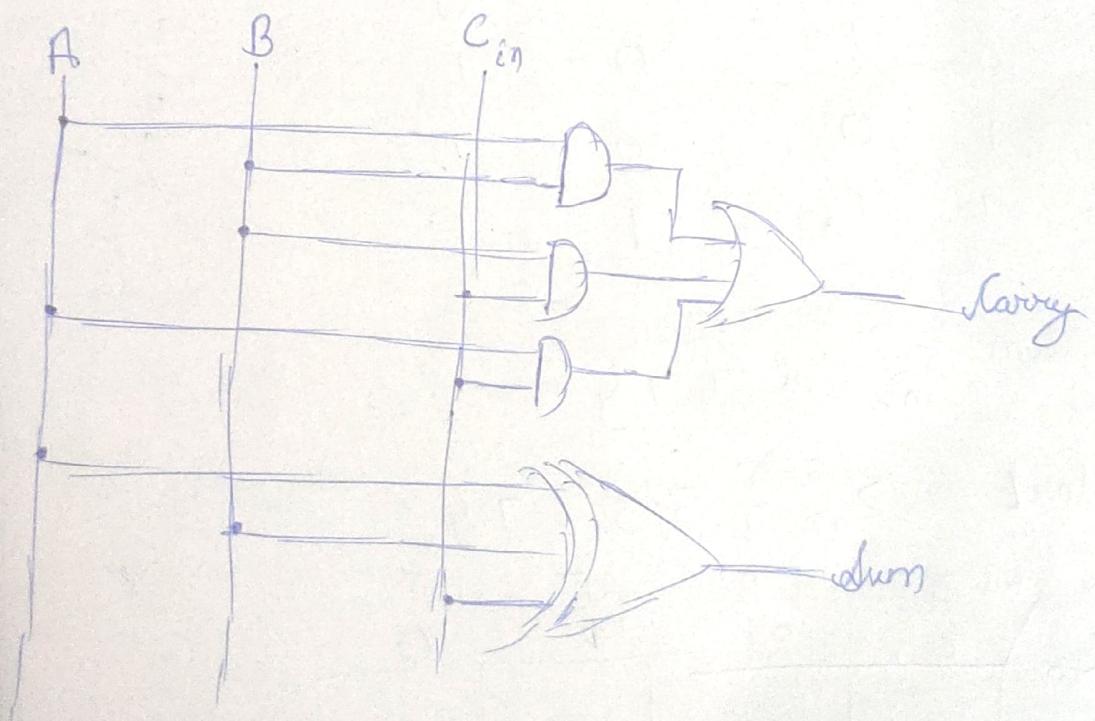
$$\text{xOR} = \bar{A}(B \oplus C_{in}) + A(B \odot C_{in}) \quad \text{xNOR}$$

$$= \bar{A}(B \oplus C_{in}) + A(\overline{B \oplus C_{in}})$$

Sum =  $A \oplus B \oplus C_{in}$

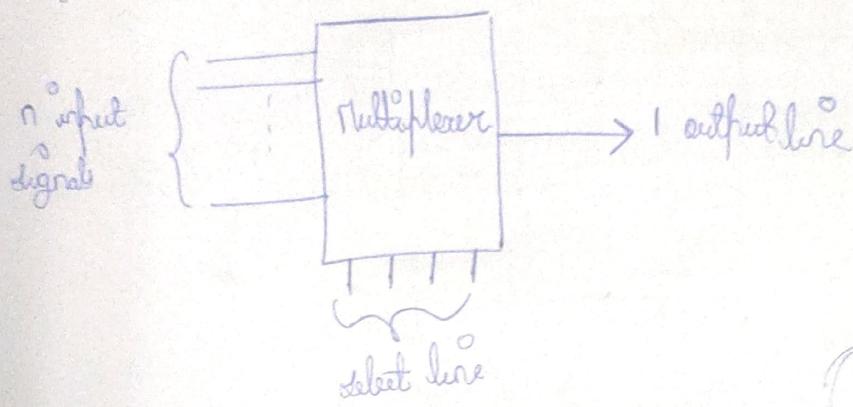
		AB	00	01	11	10
		$C_{in}$	0	2	6	4
$C_{in}$	0	0	1	1	0	1
	1	1	1	0	1	0

$$\Rightarrow AB + BC_{in} + A \oplus C_{in}$$

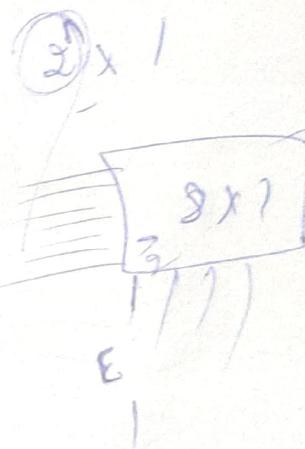
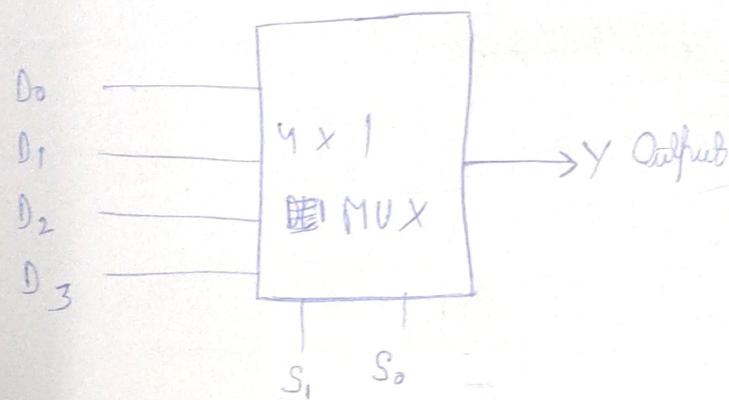


## Demultiplexer

Multiplexer : ① Combinational circuit  
 ② It has  $2^n$  input lines and single output line  
 { Many to one }



Here m select line,  
 $2^m$  input line  
 1 output line

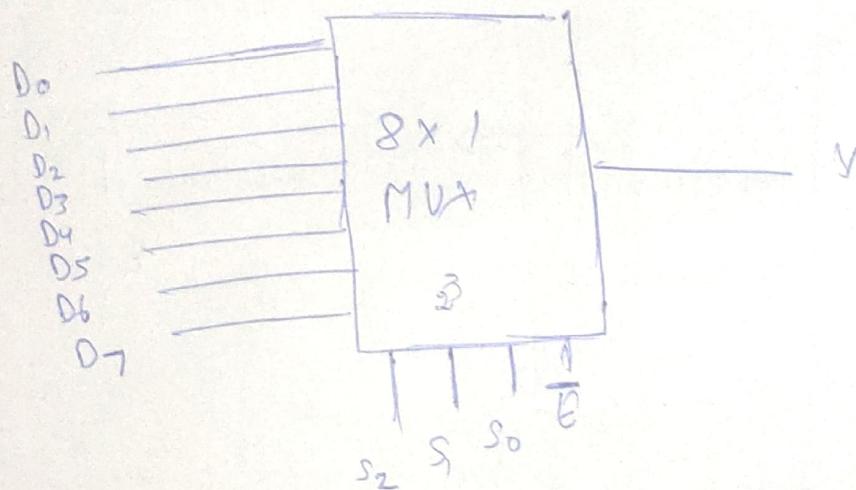
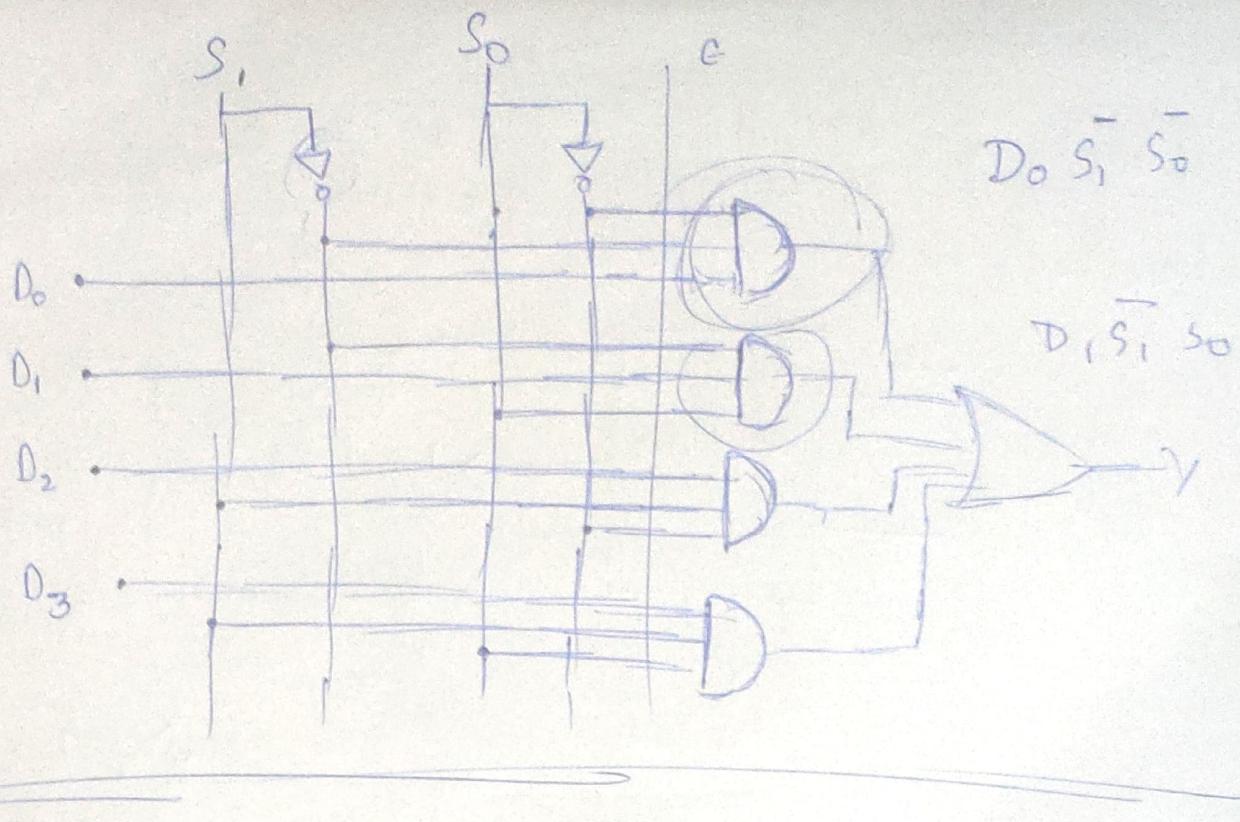


Data Input	Select Input	Output
D <sub>0</sub> (S <sub>1</sub> )	S <sub>0</sub>	Y
0	0	D <sub>0</sub>
0	1	D <sub>1</sub>
1	0	D <sub>2</sub>
1	1	D <sub>3</sub>

$$\begin{aligned} & \text{Data } D \\ & 2^n \rightarrow 1 \\ & \text{Select } S_1, S_0 \\ & \begin{array}{c|c|c} S_0 & S_1 & \text{Out} \\ \hline 0 & 0 & D_0 \\ 0 & 1 & D_1 \\ 1 & 0 & D_2 \\ 1 & 1 & D_3 \end{array} \end{aligned}$$

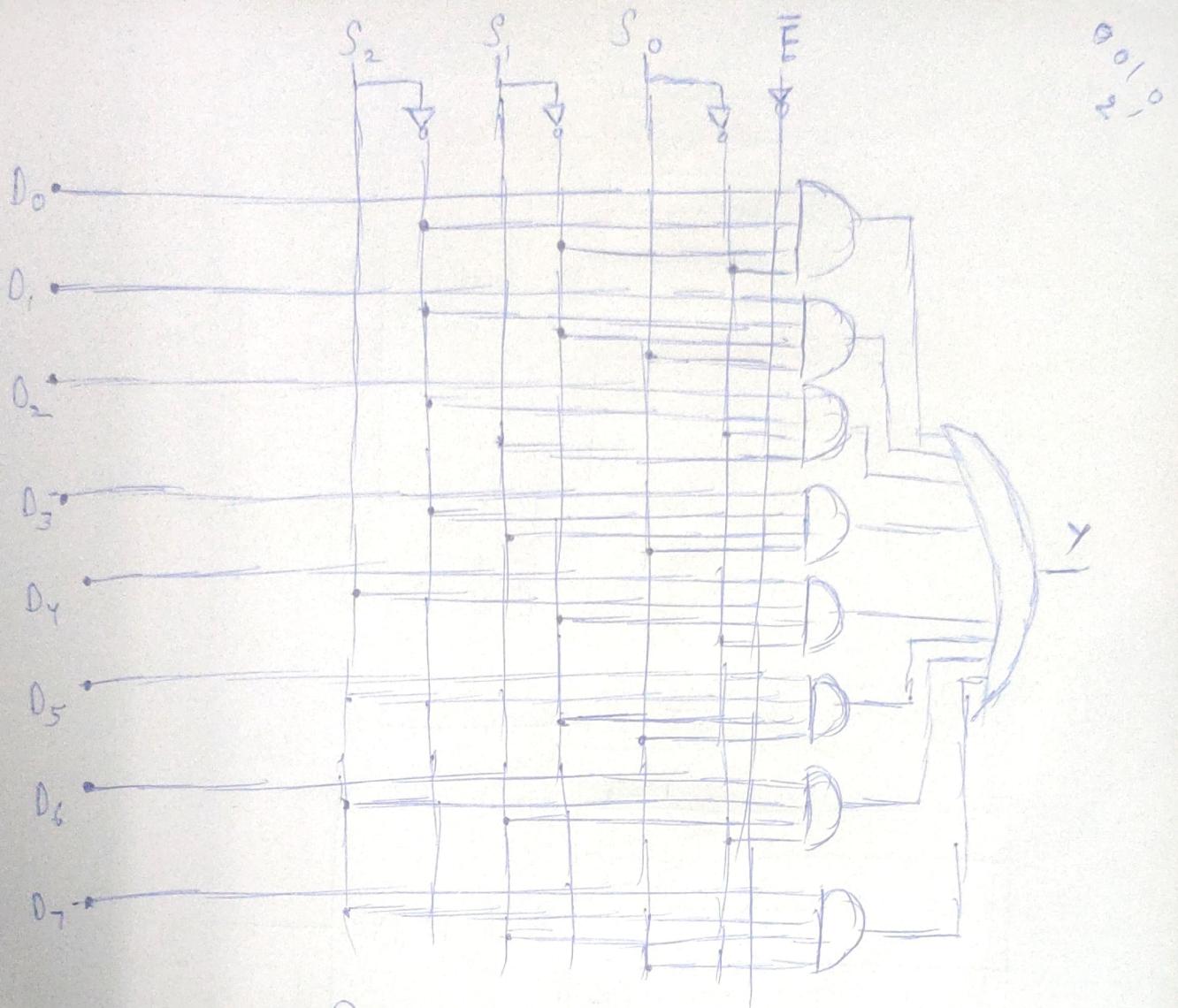
$$X = D_0 \bar{S}_1 \bar{S}_0 + D_1 \bar{S}_1 S_0 + D_2 S_1 \bar{S}_0 + D_3 S_1 S_0$$

$$\bar{S}_1 (D_0 \bar{S}_0 + D_1 S_0)$$



E	Data Input			Output $Y$
	$S_2$	$S_1$	$S_0$	
0	0	0	0	$D_0$
0	0	0	1	$D_1$
0	0	1	0	$D_2$
0	0	1	1	$D_3$
0	1	0	0	$D_4$
0	1	0	1	$D_5$
0	1	1	0	$D_6$
0	1	1	1	$D_7$

$$\begin{aligned}
 Y = & D_0 \bar{S}_2 \bar{S}_1 \bar{S}_0 + D_1 \bar{S}_2 \bar{S}_1 S_0 + \\
 & D_2 \bar{S}_2 S_1 \bar{S}_0 + D_3 \bar{S}_2 S_1 S_0 + \\
 & D_4 S_2 \bar{S}_1 \bar{S}_0 + D_5 \bar{S}_2 \bar{S}_1 \bar{S}_0 + \\
 & D_6 S_2 S_1 \bar{S}_0 + D_7 S_2 S_1 S_0
 \end{aligned}$$



### K-Map Designing Plan

Q1  $P(A, B, C, D) = \sum m(0, 1, 3, 5, 9, 13, 15)$

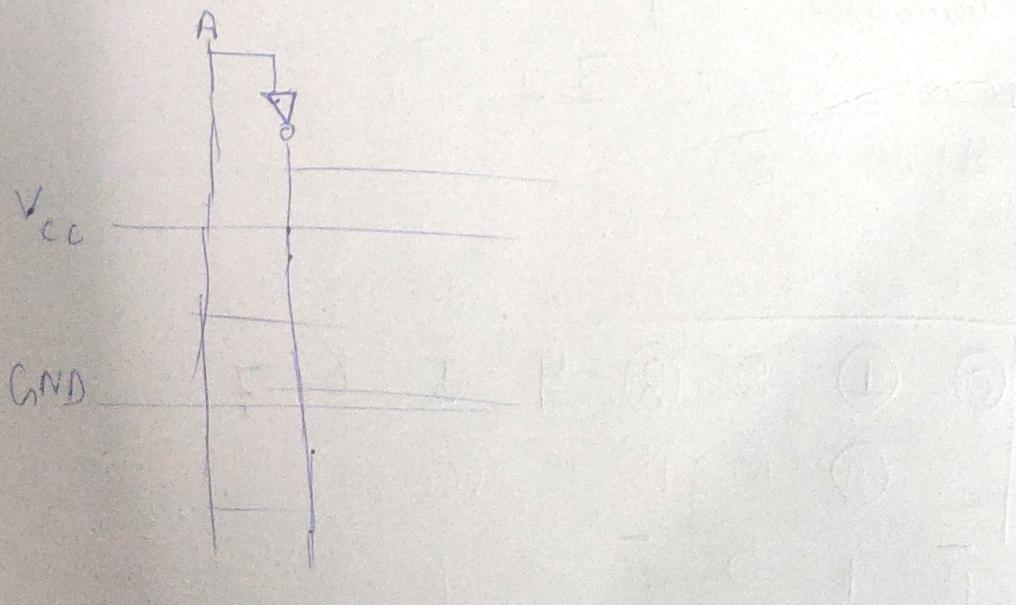
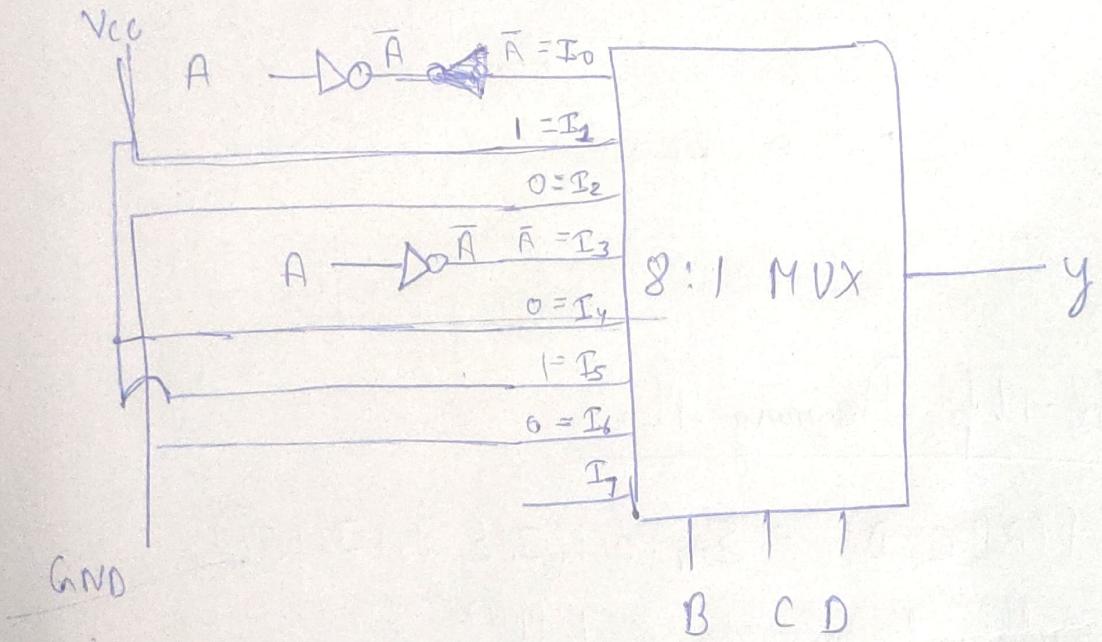
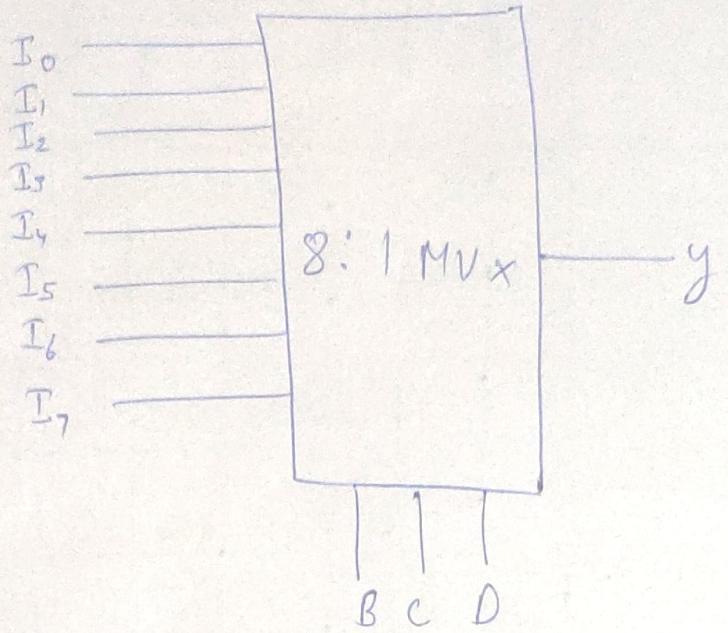
Implement it using 8:1 MUX?

No. of variables = 4

Deleted lines =  $\{2^3\}$

Inputs = 8

		BCD	000	001	010	011	100	101	110	111
		A	0	1	2	3	4	5	6	7
		$\bar{A}$	8	9	10	11	12	13	14	15
		A	1	0	1	$\bar{A}$	0	1	0	$\bar{A}$



$$\text{Ques} \quad f(A, B, C, D) = \sum_m(0, 1, 3, 5, 9, 13, 15)$$

Complement using 4:1 MUX

No of variables = 4

No of select line = 2 ( $2^2$ )

No of output = 4

ADD D

OR

$\Sigma$

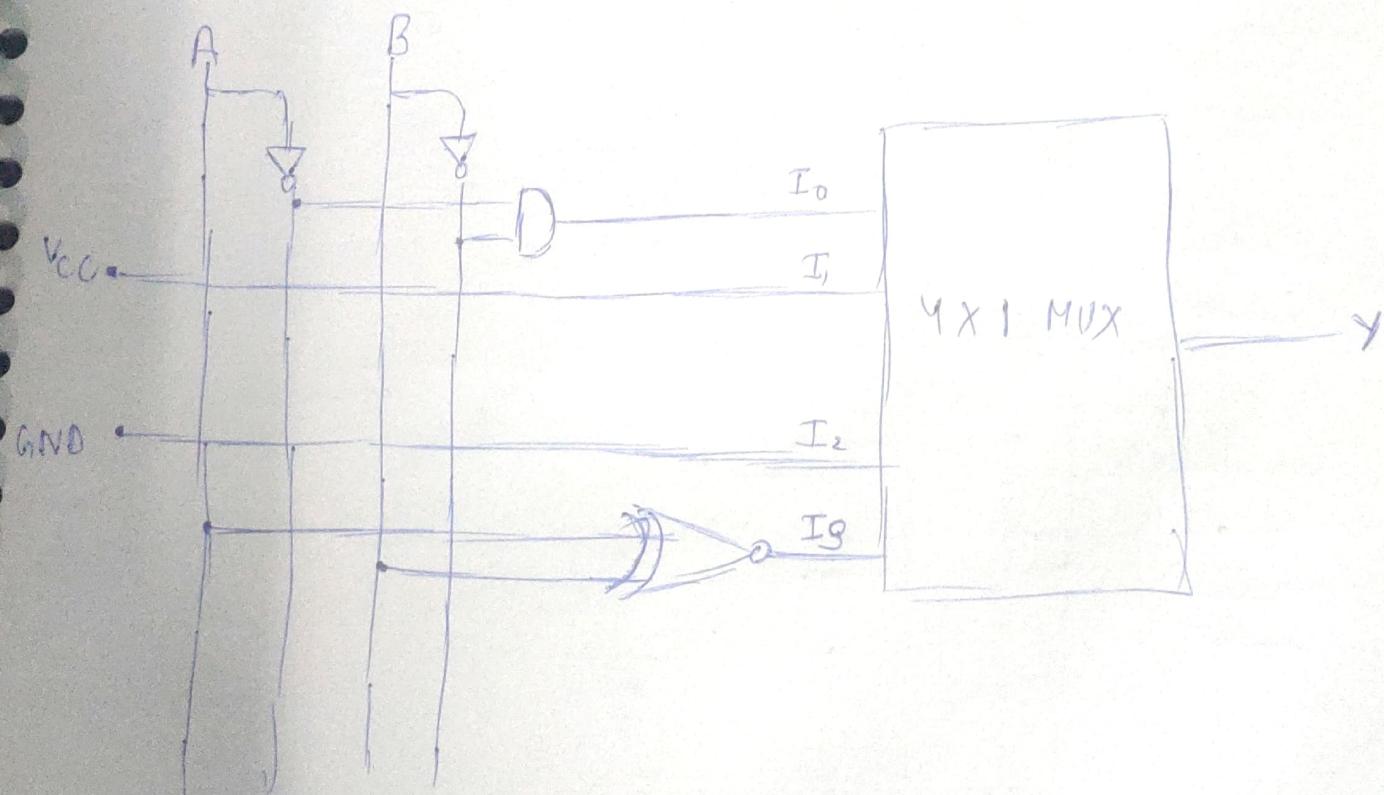
XOR

$\oplus$

		CD			
		00	01	10	11
AB	00	①	②	③	
	01	4	⑤	6	7
AB	10	8	⑨	10	11
	11	12	⑬	14	⑮
		$\bar{A}\bar{B}$	1	0	$A \odot B$

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

$$A \odot B$$



$$\text{Ques } P(A, B, C, D) = \sum m(0, 1, 3, 5, 9, 13, 15)$$

Implement it using 2x1 MUX.

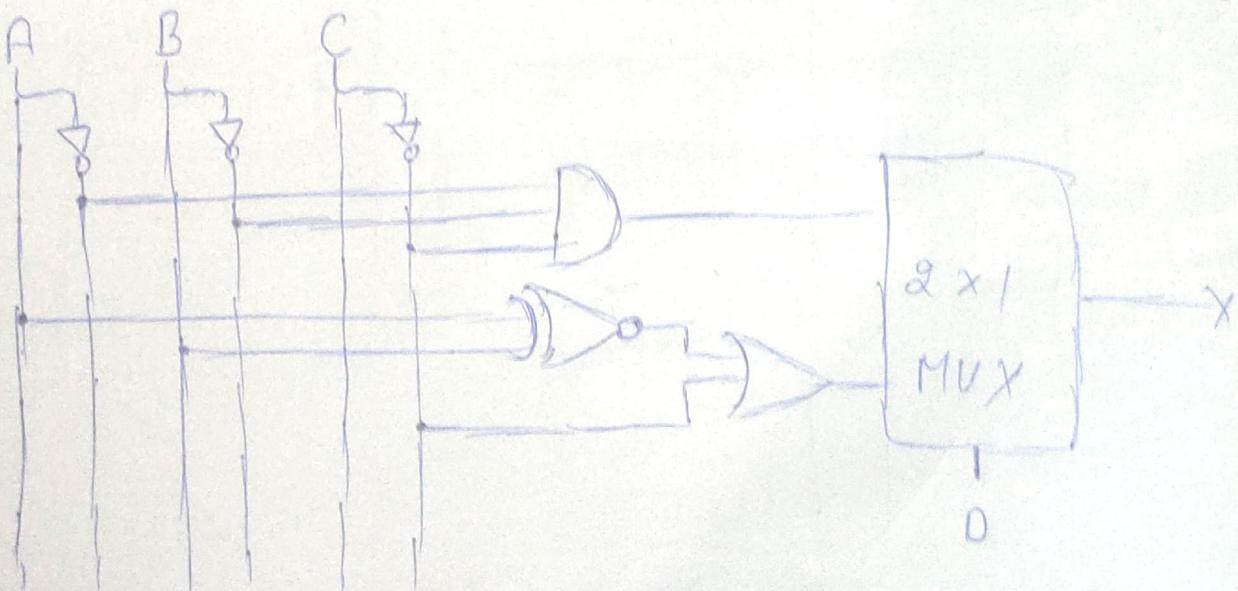
No of variable = 4

No of std. func. =  $2^4 = 16$

No of enpd = 2

$A B C$	$D$	
000	0	①
001	2	③
010	4	⑤
011	6	7
100	8	⑨
101	10	4
110	12	⑬
111	14	⑮
	$\bar{A} \bar{B} \bar{C}$	$A \oplus B + \bar{C}$

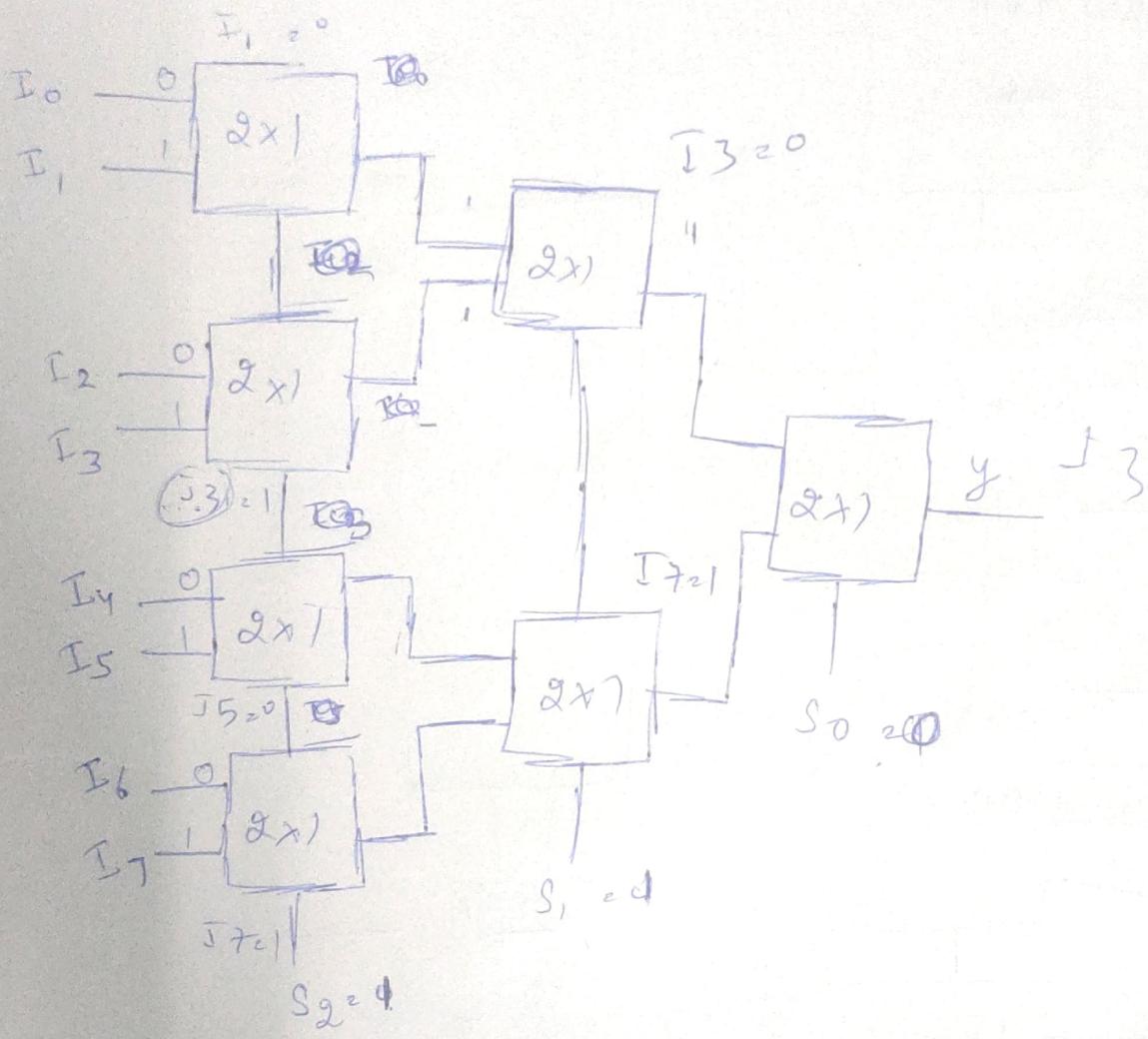
$$\begin{aligned}
 & \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + \bar{A} B \bar{C} + A \bar{B} \bar{C} \\
 & + A B \bar{C} + A B C \\
 & = \bar{A} \bar{B} (\bar{C} + C) + A B (C + \bar{C}) + \bar{A} B C \\
 & = \bar{A} \bar{B} + A B + \bar{C} (\bar{A} B + A B) \\
 & = A \oplus B + \bar{C} (A \oplus B) \\
 & = \overline{A \oplus B} + \bar{C} (A \oplus B) \\
 & = (\overline{A \oplus B} + \bar{C})(\overline{A \oplus B} + A \oplus B) \\
 & = \overline{A \oplus B} + \bar{C} = A \oplus B + \bar{C}
 \end{aligned}$$



# Higher Order Mux Using Lower Order

Eg ① Design 8x1 Mux using 2x1 MUX.

$$= \frac{8}{2} = \frac{4}{2} = \frac{2}{2} = 1$$

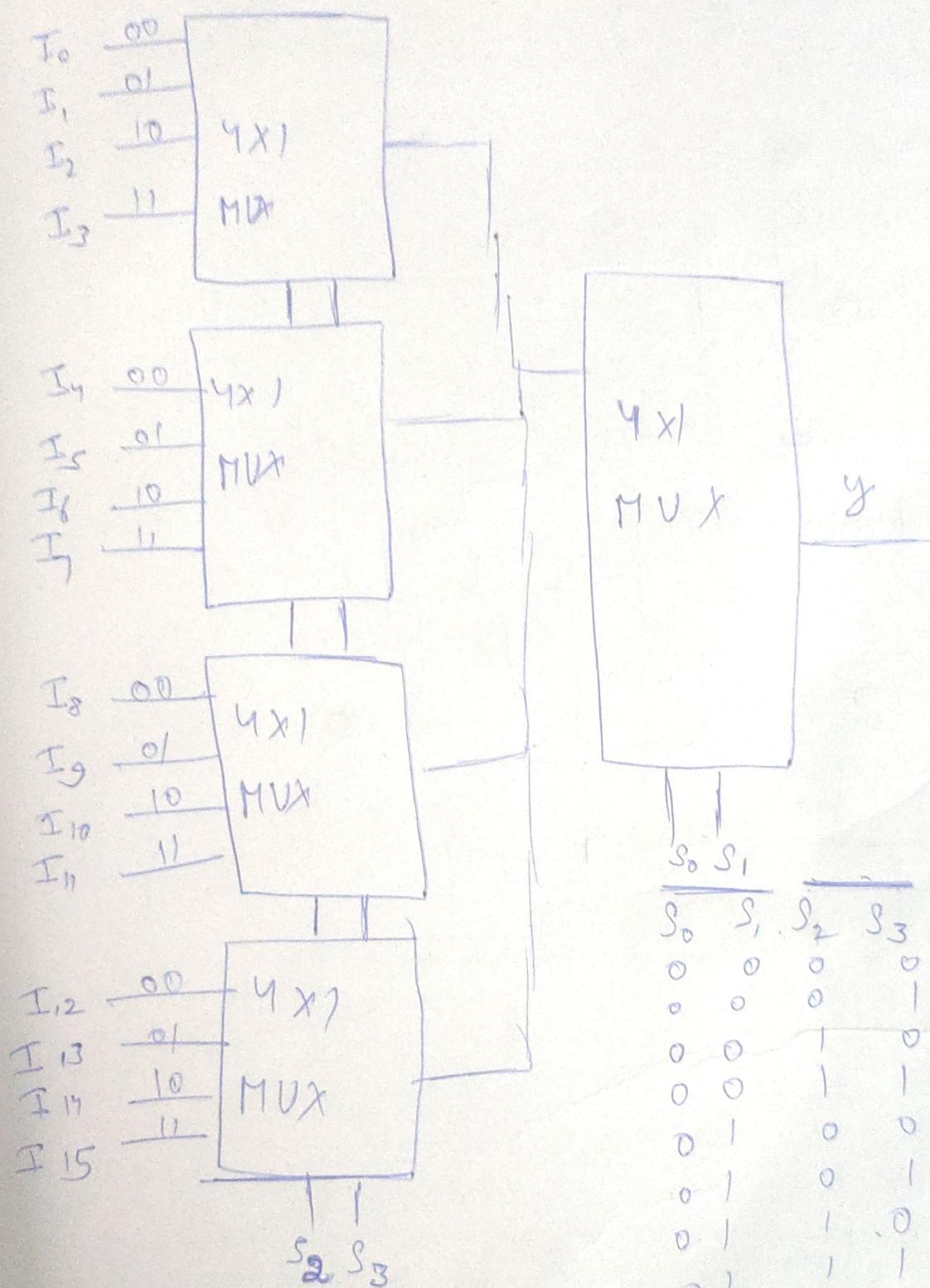


Truth Table

S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	Y
0	0	0	I <sub>0</sub>
0	0	1	I <sub>1</sub>
0	1	0	I <sub>2</sub>
0	1	1	I <sub>3</sub>
1	0	0	I <sub>4</sub>
1	0	1	I <sub>5</sub>
1	1	0	I <sub>6</sub>
1	1	1	I <sub>7</sub>

⑪ Design 16x1 Mux using 4x1 Mux

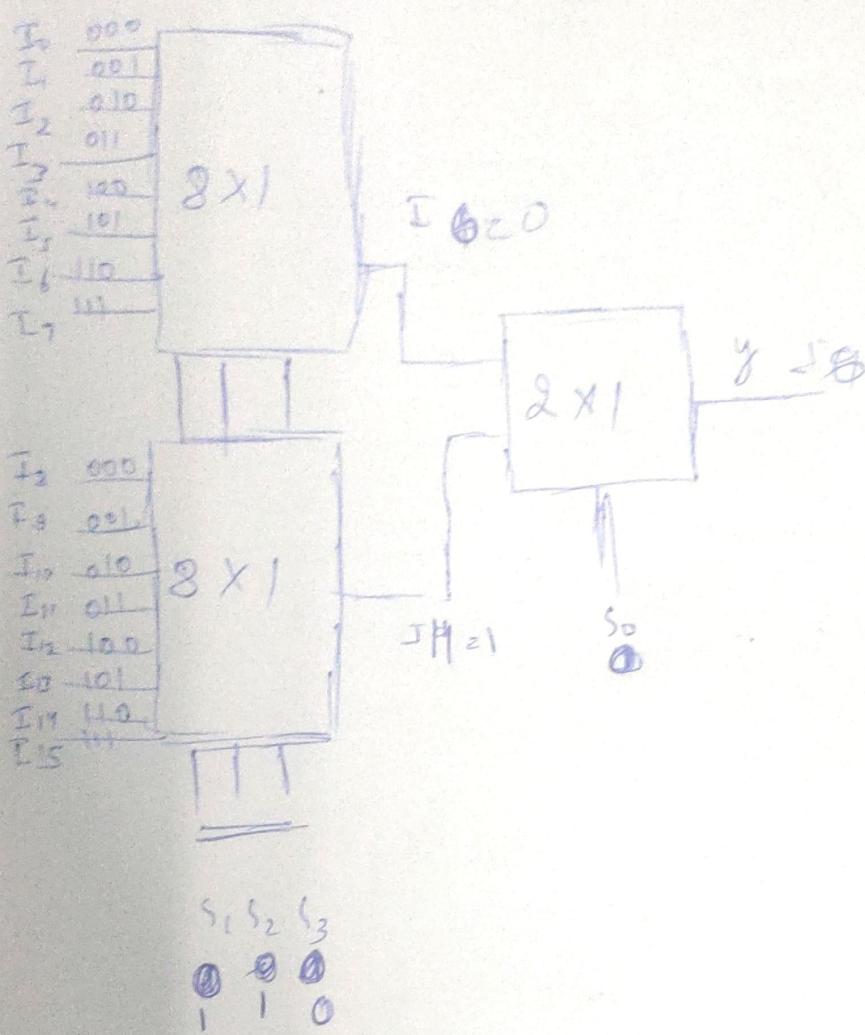
$$\frac{16}{4} = \frac{4}{4} = 1$$



$S_0, S_1$	$S_2$	$S_3$	$y$
0 0	0	0	$I_0$
0 0	0	0	$I_1$
0 0	1	0	$I_2$
0 0	1	1	$I_3$
0 1	0	0	$I_4$
0 1	0	1	$I_5$
0 1	1	0	$I_6$
0 1	1	1	$I_7$
1 0	0	0	$I_8$
1 0	0	1	$I_9$
1 0	1	0	$I_{10}$
1 0	1	1	$I_{11}$
1 1	0	0	$I_{12}$
1 1	0	1	$I_{13}$
1 1	1	0	$I_{14}$
1 1	1	1	$I_{15}$

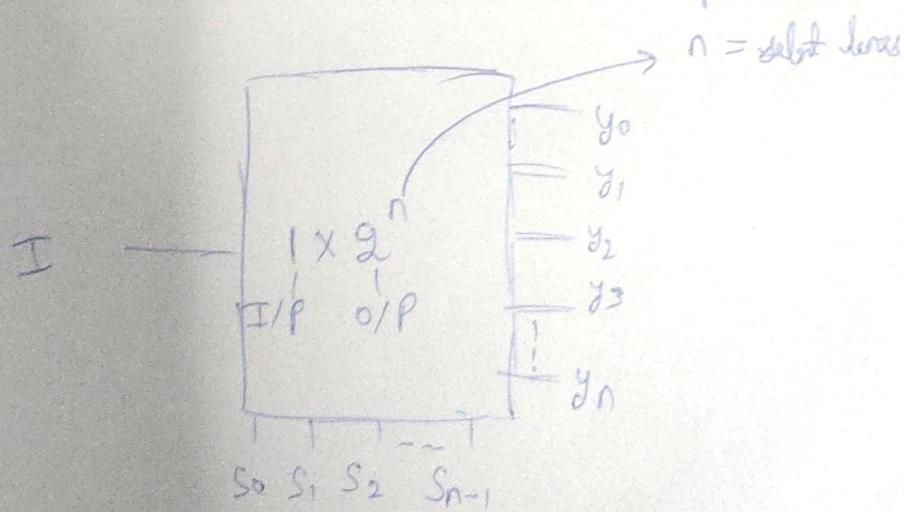
Design 16x1 MUX using 8x1 MUX.

$$\frac{16}{8} = \frac{2}{2} = 1$$

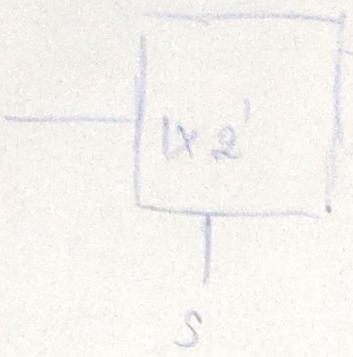


S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	Y
0	0	0	0	I <sub>0</sub>
0	0	0	1	I <sub>1</sub>
0	0	1	0	I <sub>2</sub>
0	0	1	1	I <sub>3</sub>
0	1	0	0	I <sub>4</sub>
0	1	0	1	I <sub>5</sub>
0	1	1	0	I <sub>6</sub>
0	1	1	1	I <sub>7</sub>
1	0	0	0	I <sub>8</sub>
1	0	0	1	I <sub>9</sub>
1	0	1	0	I <sub>10</sub>
1	0	1	1	I <sub>11</sub>
1	1	0	0	I <sub>12</sub>
1	1	0	1	I <sub>13</sub>
1	1	1	0	I <sub>14</sub>
1	1	1	1	I <sub>15</sub>

Demultiplexer - one to many means we have single i/p & have multiple o/p.



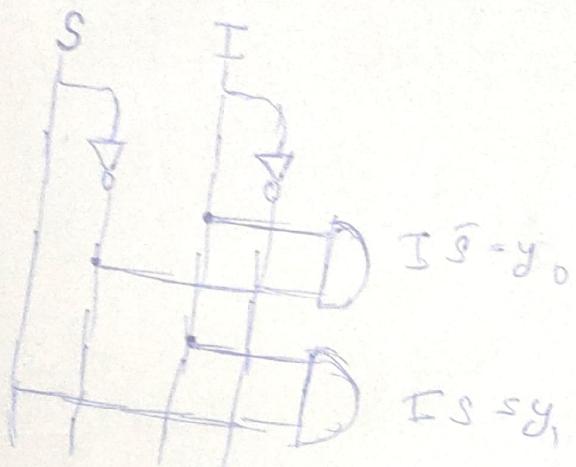
By Design and implement 1x2 Demux



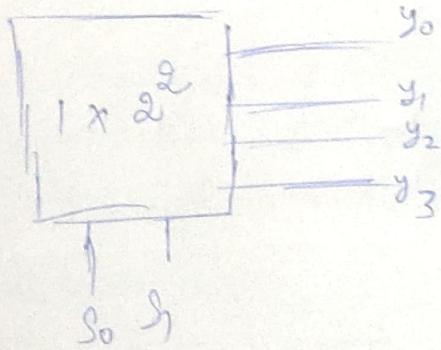
S	y <sub>0</sub>	y <sub>1</sub>
0	I	0
1	0	I

$$y_0 = I \bar{S}$$

$$y_1 = I S$$



By Design & implement 1x4 Demux



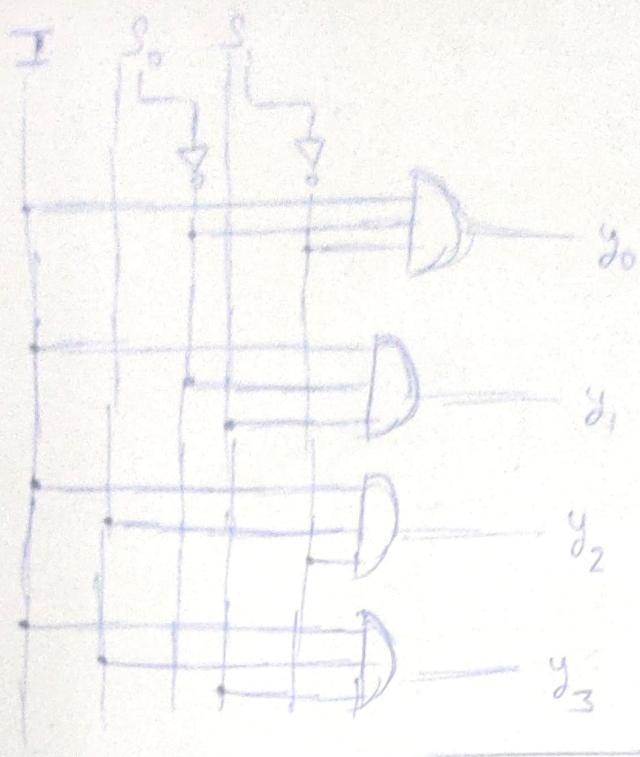
$$y_0 = I \bar{S}_0 \bar{S}_1$$

$$y_1 = I \bar{S}_0 S_1$$

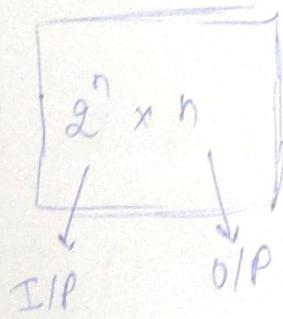
$$y_2 = I S_0 \bar{S}_1$$

$$y_3 = I S_0 S_1$$

S <sub>0</sub>	S <sub>1</sub>	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I



Encoder (Multiplexer) — Many to Many words { O/P is in form of binary }



$$E_s \quad (8 \times 3) = 2^3 \text{ I/P} \\ 3 \text{ O/P (Binary)}$$

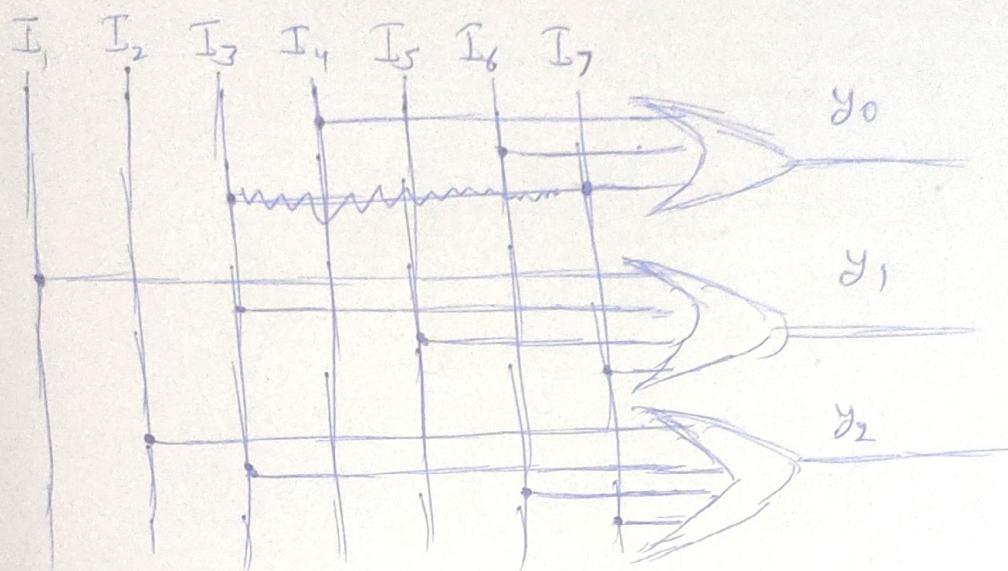
I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	0	1	0	1	0	1
0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$y_0 = I_4 + I_5 + I_6 + I_7$$

$$y_1 = I_1 + I_3 + I_5 + I_7$$

$$y_2 = I_2 + I_3 + I_6 + I_7$$

11111111  
3421  
15



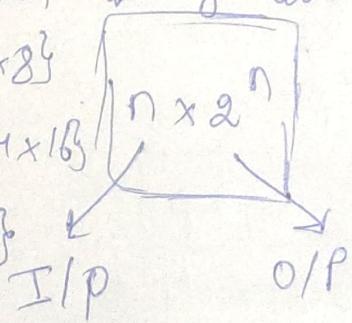
Decoder: ① multiple input & output

② converts binary code to other

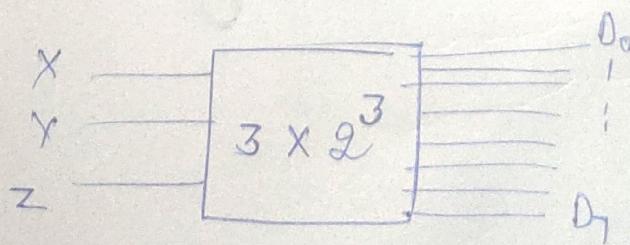
③ (2)<sub>2</sub> to (2)<sub>8</sub> {3x8}

④ (2)<sub>2</sub> to (2)<sub>16</sub> {4x16}

⑤ (2)<sub>2</sub> to (2)<sub>10</sub> {4x10}



⑥ 3x8 Decoder

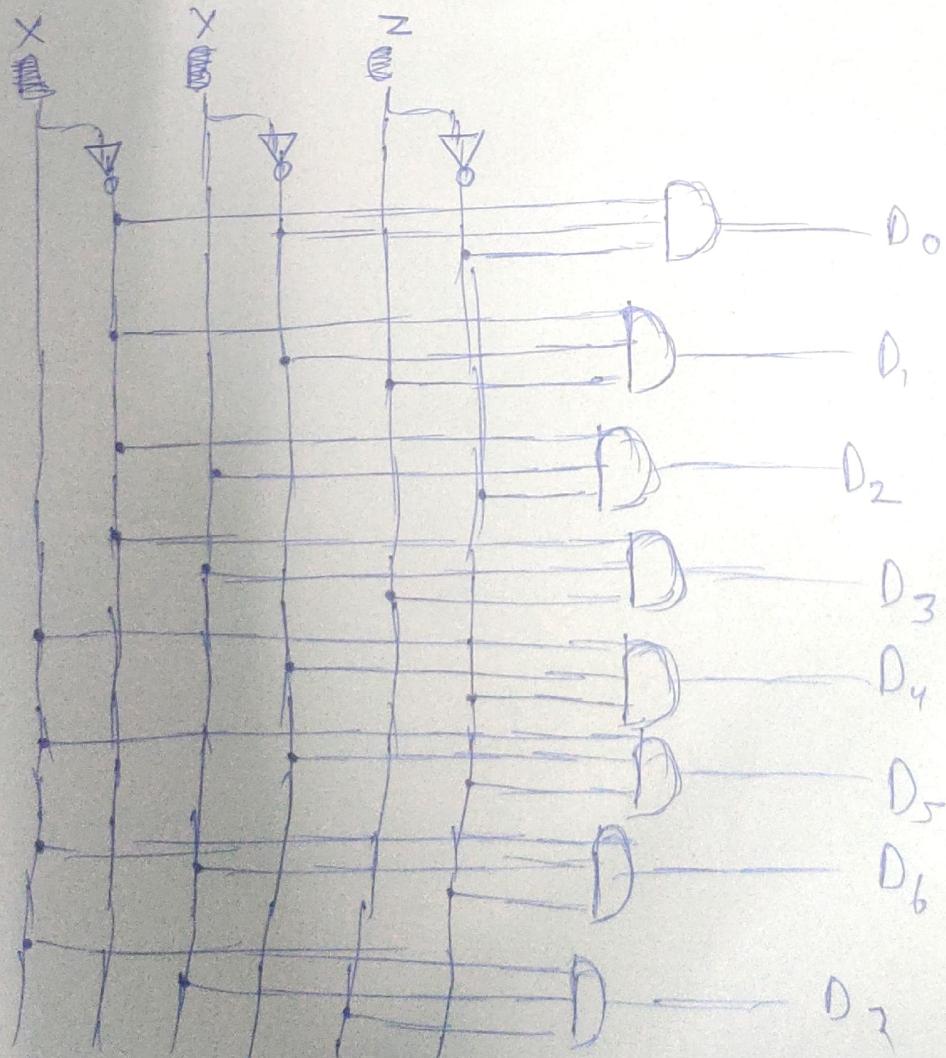


	$x$	$y$	$z$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

$$D_0 = \overline{ABC}, D_1 = AB$$

$$D_0 = \overline{XYZ}, D_1 = \overline{XY}Z, D_2 = \overline{X}YZ, \cancel{D_2 = XYZ}, D_3 = \overline{X}Y\bar{Z}$$

$$D_4 = X\overline{Y}\bar{Z}, D_5 = X\overline{Y}Z, D_6 = XY\bar{Z}; D_7 = XYZ$$



Comparator's combination used to compare bits.

1 bit comparator  $\Rightarrow 2 \text{ I/Ps}$

2 bit comparator  $\Rightarrow 4 \text{ I/Ps}$

3 bit comparator  $\Rightarrow 6 \text{ I/Ps}$

Eg 1 Bit comparator

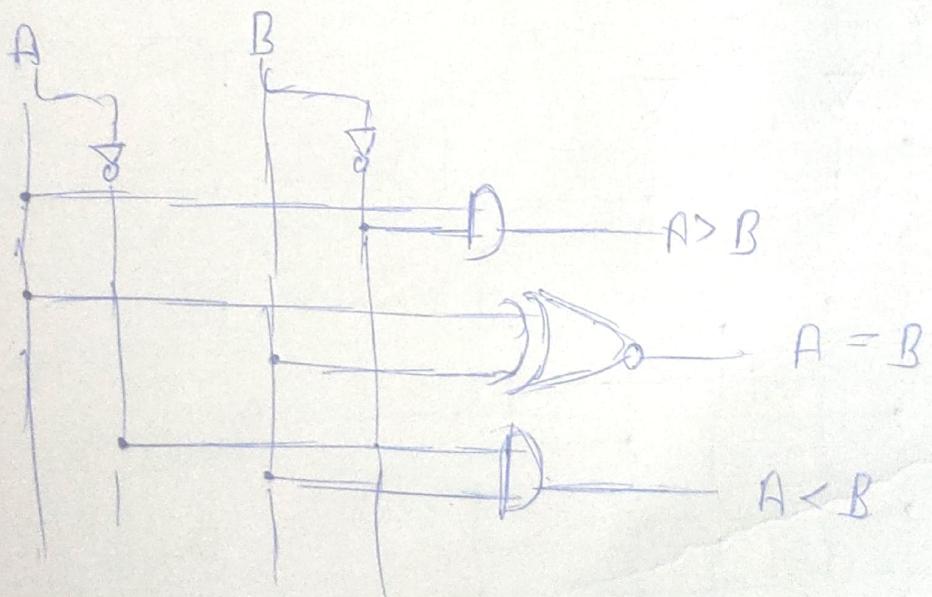
A	B	$A > B$	$A = B$	$A < B$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Boolean expression

$$A > B \Rightarrow A\bar{B}$$

$$A = B \Rightarrow \bar{A}\bar{B} + AB = AOB$$

$$A < B \Rightarrow \bar{A}B$$



Sequential Circuits : Circuits whose outputs are depend on past and present input.

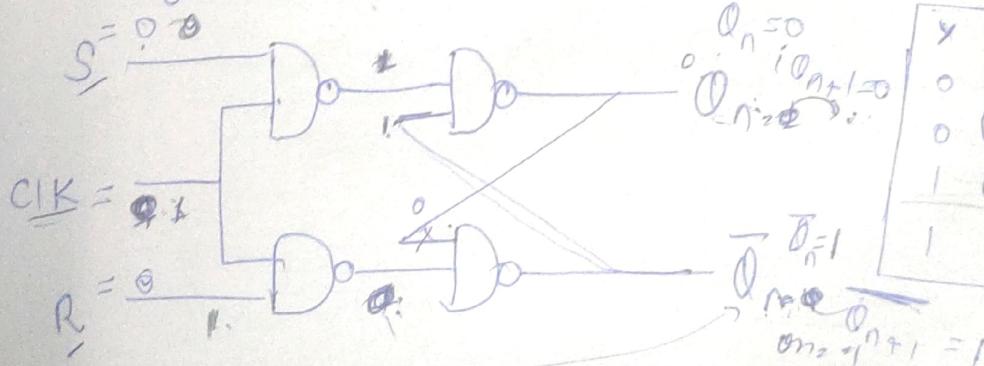
### ① Step Flops -

- ① Basic memory element which stores only one bit either 0 or 1.
- ② It has 2 O/P which are complement of each other.
- ③ Bistable Multivibrator - because it has only 2 stable states i.e either 0 or 1.
- ④ They are used in frequency divider circuit.

### ② SR flip-flop :

(Set Reset) flip flop

i) Circuit diagram



CLK = used to control memory elements

NAND

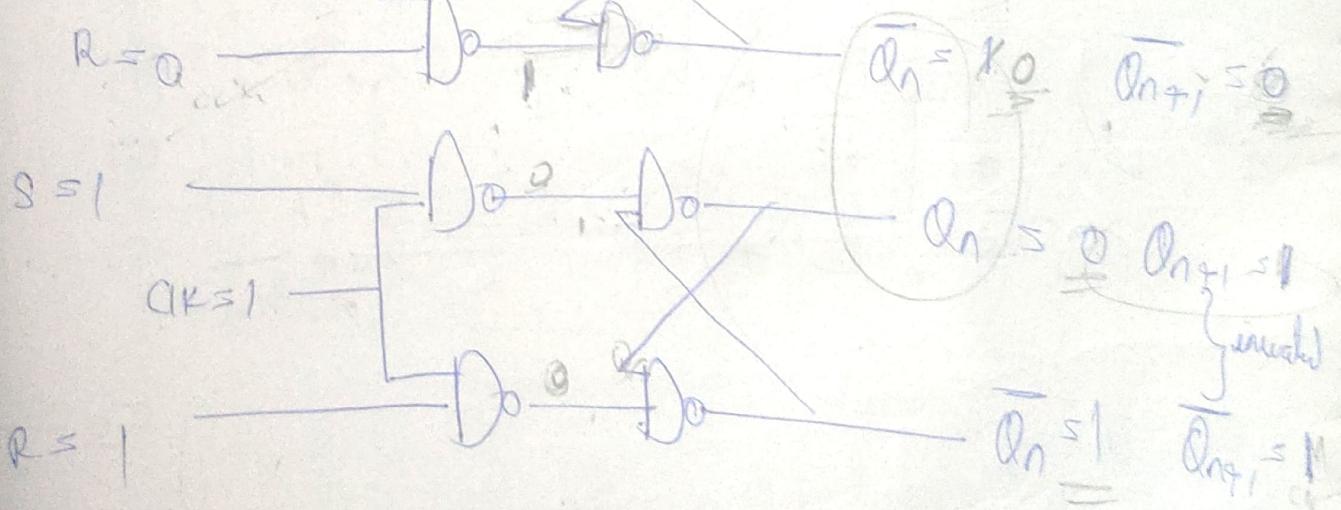
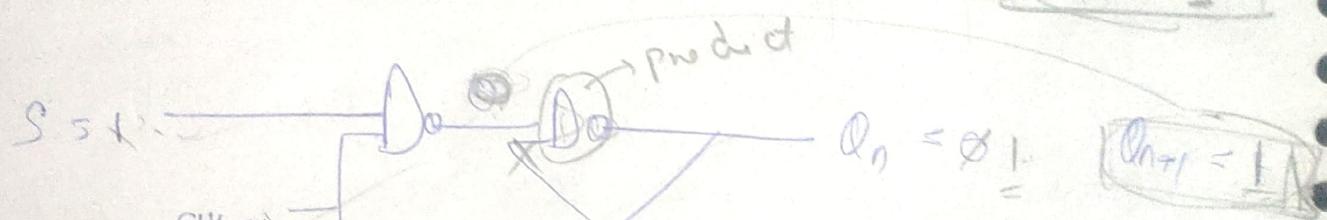
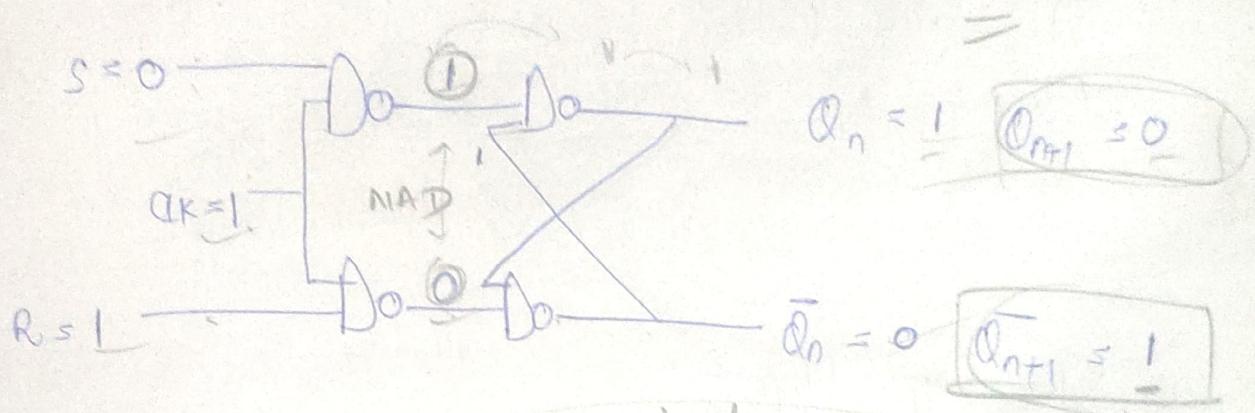
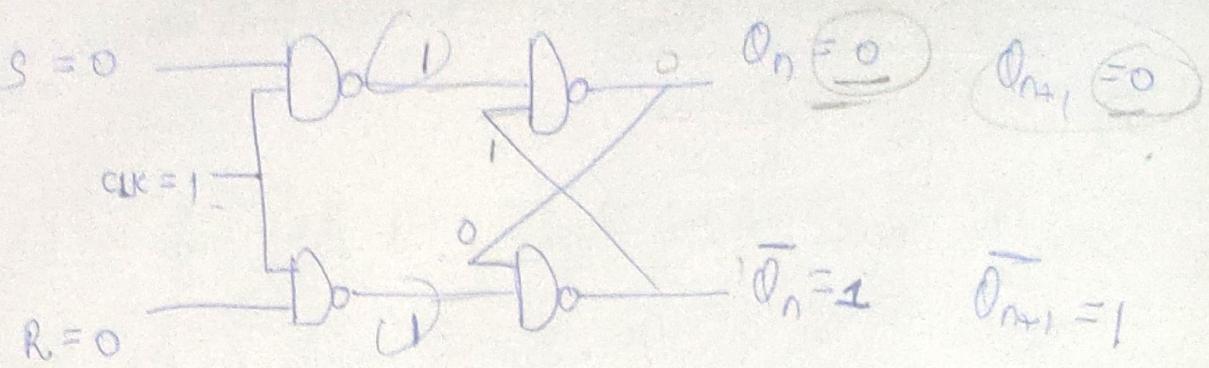
x	y	$\bar{o}$
0	0	1
0	1	0
1	0	0
1	1	1

$$Q_n = \bar{Q}_{n+1} = 1$$

Truth Table

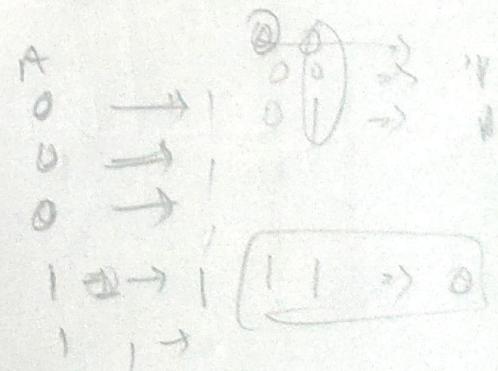
$\bar{o} \rightarrow 1$

CLK	S	R	$Q_{n+1}$
0	*	*	$Q_n$ (no change)
1.	0	0	$Q_n$
1	0	1	0
1	1	0	1
1	1	1	Invalid



Characteristic Table

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



{do not care conditions}

	00	01	11	10
0	0	1	3	2
1	1 <sub>y</sub>	1 <sub>5</sub>	x <sub>7</sub>	x <sub>c</sub>

$$C.E. = S + \bar{R}d_n$$

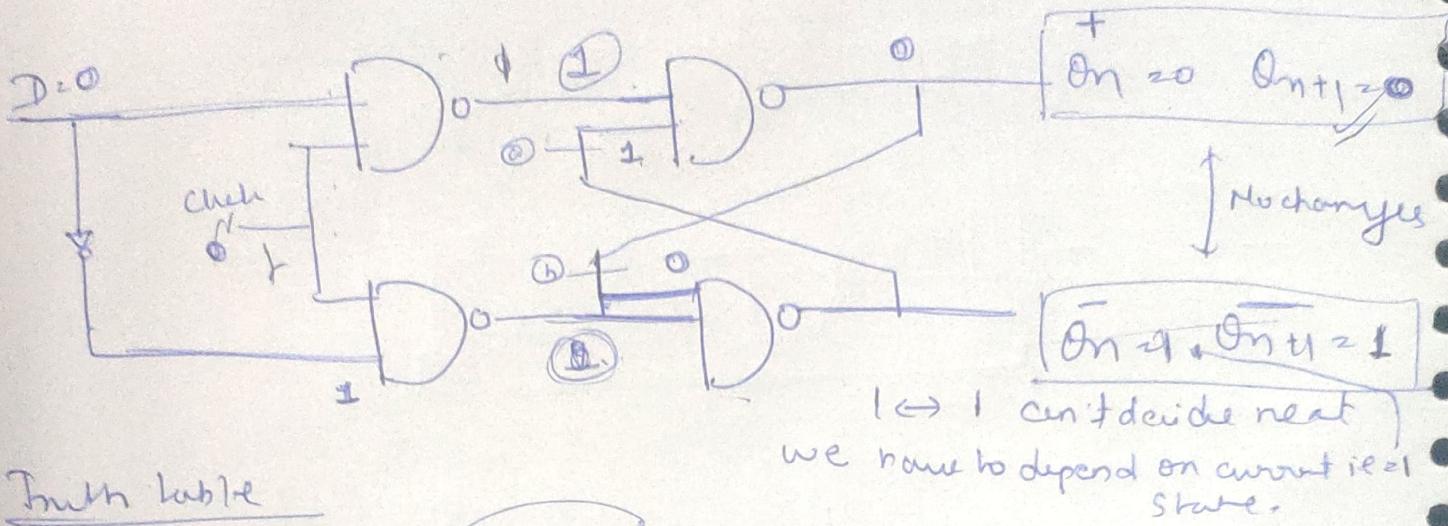
Excitation Table

$O_n$	$O_{n+1}$	$S$	$R$
0	0	0	*
0	1	1	0
1	0	0	1
1	1	*	0

$S$	$R$	$O_n$	$O_{n+1}$	$O_n$	$O_{n+1}$	$S$	$R$
0	0	0	0	-	0	0	0
0	0	1	1	-	0	0	0
0	1	0	0	-	0	1	0
0	1	1	0	-	0	1	0
1	0	0	1	-	1	0	1
1	0	1	1	-	1	1	0
1	1	0	1	-	1	0	0
1	1	1	1	-	1	1	0

## D flip flop

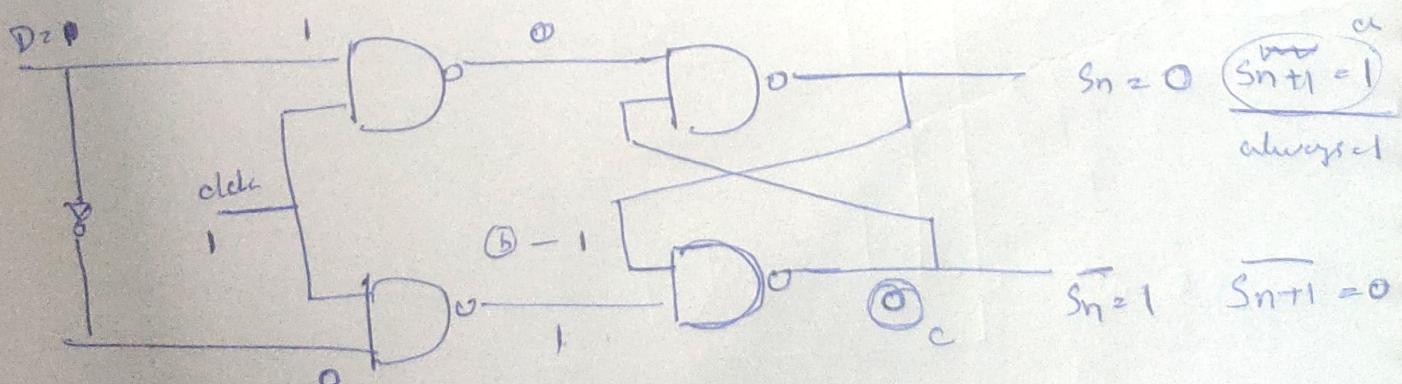
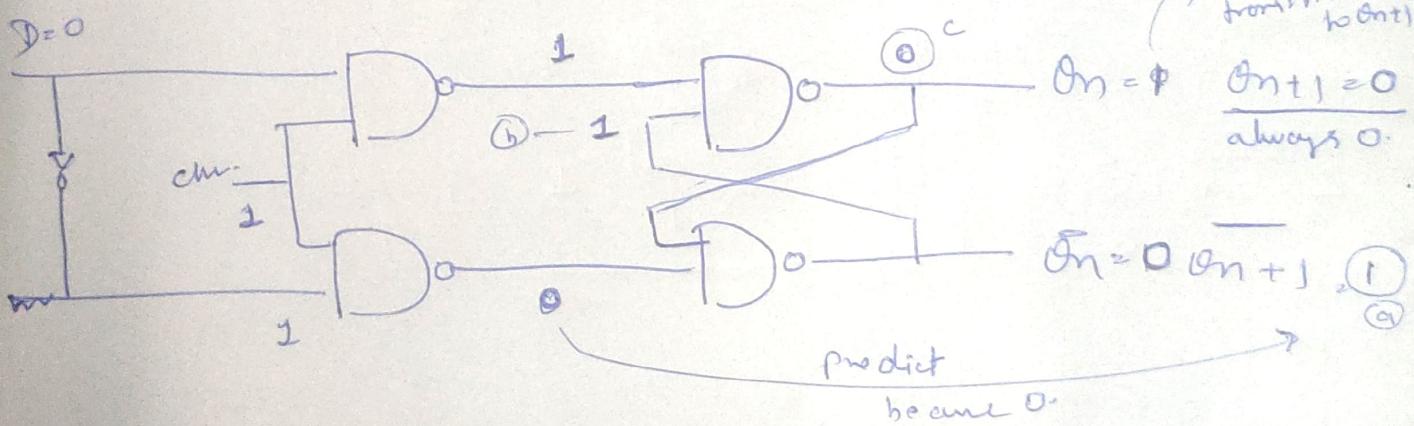
Input news can be same  
if input is invalid  
in case of SR  
flip flop.



Truth table

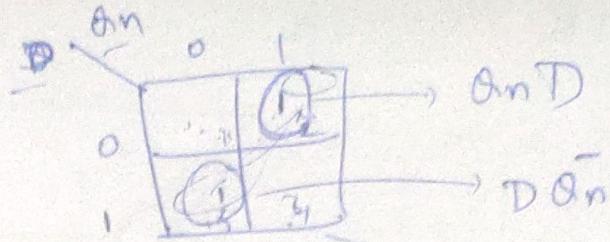
clk	D	$Q_{n+1}$
0	X	On
1	0	0
1	1	1

Shorted column.



### Characteristic table

D	$\theta_m$	$\theta_{n+1}$	$D$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	2



Find the value in left side  
of right 1's.

### Characteristic Eq'n

$$\theta_{n+1} = D\bar{\theta}_m + D\theta_n$$

$$\theta_{n+1} = D(\bar{\theta}_m + \theta_n)$$

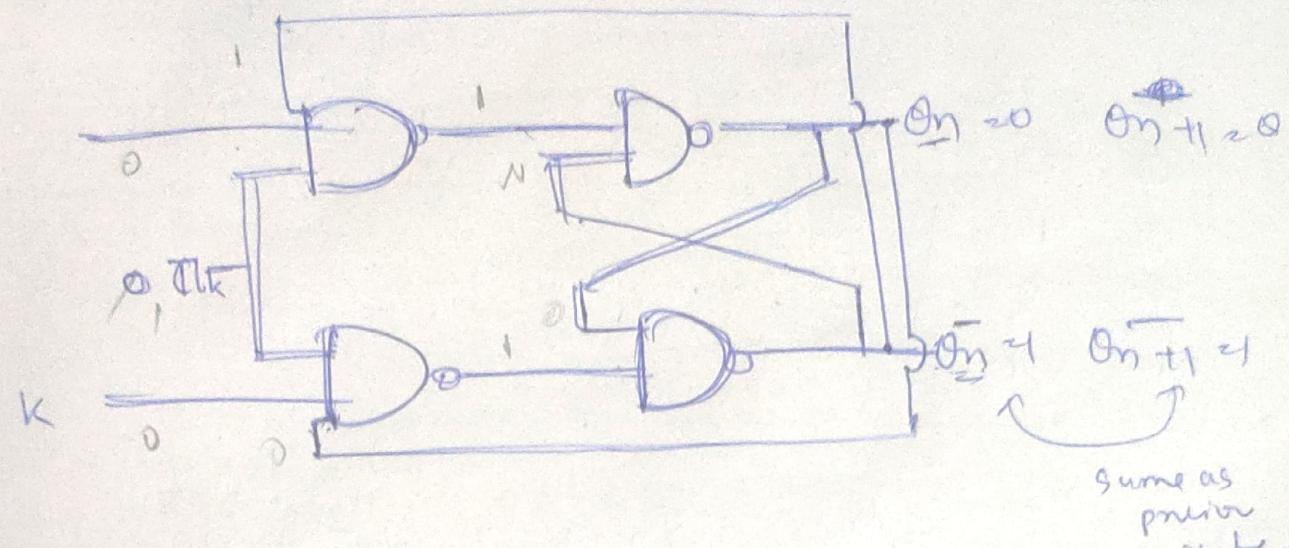
$$\boxed{\theta_{n+1} = D}$$

### Execution table

$\theta_m$	$\theta_{n+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

# JK Flip Flop

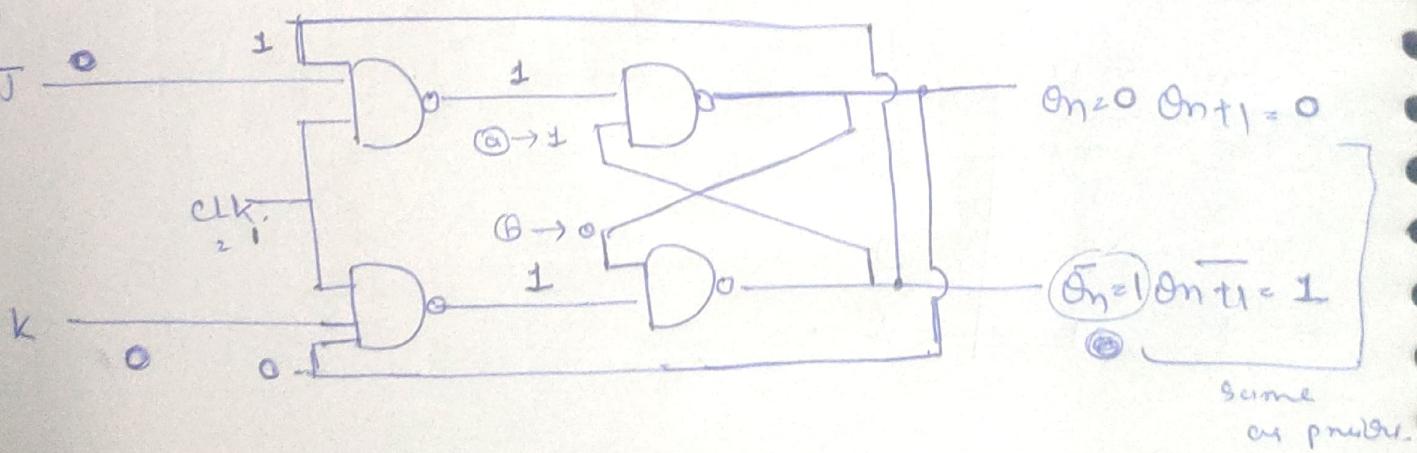
①



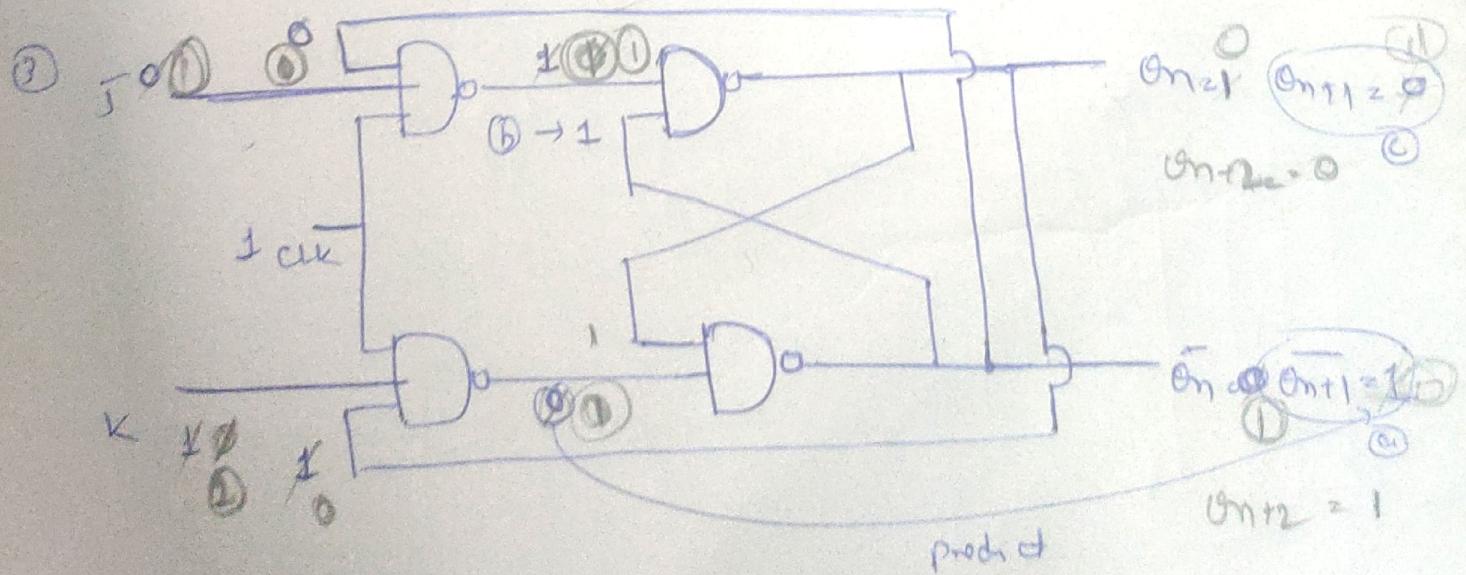
Truth table

CLK	J K	$\bar{Q}_n + 1$
0	X X	$\bar{Q}_n$
1	0 0	

②



③



## Truth Table

En	J	K	On+1
0	X	X	On
1	0	0	On
1	0	1	0
1	1	0	1
1	1	1	On + toggling action

SOP

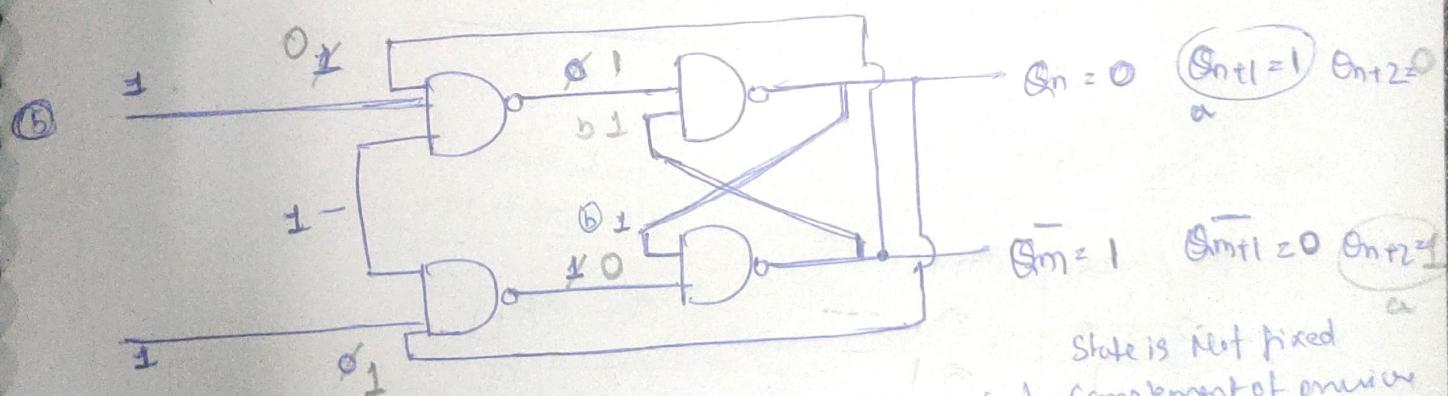
J	K	00	01	11	10
0	0	1	1	0	0
1	0	1	1	0	1

$$CE(On+1) = \bar{K}On + J\bar{O}n$$

$$CE(On+1) = J\bar{O}n + \bar{K}On$$

## Characteristic Table

J	K	On	On+1	On	On+1	On J	K
0	0	0	0	0	0	0	0
0	0	1	0	0	1	10, 11	
0	1	0	0	1	0		
0	1	1	0	1	1	01, 11	
1	0	0	1	1	0	10, 00	
1	0	1	1	1	0		
1	1	0	0	0	1		
1	1	1	1	1	1		



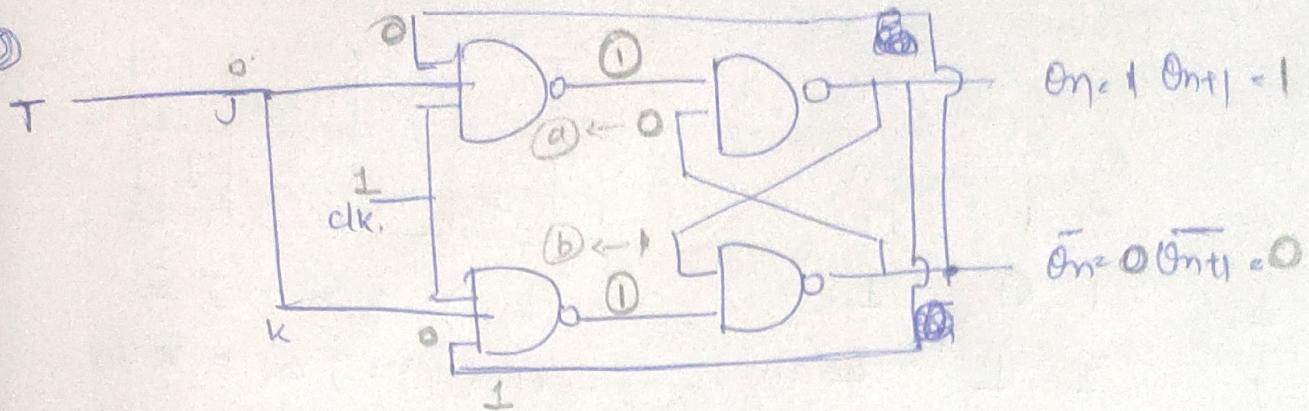
State is not fixed  
just complement of previous state.

## Excitation table

En	On+1	J	K	J <sub>R</sub>	K <sub>R</sub>	On	On+1	J	K
0	0	0	0	0	X	0	0	0	X
0	1	1	0	1	X	0	1	1	X
1	0	0	1	1	X	1	0	X	1
1	1	0	0	X	0	1	1	X	0

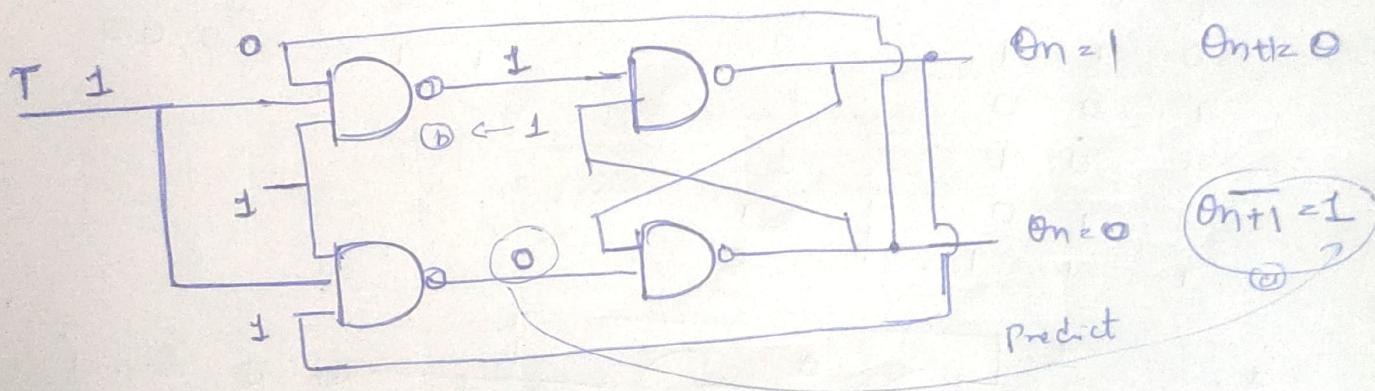
## T flip-flop

②



Truth table

CLK	T	$Q_{n+1}$
0	X	$Q_n$
1	0	$Q_n$
1	1	<u><math>Q_n</math></u>

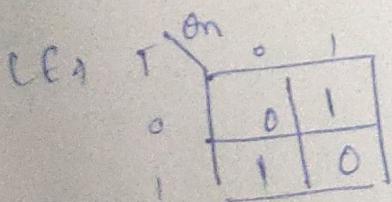


Characteristic table

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

Excitation table

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



$$CF = T \bar{Q}_n + T Q_n$$

$$CF = T \oplus Q_n$$

\* Flip-flop conversion (Remember the combination table of each flip-flop)

Q1 Convert T flip-flop into SR.

A1 Given - T

Required - SR

② CT of T

On	Untl	T
0	0	0
0	1	1
1	0	1
1	1	0

CT of SR

On	Untl	S	R
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

CT of SR

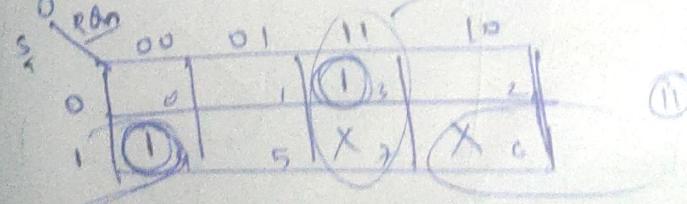
On	Untl	S	R	On	Untl
0	0	0	0	0	0
0	0	0	1	0	1
0	1	0	0	0	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	X	1	0
1	1	1	X	1	1

③ Combine both table  
Row(SR)      Row(T)

On	Untl	copy	find from T
S	R	Bn	Bn+1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	X
1	1	1	X

④ K-map

only consider S R On

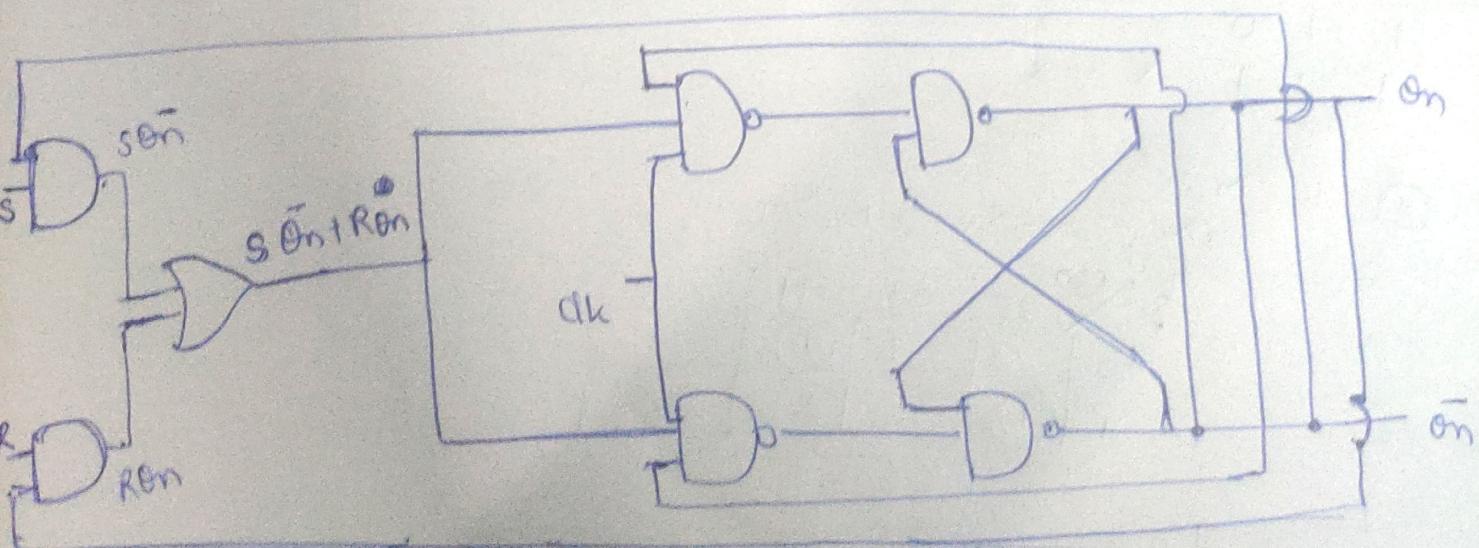
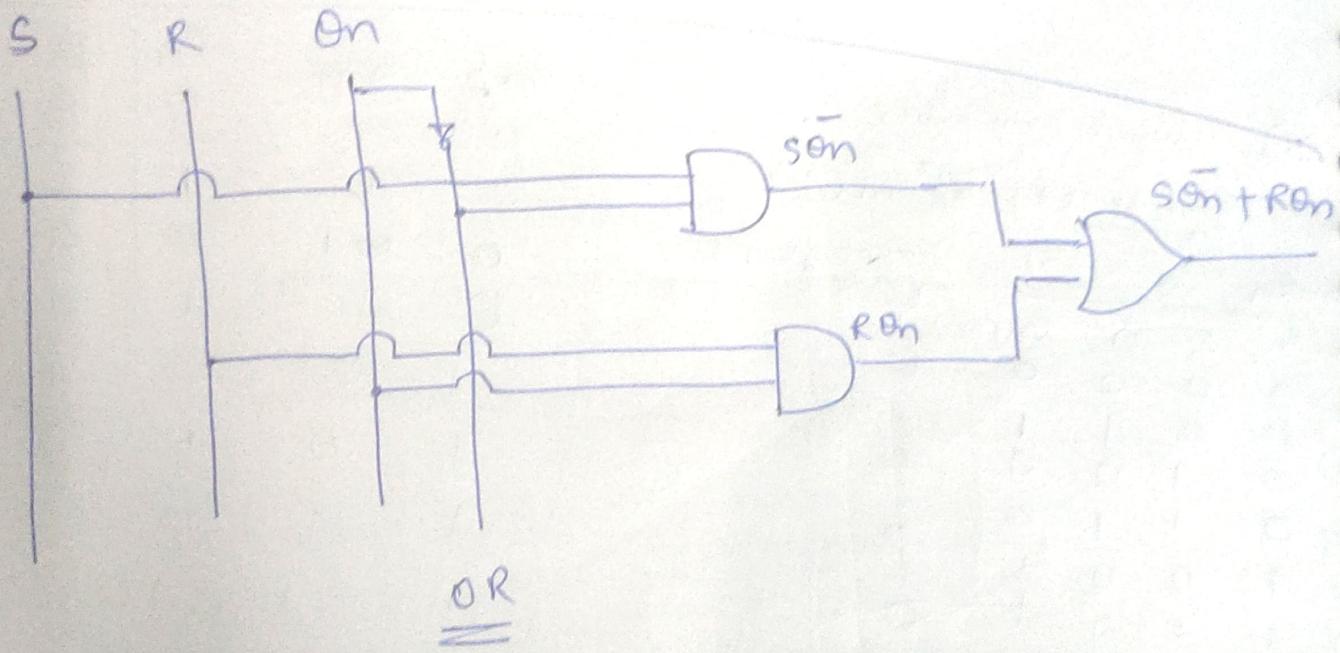
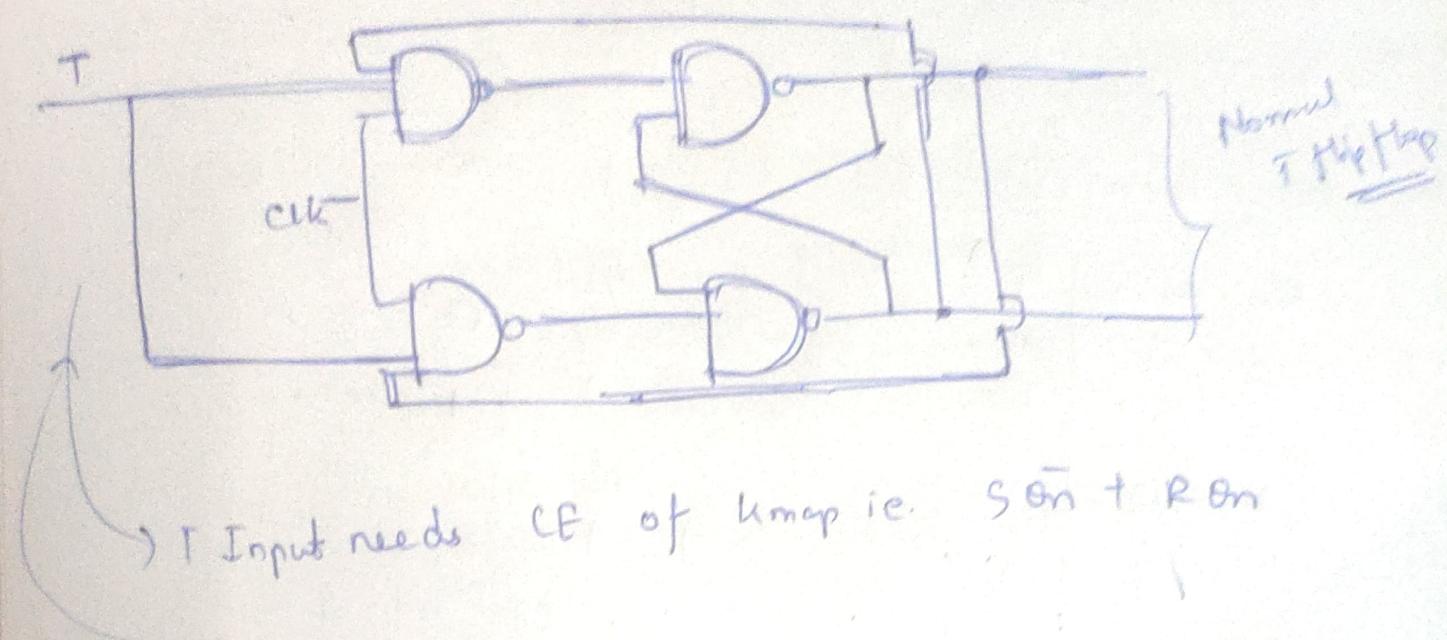


① R On    ② S On

$$CE = \bar{R} \cdot \bar{S} \cdot \bar{On} + R \cdot \bar{On}$$

⑤ Implementation

Create the TNS diagram / Input diagram / gms.



## \* Latches (Hold, lock)

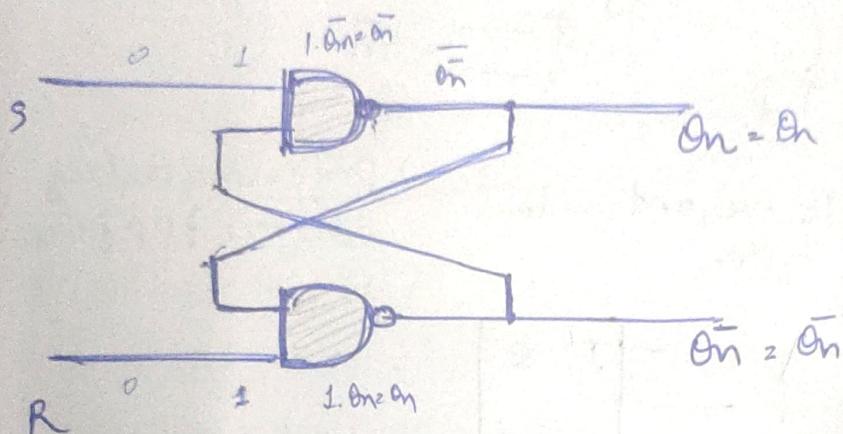
① basic memory element

② latches are level triggered.

③ basic building block using which flip-flop are made.  
(construction)

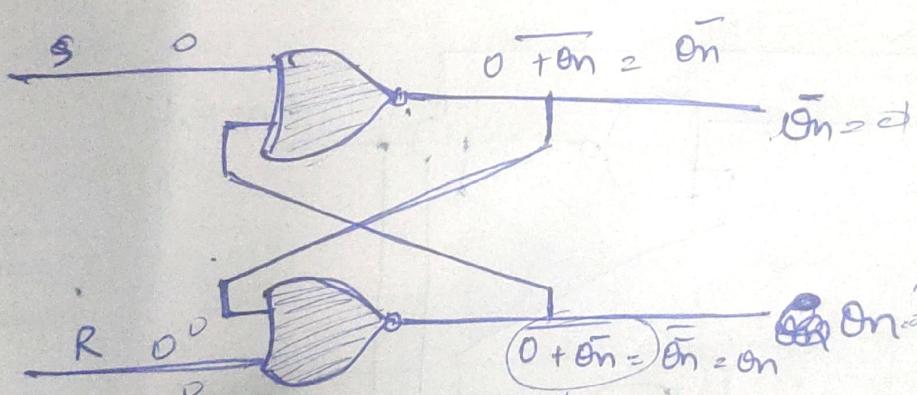
④ Capability of storing 1 bit.

### SR latch



S R	$0n + 1$
0 0	$\rightarrow 0$ (Invalid)
0 1	$\rightarrow 1$
1 0	$\rightarrow 0$
1 1	$\rightarrow 0n$ (Hold)

### SR latch (using NOR Gate)



If any one input is 1  
than out is 0  
opposite ~~NOR~~ NAND.

Answer Out from  
this side.

S R	$0n + 1$
0 0	$0n$ (Hold)
0 1	0 0
1 0	1
1 1	(Invalid State)

NOR

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

Solution:

$$\begin{aligned} & \overline{0 + 0n} \\ & \overline{1 \cdot 0n} = \overline{0n} \end{aligned}$$

$\circledcirc 0n$

## Race Around Condition (only in J-K flip flop)

↳ only occurs when condition 1 & condition 2 & condition 3 are true.

→ Type of J-K flip flop in which clock is used

condition 1: level triggered J-K flip flop. (clock must be used in same true level)

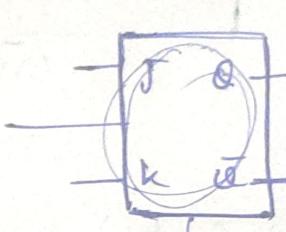
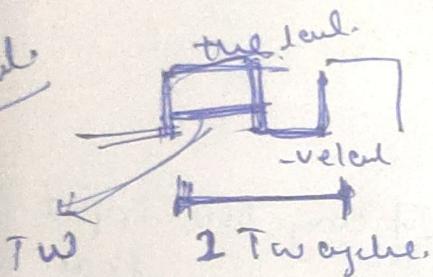
condition 2: when  $J = K = 1$  (Toggle mode)

condition 3:  $\text{TW} \geq T_d$

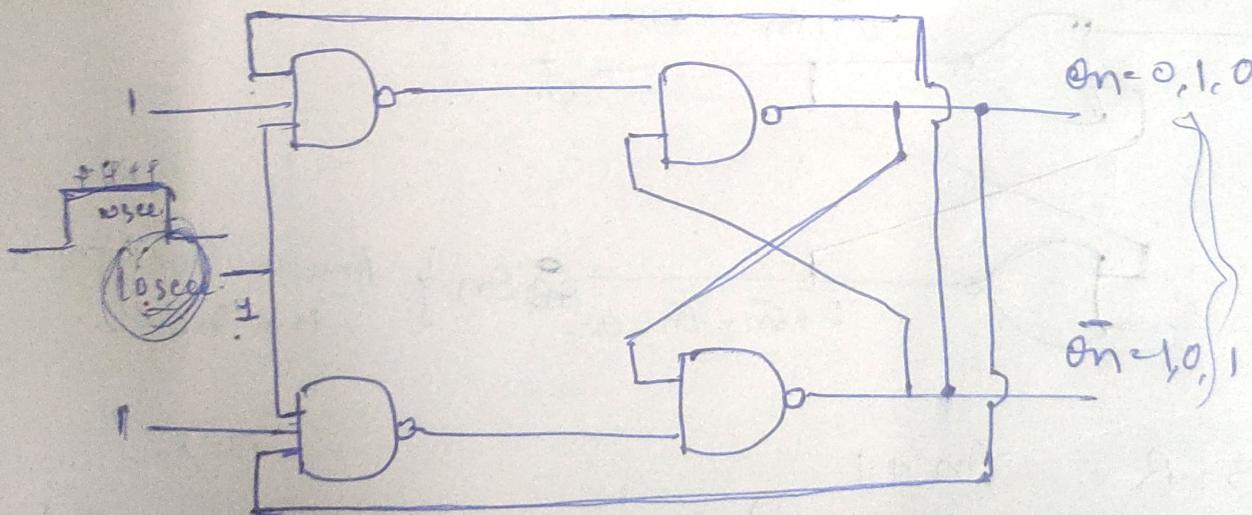
Time taken <sup>high</sup> single true, and -level clock cycle

processing delay inside flip-flop

or  
cycle of  
true & -level



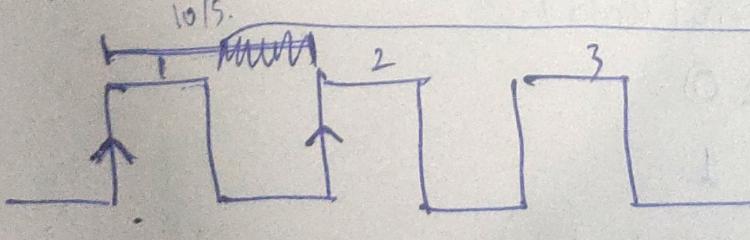
processing delay ( $T_d$ )



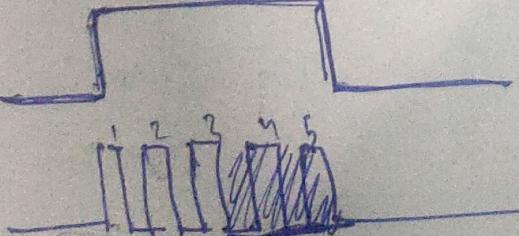
single clock pulse

clock

regured  
 $Q = 0$

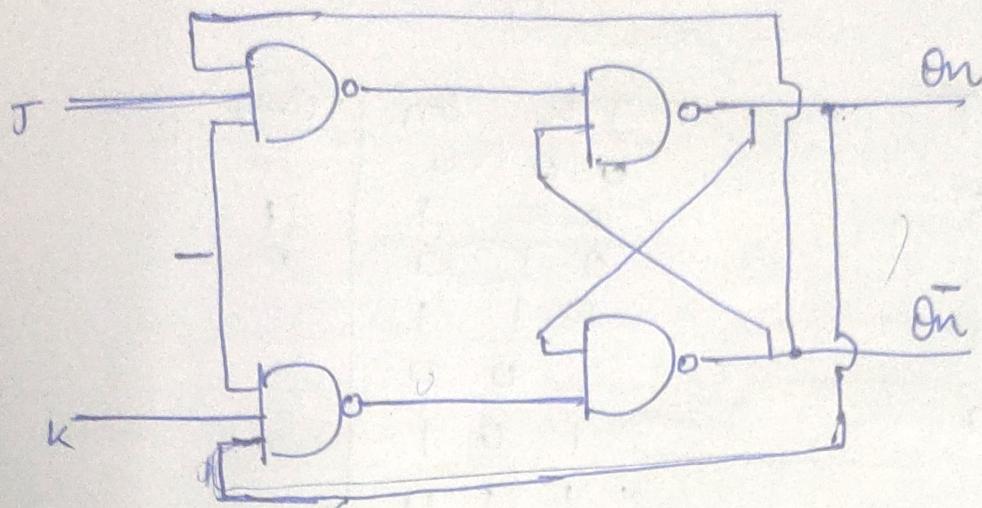


Output  $Q = 0$



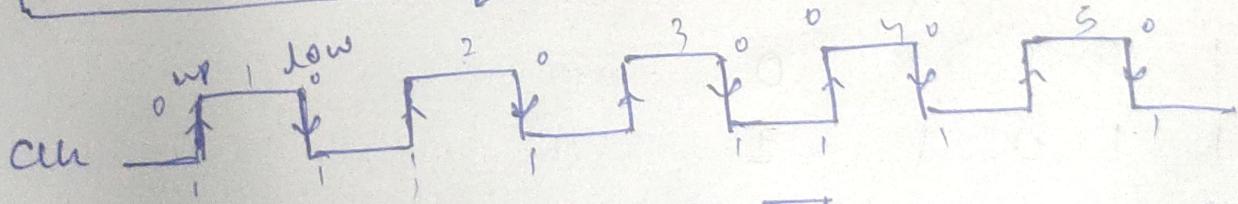
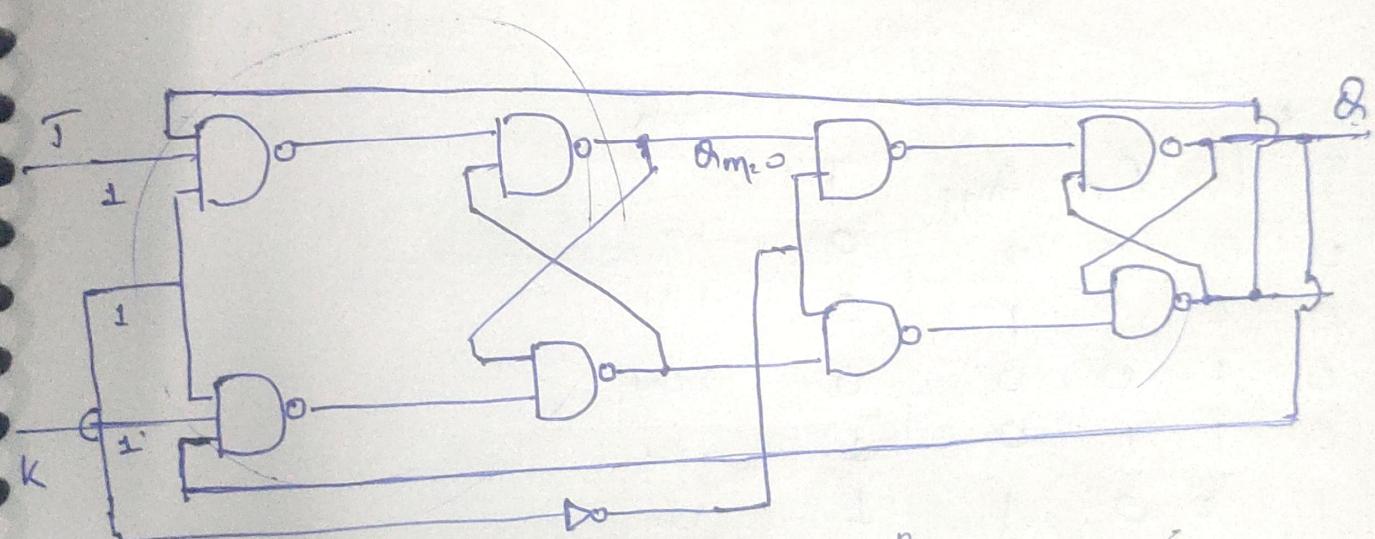
→ bits are toggling b/w 0 and 1 in single clock pulse. We are unable to predict output in a single clock pulse.

\* Master Slave - Jk Flip Flop (Solution of Race around condition).



\* Control This feed back lines ~~these~~ are creating toggling.  
Now 0 and 1 in Jk flip flop.

Solution: we make Master Jk flop and slave Jk flop.



$C_m \oplus C_n$

$(C_m \oplus C_n) \oplus Q_m$

low for long duration.

as  $Q_m$  is only working on up.

At single time only one value charged either  $Q_m$  or  $Q$  other works as memory.

hence when  $J=1, K=1$

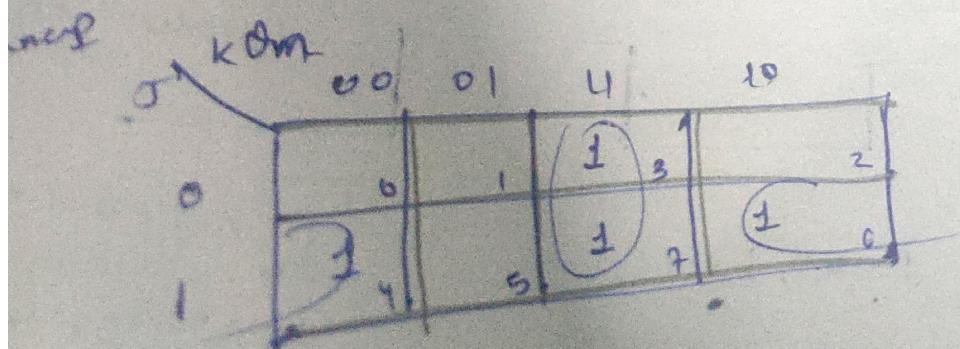
$Q_m$  achieve  $\{0, 1\}$  hence  $Q$  inacheive  $\{0, 1\}$ . Now toggling is output

Ques Convert T to J & K flip flop.

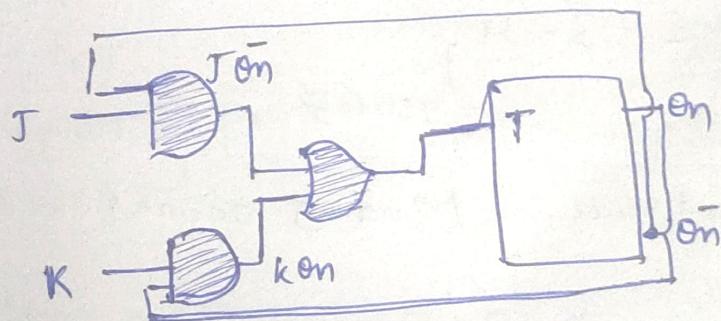
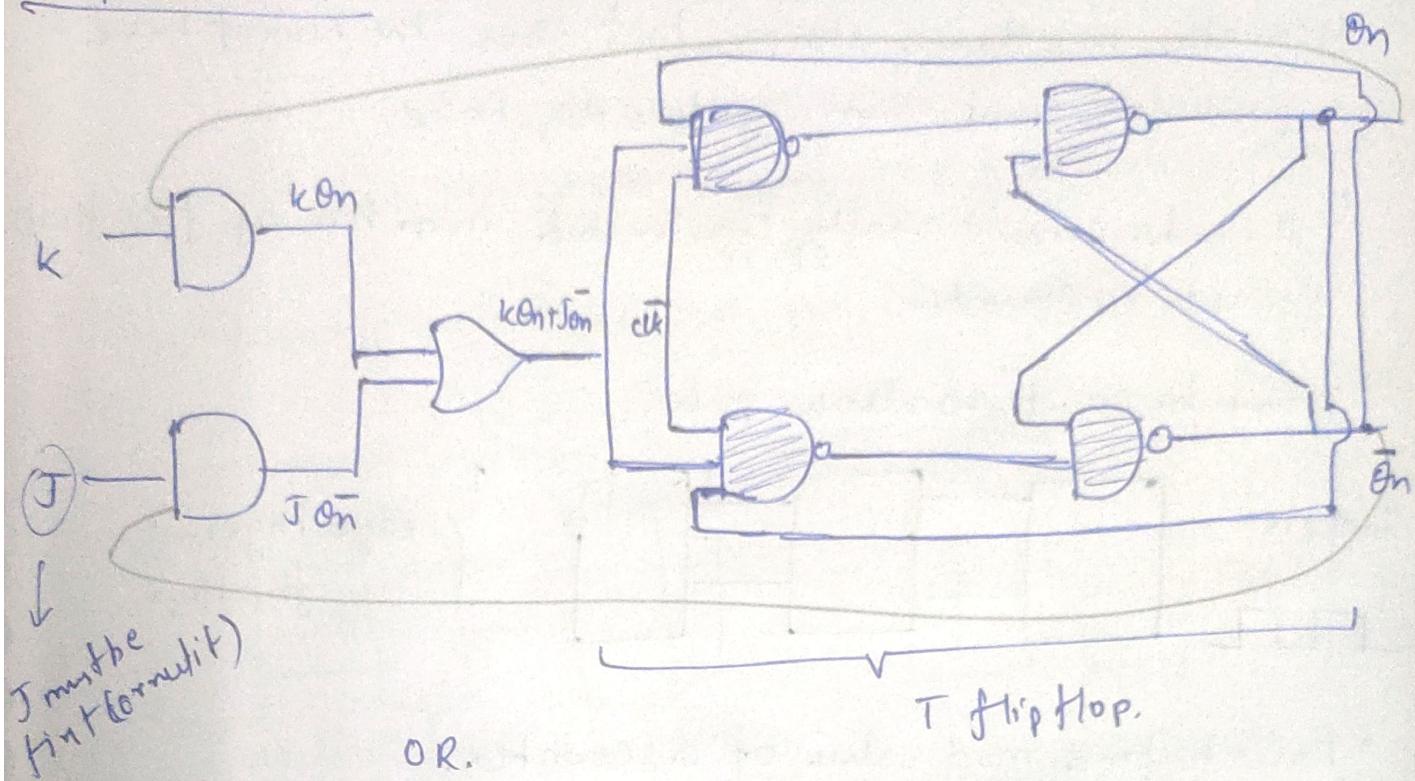
Given (T)			Required (JK)		
Qn	Qn+1	T	J	K	Qn+1
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	0
1	1	0	1	1	0

Rough  
Qn      Qn+1  
 J      K  
 0      0  
 0      0  
 1      0  
 1      1

J	K	Qn	Qn+1	T
0	0	0	0	0 → 0th
0	0	1	1	0 → 1st
0	1	0	0	0 → 2nd
0	1	1	0	1 → 3rd
1	0	0	1	1 → 4th
1	0	1	1	0 → 5th
1	1	0	1	1 → 6th
1	1	1	0	1 → 7th



## Implementation

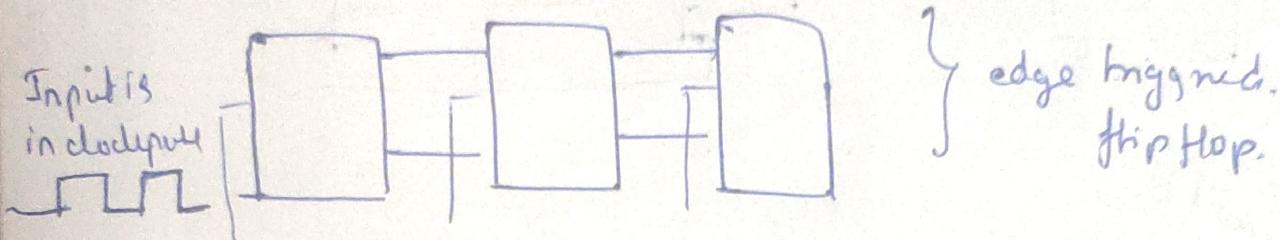


## \* Counters

4. Counter is a device stores that stores the No. of time particular count occur. Eg like AC, Fridge.

↳ a counter circuit usually constructed number of flip-flop connected in cascade

4 min to count no clean parts.



- How to find mod value of a counter,

CH<sub>1</sub> CH<sub>2</sub> CH<sub>3</sub> CH<sub>4</sub>  
 C<sub>2</sub>H<sub>5</sub> O - 1 - 2 - 3 - 4 ) 2CH<sub>3</sub>C<sub>2</sub>H<sub>5</sub>

1st No Different Stems. Mod 5 occurs.

Eg 2  $(1-2-3-7-5)$  Mod 8 counter.

- o No. of Flip-Flops required to design Mod-n counter.

$$\boxed{2^n \geq m} \quad \Rightarrow n = \text{No. of Flip Flops required.}$$

$$\alpha^3 \geq 5$$

No. of flip-flops = 3

or

$$\left[ \log_2 \right]$$

Eq 972

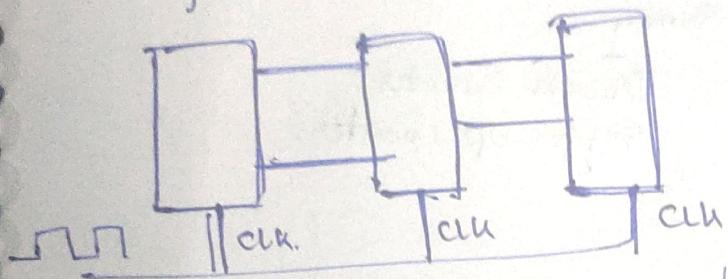
$$\lceil (\log_2 272) \rceil = \lceil 8 \rceil = 9$$

Ex 0-1-4 8-5, find No. of Flip Flops.

→ 4 of check No. of bits required given sequence)

\* Synchronous vs Asynchronous Counters.

Synchronous Counter.



perform action's simultaneously.  
all flip-flop are triggered simultaneously  
with single clock pulse

↳ Allow jumping sequences (Random sequence)  
i.e.,  $0 \rightarrow 4 \rightarrow 2 \rightarrow 6 \rightarrow 3$

↳ Perform fast bit computations.

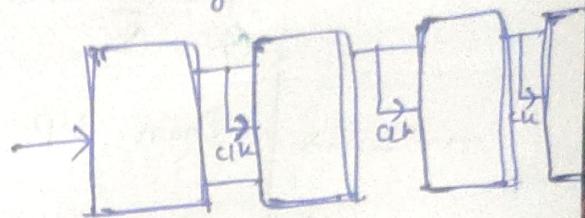
↳ Design is complex as requires  
extra gates for jumping in sequence b  
or to reach at random sequence.

↳ cost is high as it has high  
computational power & complex design.

↳ less delay as compared to asyn..

↳ Ring, Twisted-Ring counter.  
(Johnson counter).

Asynchronous Counter.



skewless triggering of  
asynchronous flip-flop  
clock depend upon previous flip-flop  
output.

up Asynchronous counter.

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

up down Asynchronous counter

$4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$

↳ Allow up sequentially  
or down sequentially

↳ slower compare to  
synchronous counter.

↳ low cost as  
compared to Synchron.

↳ greater delay as  
compared to Synchron.

↳ Ripple counters.

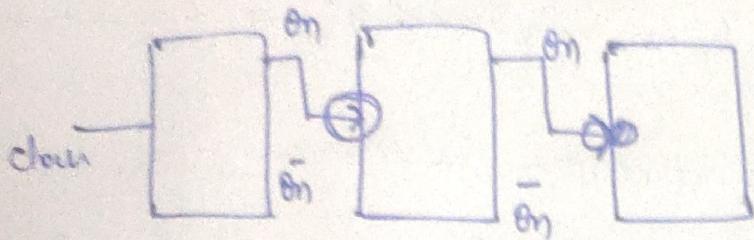
## Asynchronous Counter Types

Up Asy. Counter

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 3$$

Down Asy. Counter

$$3 \rightarrow 2 \rightarrow 1 \rightarrow 0.$$

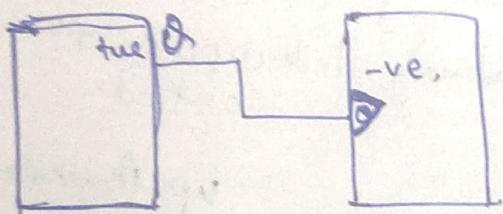


	True	Not True	
True	Down	Up	
Not True	Up	Down	

Summ

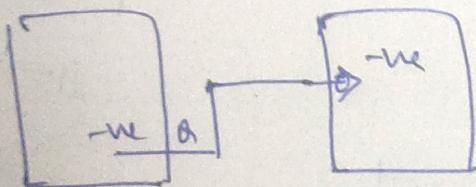
↓  
Down Counter  
Diff. = up Counter.

Ex 1



true  $\rightarrow$  not true {up counter}

Ex 2



not true  $\rightarrow$  not true {down counter}

• Design Asynchronous Counters

$$\rightarrow 0 - 1 - 2 - 3$$

max No = 3 ie 11

∴ 2 flip  
flop required

Present state

M3B	$Q_1$	$Q_0$
$Q_1$	0	0
0	0	1
1	0	0
1	1	1

Next state.

$Q_1 +$	$Q_0 +$	$Q_0$
0	1	1
1	1	0
0	0	0
1	0	1

$D_1$	$D_0$
0	1
1	1
0	0
1	0

↓ required to HII

$D$  flip flop E.T. checks to calculate  $D$

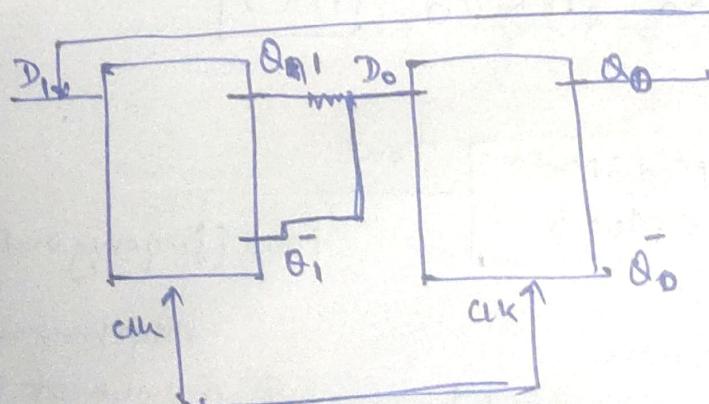
$D_m$	$B_{m+1}$	$D$
0	0	0
0	1	1
1	0	0
1	1	1

or K-map (SOP) of  $Q_1, Q_0$ .

$$\text{For } D_1 = \begin{array}{|c|c|c|} \hline Q_1 & Q_0 & \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = Q_0$$

$$\text{For } D_0 = \begin{array}{|c|c|c|} \hline Q_1 & Q_0 & \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \bar{Q}_1$$

Design Implementation [ 2 flip from present state sequence given in question ]



## Counters Basic

$$2^n \geq N$$

$n = \text{no. of flip flops}$   
 $N = \text{no. of states.}$

Eg Mod-5 counter

$$2^n \geq 5$$

$$8 \geq 5$$

$\therefore n = 3$  flip flop required.

Eg (3) Bit counter.

↳ when bit is given <sup>in</sup> question { as each flip-flop can store <sup>1</sup> single bit }

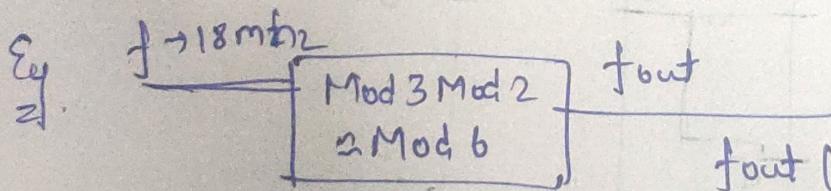
$$\text{No. of flip flops}(n) = 3.$$

$$2^3 \geq 8 \quad n = 3.$$

Eg Mod 5 Mod 4 counter

≈ Mod 20 counter.

$$N = 20, 2^n \geq 20, \boxed{n = 5}$$



$$f_{\text{out}} (\text{frequency out}) = \frac{18}{6} = 3 \text{ MHz}$$

Synchronous.

① all flip-flop are connected with single clock.

② They are faster.

③  $T_{ck} > T_p \rightarrow$  propagation delay of flip flop.

④ any random sequence possi.

Asynchronous.

① output of single flip-flop are connected with different clock (Op of one flip flop connected with another flip flop (next))

② Slower.

③  $T_{ckL} > (n) T_p \rightarrow$  no. of flip flop.

④ only fixed sequence is possible.

Q1 design Asynchronous divide by 8 up counter.  
Draw its timing waveform.

Part-1  
①

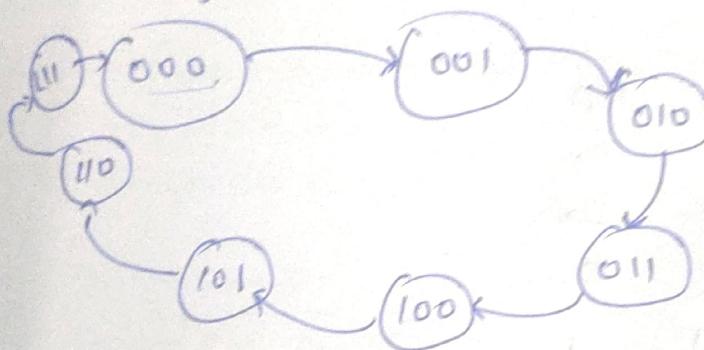
Mod-8

$2^n \geq 8$

$n=3$

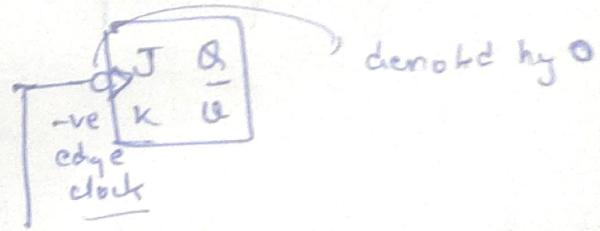
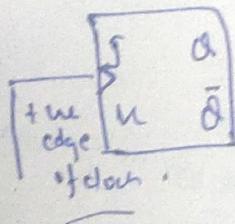
No. of flip-flops required = 3.

② State diagram.

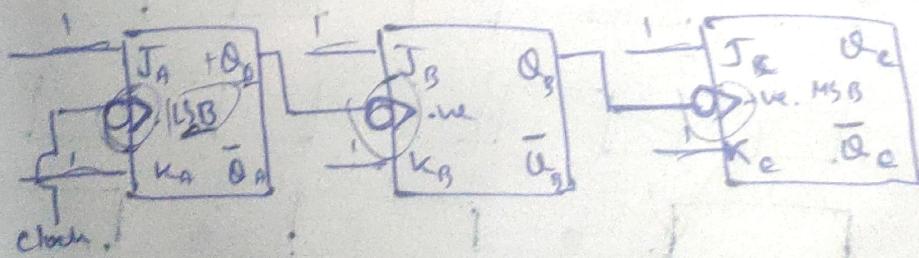


③ Asynchronous counter. only design by using JK, T flip-flops  
fact 1 only when  $[J=1, K=1]$   $[T=1]$  also require toggling action

fact 2 In order to design up counters.  $Q \rightarrow -ve$  edge of clock.  
 $\bar{Q} \rightarrow +ve$  edge of clock



Step-4 Circuit diagram.



Step-5 The flip-flop with first clock at act LSB.

- Step-6
- ①  $Q_A$  toggles when -ve edge of clock applied.
  - ②  $Q_B$  toggles when  $Q_A$  changes from 0 to 1
  - ③  $Q_C$  toggles when  $Q_B$  changes from 1 to 0
- MSB      LSB  
0      1

Step-7 Truth Table.

clk     $Q_A$      $Q_B$

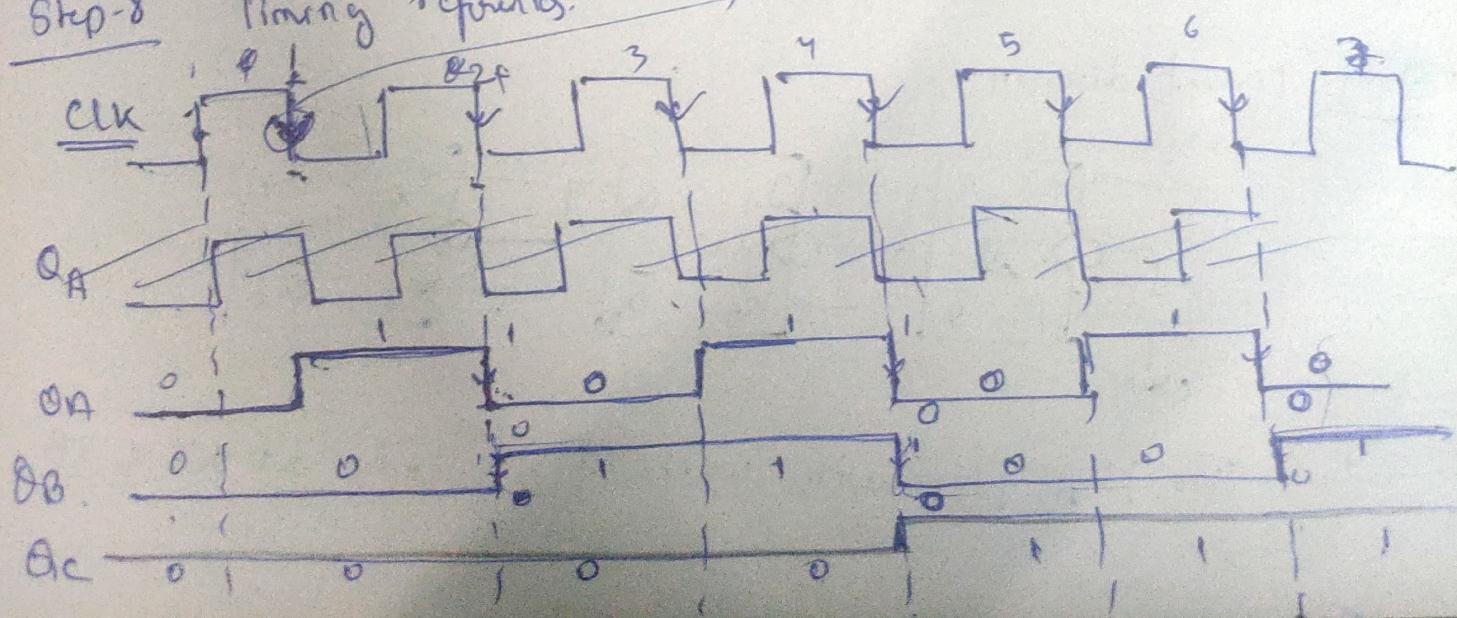
clk	$Q_C$	$Q_B$	$Q_A$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

$Q_A$  only toggles with -ve edge of clock

Step-8

Timing references.

→



Ques-2 Design divide by 8 Asynchronous <sup>down.</sup> counter & draw timing wave-form?

Ans-2

Step1:

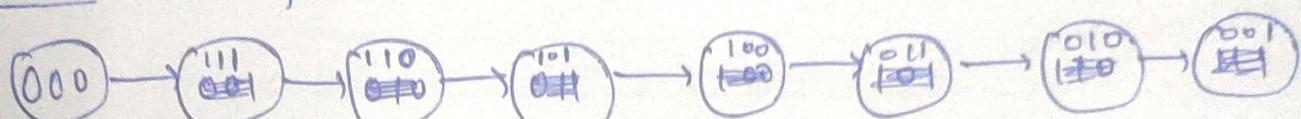
$$2^n \geq 8$$

$$2^3 \geq 8$$

$$n = 3.$$

3 No. of Flipflop reqd.

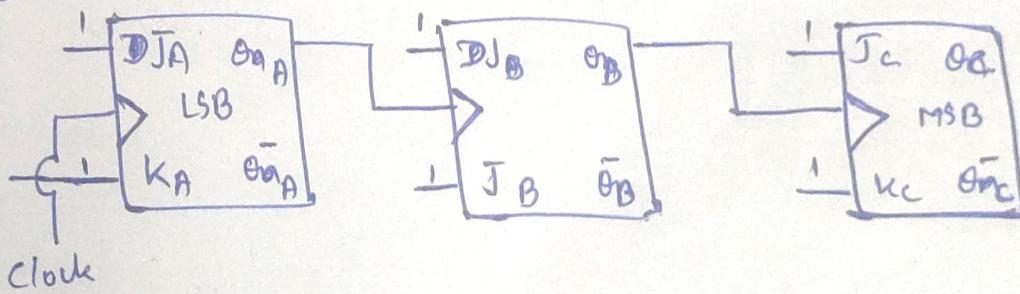
Step2: State diagram.



Step3: J, K, T must be 1 for toggling action.

Step4: Design flip flop. +ve and -ve on clock and -ve and +ve.

Step5: Design circuit



The very first flip-flop where first clock applied act on LSB.

Step6: ① Q<sub>A</sub> toggles at edge of clock applied. (+ve pulse).

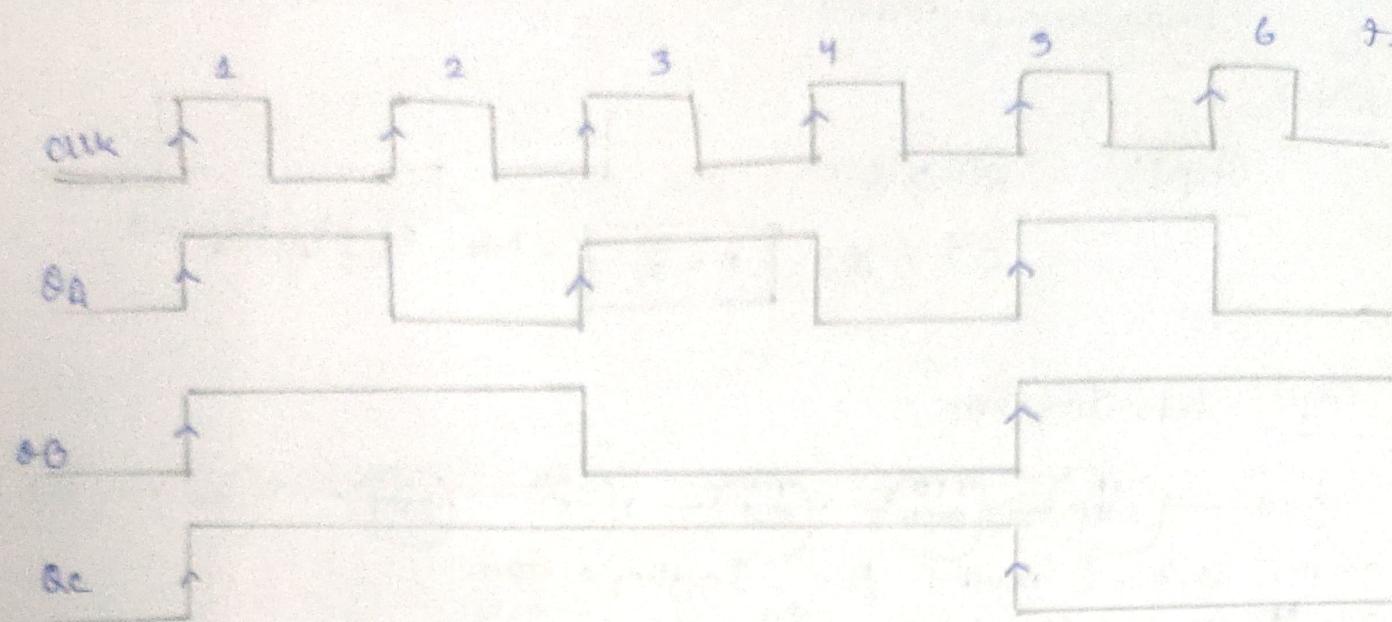
② Q<sub>B</sub> toggles when Q<sub>A</sub> changes 0 to 1. (+ve) pulse.

③ Q<sub>C</sub> toggles when Q<sub>B</sub> changes 0 to 1 (-ve) pulse.

Step7: Truth Table.

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

Step 8: Timing wave forms.



## \* Synchronous Counter

Ques Design a mod-8 Synchronous up Counter using Jk flip flop?

Ans Step1: Determine flip flop needed.

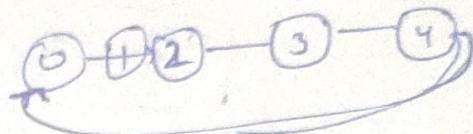
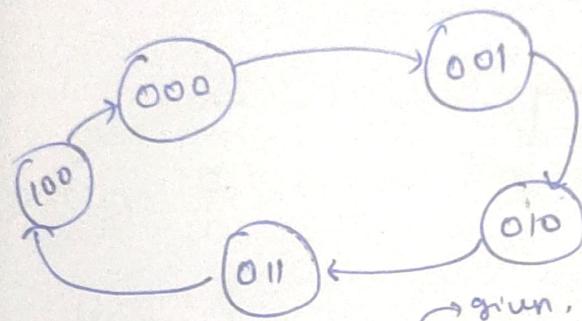
$$2^n > S$$

$$n = \text{No. of flipflop}$$

$$g = \text{No. of states.}$$

3 bit.  
no. of flipflops  
 $N=8$

Step2: State Diagram.



Step3! Write ET of Jk flip flop.

On Qn+1	J	K
0 0	0	X
0 1	1	X
1 0	X	1
1 1	X	0

Step4: Exkstetion table for counter state.

Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>A+1</sub>	Q <sub>B+1</sub>	Q <sub>C+1</sub>	J <sub>A</sub> K <sub>A</sub>	J <sub>B</sub> K <sub>B</sub>	J <sub>C</sub> K <sub>C</sub>
0	0	0	0	0	1	0 X	0 X	1 X
0	0	1	0	0	0	0 X	0 X	0 X 1
0	1	0	0	1	1	0 X	X 0 1	1 X
0	1	1	1	0	0	1 X	X 1 X	1
1	0	0	0	0	0	X 1	0 X 0	X
1	0	1	X	X	X	X X	X X X X	
1	1	0	X	X	X	X X	X X X X	
1	1	1	X	X	X	X X	X X X X	

Step 6: Only consider variable for creating k-map (SOP),  
i.e.,  $Q_A Q_B Q_C$

$J_A = \underline{Q_A Q_B Q_C}$

$Q_A$	00	01	11	10
0	X	X	(X)	X
1				

$$\boxed{J_A = Q_B Q_C}$$

$K_A = \underline{Q_A Q_B Q_C}$

$Q_A$	00	01	11	10
0	X	X	X	X
1	X	X	X	X

$$\boxed{K_A = 1}$$

$J_B = \underline{Q_A Q_B Q_C}$

$Q_A$	00	01	11	10
0	(X)	X	X	X
1	X	X	X	X

$$\boxed{J_B = Q_C}$$

$K_B = \underline{Q_A Q_B Q_C}$

$Q_A$	00	01	11	10
0	X	(X)	(X)	X
1	X	X	X	X

$$\boxed{K_B = \underline{Q_B Q_C}}$$

$J_C = \underline{Q_A Q_B Q_C}$

$Q_A$	00	01	11	10
0	(X)	X	X	(X)
1	X	X	X	X

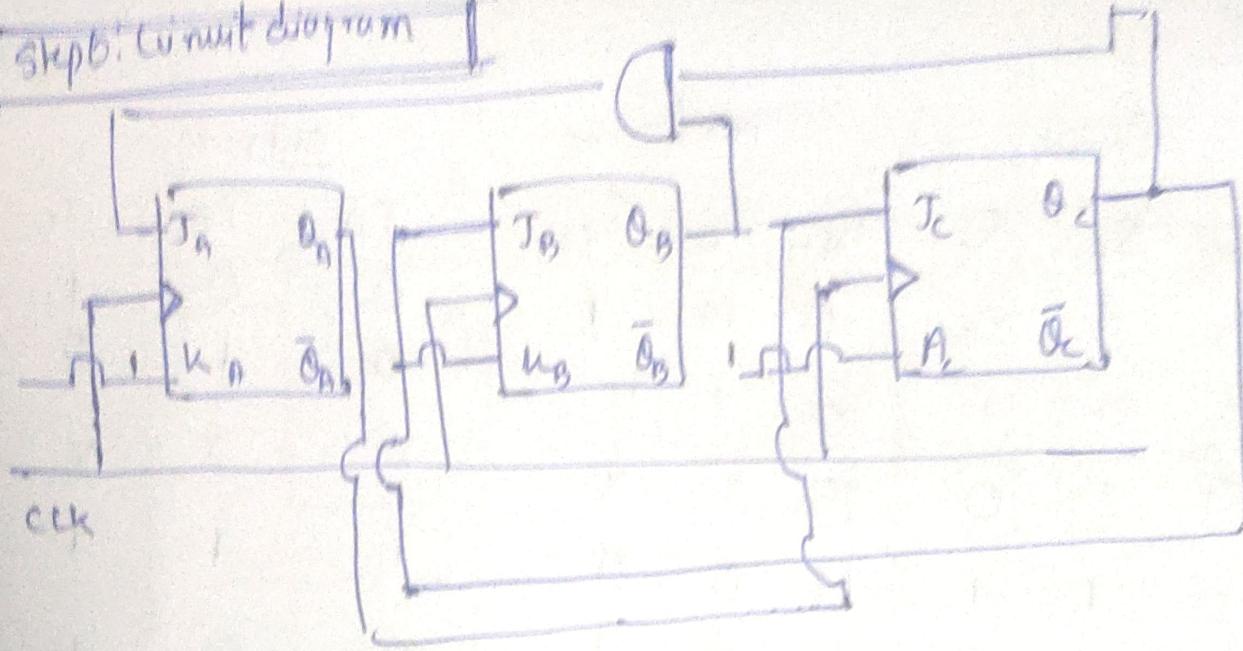
$$\boxed{J_C = Q_A}$$

$K_C = \underline{Q_A Q_B Q_C}$

$Q_A$	00	01	11	10
0	(X)	(X)	(X)	(X)
1	X	X	X	X

$$K_C = 1$$

Step 1: Circuit diagram



Ques 2 Design Synchronous Counter for sequence.

$$0 \rightarrow 01 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 0$$

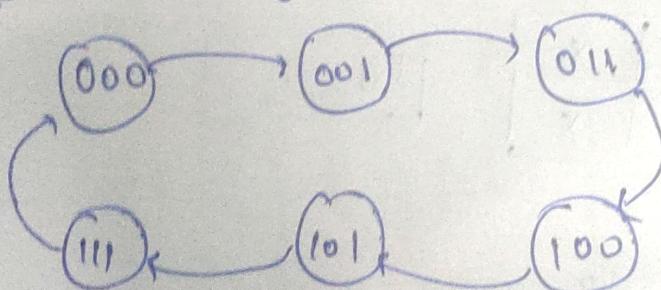
using T Flip Flop?

Ans 2 Step 1: Given for sequence find highest No. i.e. 7  
then mod = 7 + 1 = 8.

$$2^n > 8 \\ n=3$$

No. of flip flops = 3.  
No. of states = 8

Step 2: State diagram



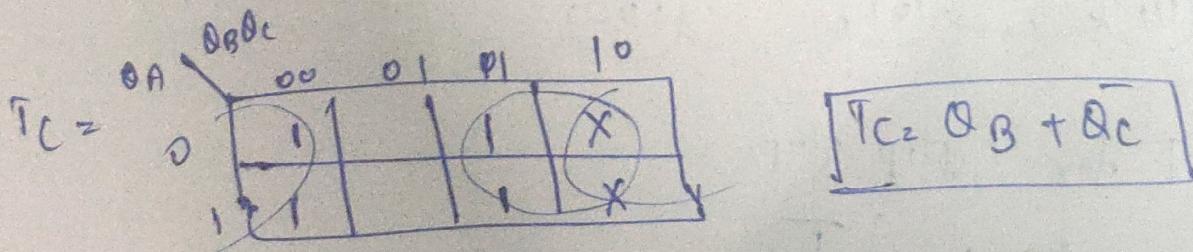
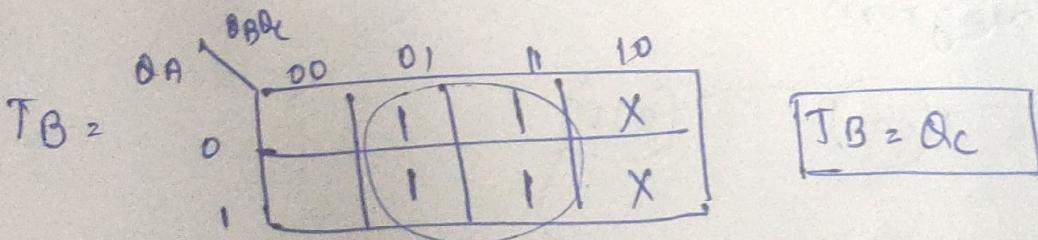
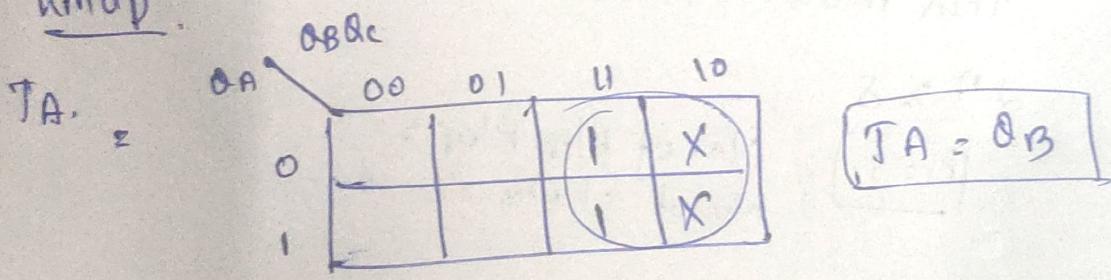
Step 3: E-T for 7 flip flop.

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

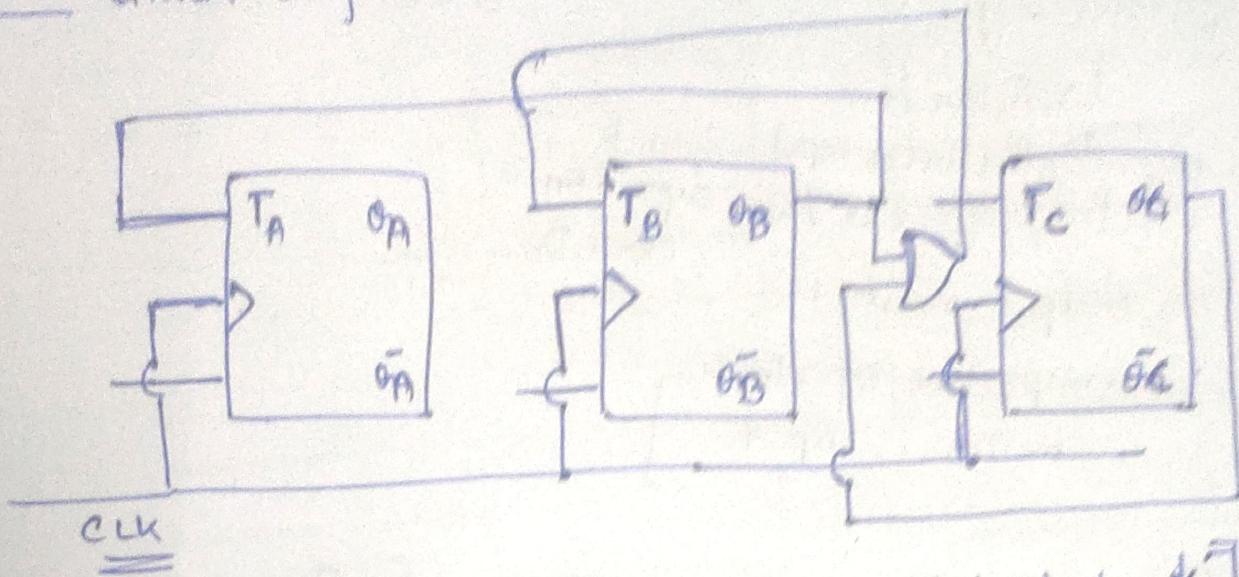
Step 4:

Present			next			Output		Tc.
QA	QB	QC	QA+1	QB+1	QC+1	TA	TB	
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	X	X	0
0	1	0	X	X	X	1	1	1
0	1	1	1	0	0	1	0	1
1	0	0	1	0	1	0	0	0
1	0	1	1	1	1	0	1	0
1	1	0	X	X	X	X	X	X
1	1	1	0	0	0	1	1	1

Kmap



8kp5! 16bit diagram.



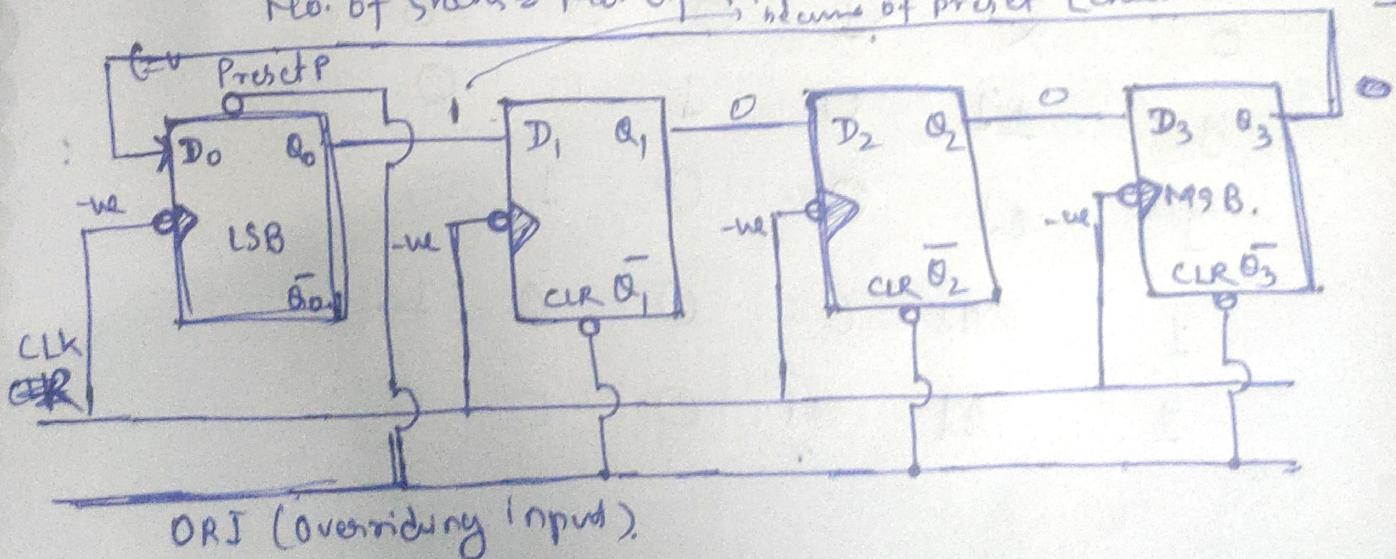
- \* Ring Counter. (4bit) → Eve. clock can supply to clock [ ]  
 → that means 4 No. of flip-flops  
 → D flipflop [ ] Same input  
 → Same output
- ① Application of shift Register.
  - ② Output of Last flip flop connected with first flip flop.

In normal any type of counter.

No. of states in 4 flip flops is  $2^4 = 16$  states.

But in Ring Counter No. of flip Flops is equal No. of states.

or  
No. of states = No. of flip flops used.  
↳ because of project [check otherwise]



This Counter fall in category in Synchronous counter as all flip-flops are connected with single clock.

It is fully a counter or as well not Shift Register.

④ (a) Preset I/p = 0.  $Q = 1$

↳ Active low

↳ Asynchronous input  $\rightarrow$  doesn't depend on clock,  $D_0$ .

Output must 1

Irrespective of clock or  $D_0 \dots$  input

(b) CLR I/p. is 0 then  $Q = 0$

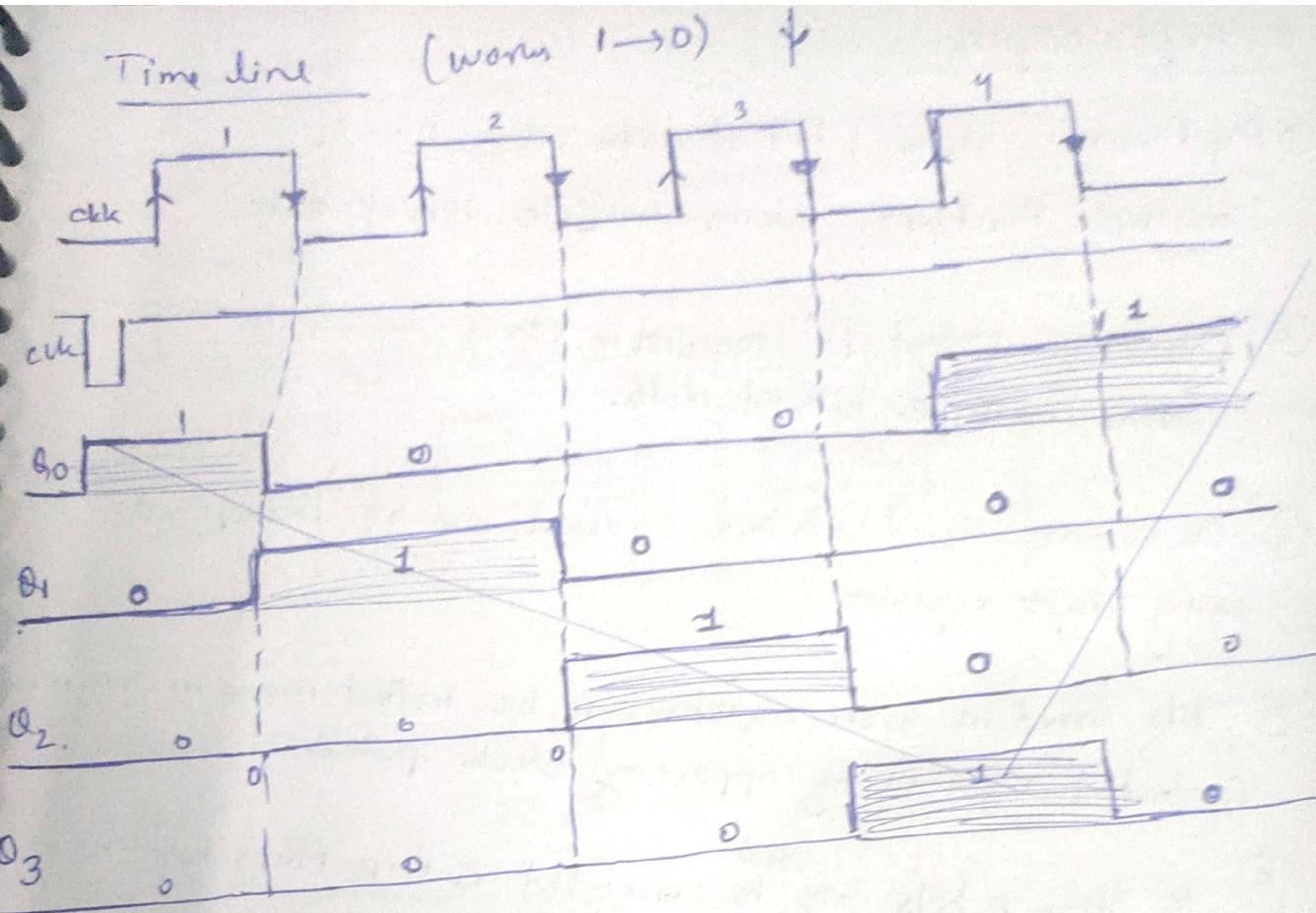
↳ Active low signals.

↳ Asynchronous.

Eg Preset is 0 then output first flip flop

$Q_0 = 0$  and when CLR input is 1 Then  $Q_2, Q_3, Q_4$  is equal 0 as they are connected

ORI	CLK	Preset.			
		$Q_0$	$Q_1$	$Q_2$	$Q_3$
0	x	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	0A↓	0	0	0	1
1	↓	1	0	0	0



Right shift of 1 bit from observation

work on low signal where, clear pulse is from.  
 $(1 \rightarrow 0)$  only. (How to get know because of  sign.

## \* Shift Register

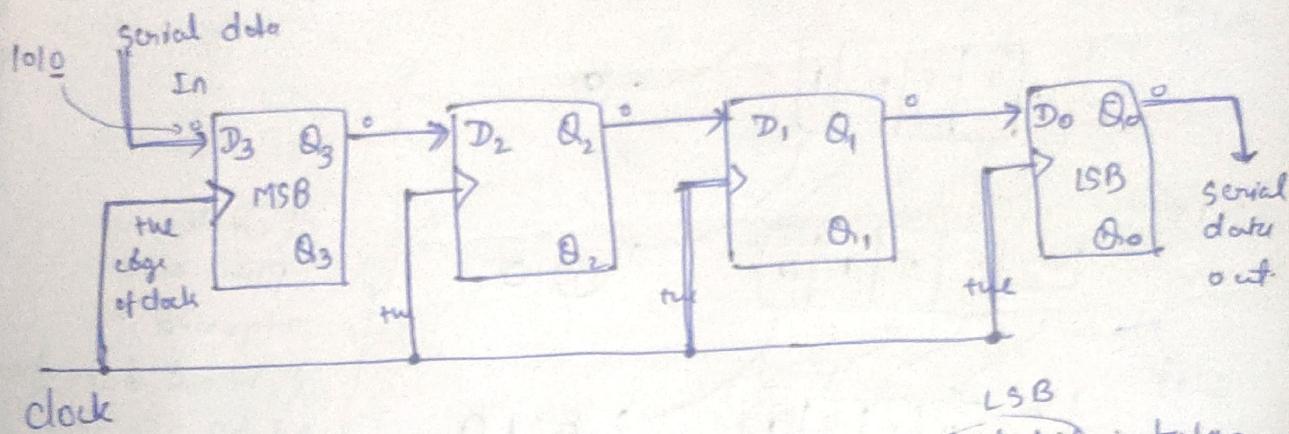
- ① Flip Flops: stores 1 bit of data each 0 or 1.
- multiple Flip Flops: stores multiple bits of data.
- ② Registers: group of FF connected in series → used to store multiple bits of data.
- ③ The inform. stored with this registered can be transferred using shift register.
- ④ Bits stored in such registers can be transferred more in registers (in/out of register) by applying clock pulse.
- ⑤ To store n bits we <sup>need</sup> to connect n flip-flops together.

## Types of Shift Registers.

- ① SISO (serial in serial out)
- ② SIPO (serial in parallel out)
- ③ PIPO (parallel in parallel out)
- ④ PIPO
- ⑤ Universal Shift Register.

Q Serial In Serial out (SISO) [4 bit] No. of flipflops = 4.

winged flip flop (To input With output) from Truth table.  
J D flip flop.



Data: 1010 MSB      LSB

& when to perform right first bit in taken input from left side.

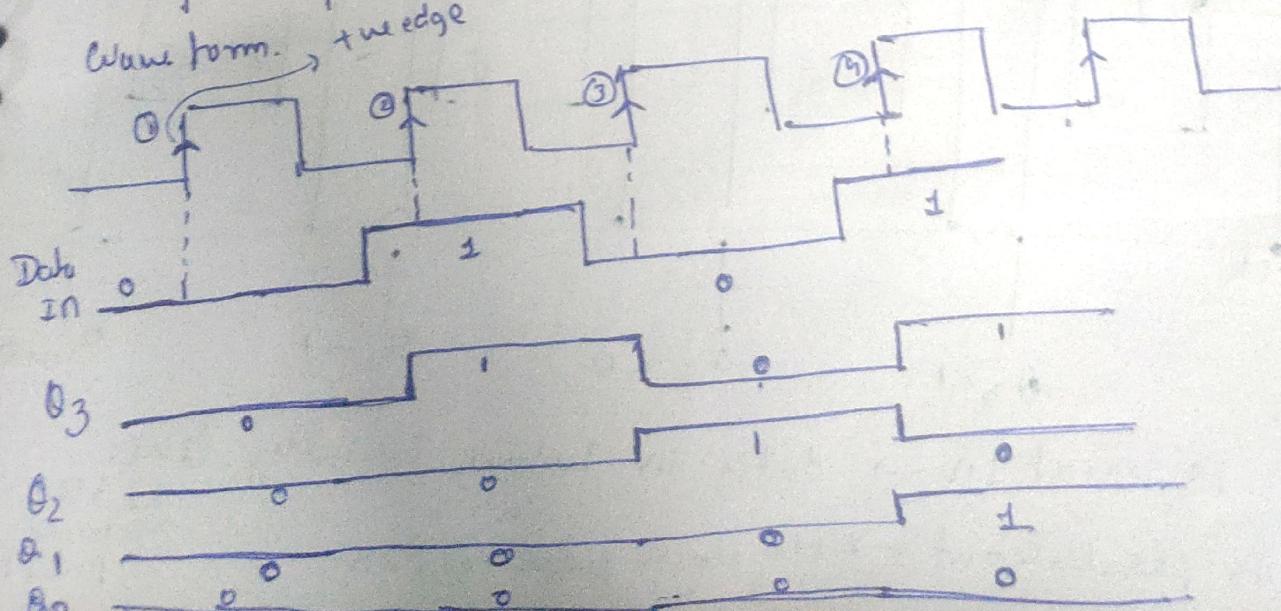
Truth Table

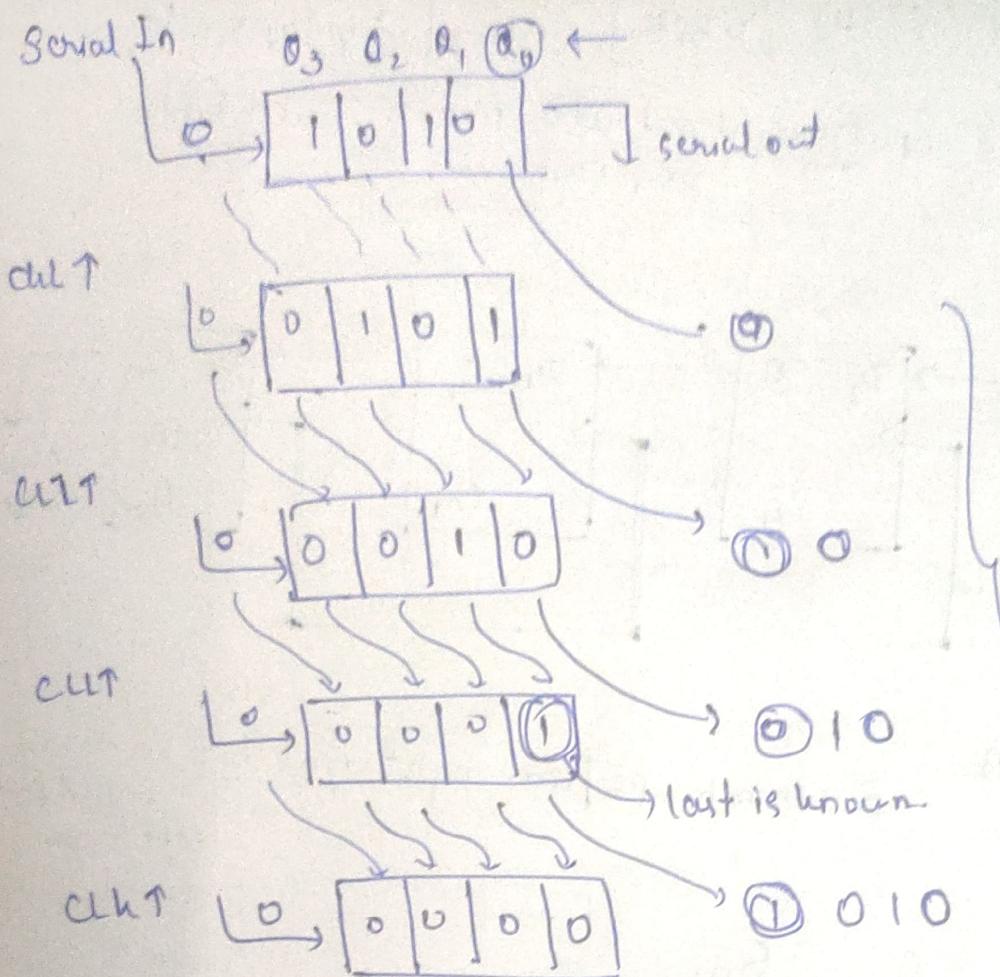
clk	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
X	0	0	0	0
↑	0	0	0	0
↑	1	0	0	0
↑	0	1	0	0
↑	1	0	1	0

Truth D  
clk D J  
0 0  
1 1

In order to enter data serial<sup>4</sup>  
No. of clock pulse required = 4 clock pulses of 4 bit SISO.

Wave form. <sup>true edge</sup>



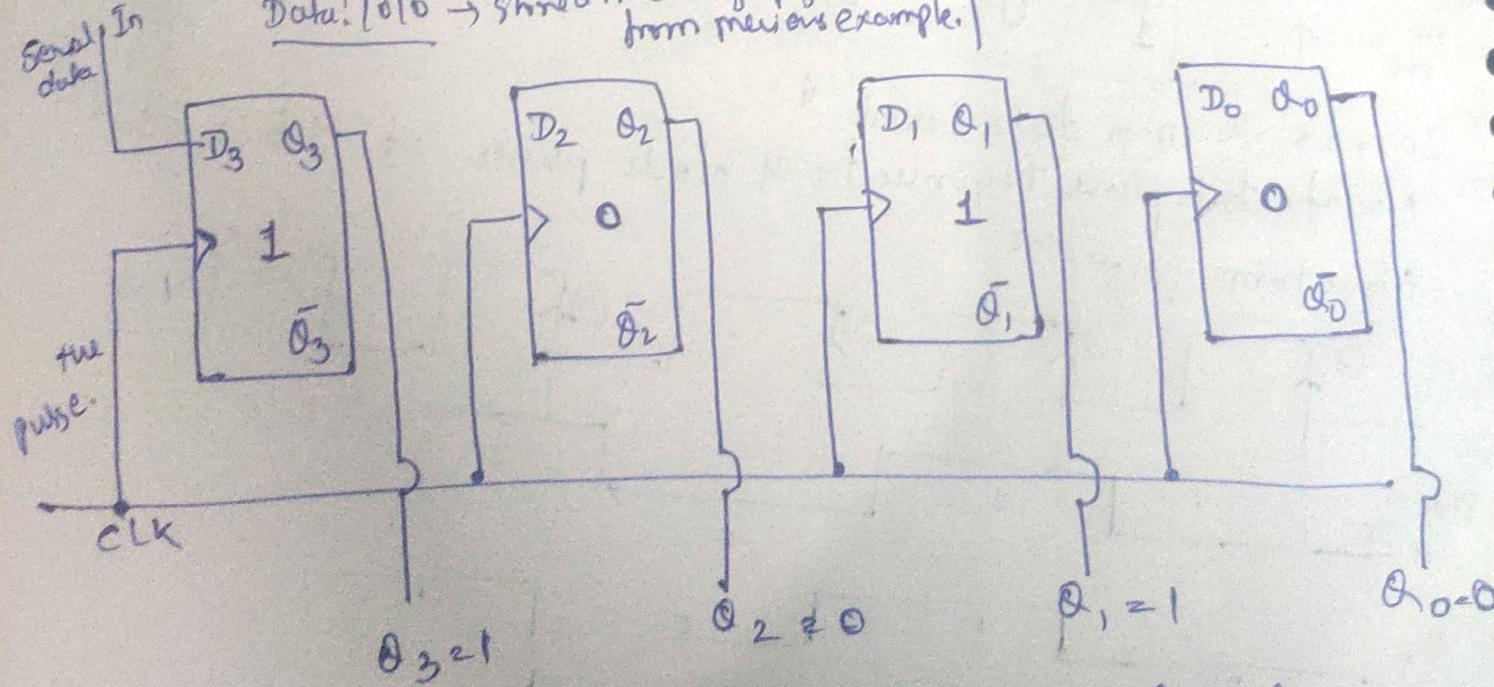


3 clock pulses are required to take out data from SISO register i.e.,  $(n-1)$  clock pulses

How to take out data from shift register by using input from serial In.

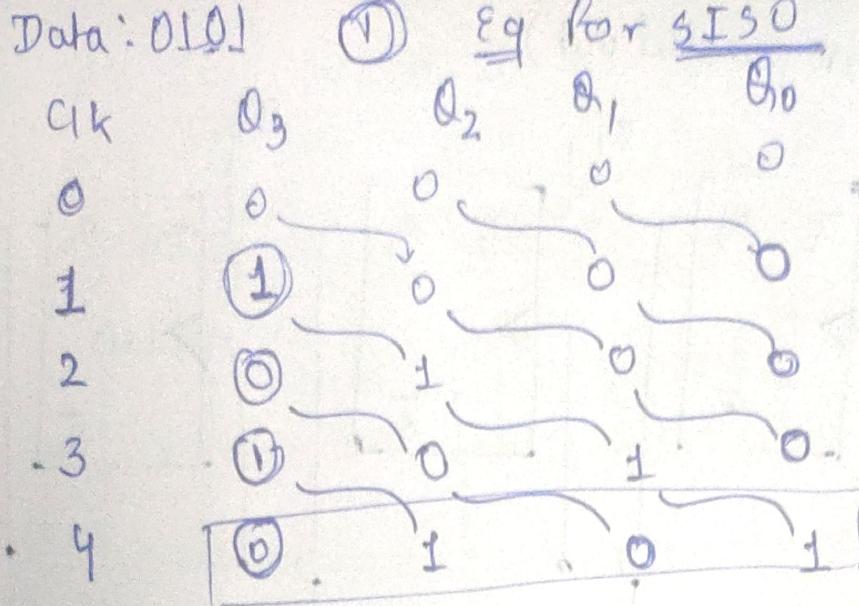
② Serial In parallel out Register (D flip flop)

In parallel out Register. (D flip flop)  
 Data: 1010 → stored in each flip flop.  
 from previous example. To input volt output (clock  
 must active)

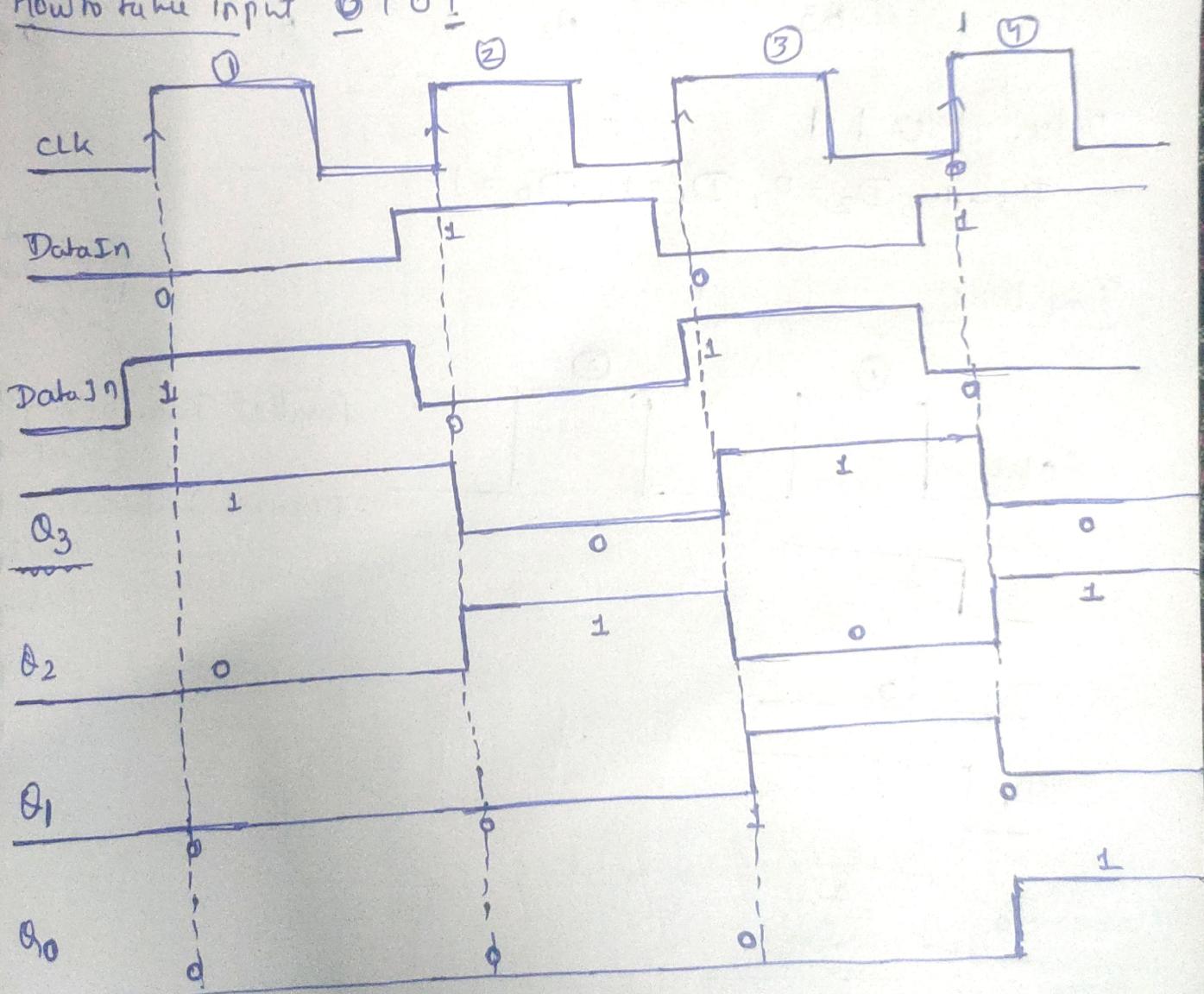


As Serial In  $\rightarrow$  data is taken serially like  
S.I.  
As parallel out  $\rightarrow$  no clock pulse needed.

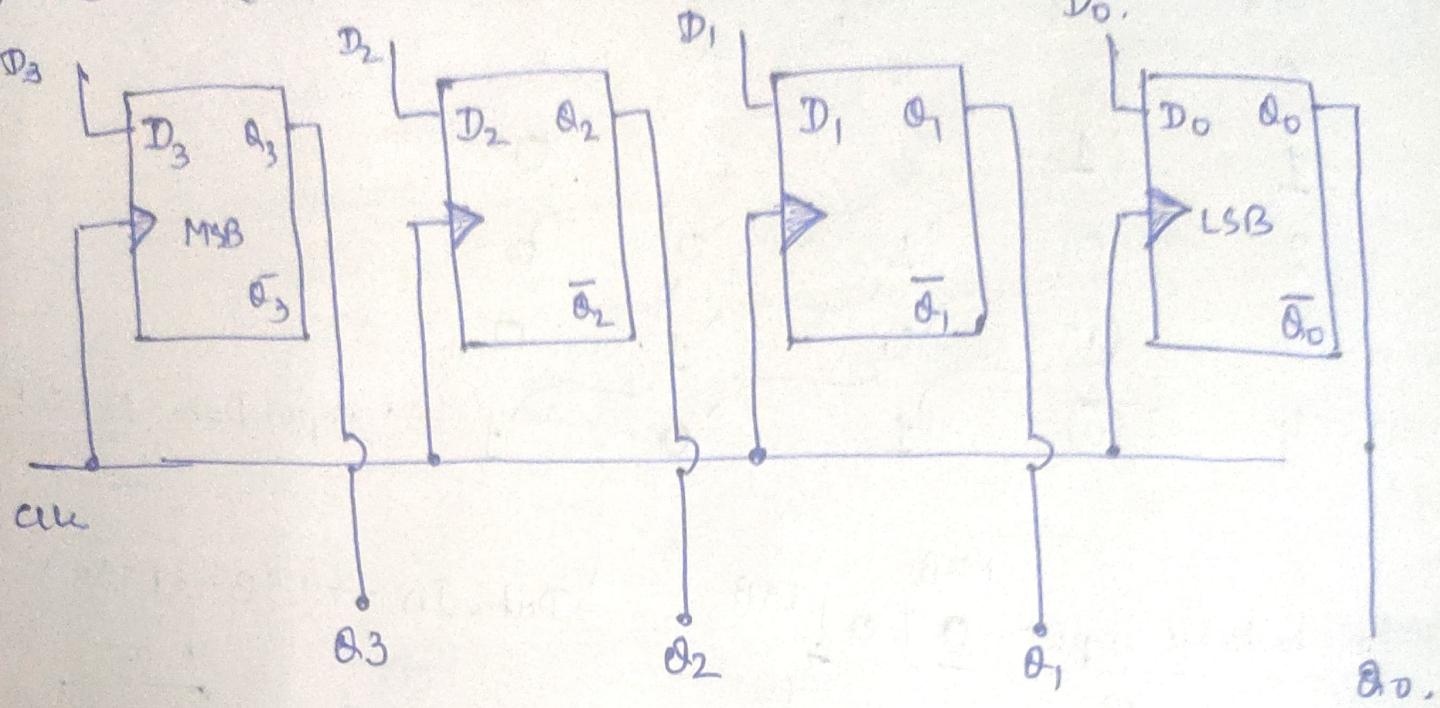
SIS 50.  
4 o'clock  
Puh Nedd



How to take input    MSB.    Data In  $\rightarrow$  (LSB  $\rightarrow$  MSB)



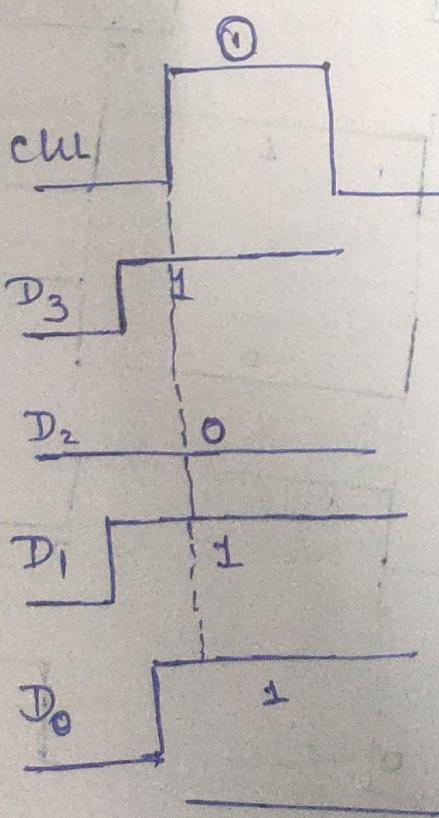
### ③ PIPD (parallel in parallel out)



Data: 1 0 1 1

$$D_3 = 1, D_2 = 0, D_1 = 1, D_0 = 1$$

#### Timeline



Required to store data from bus

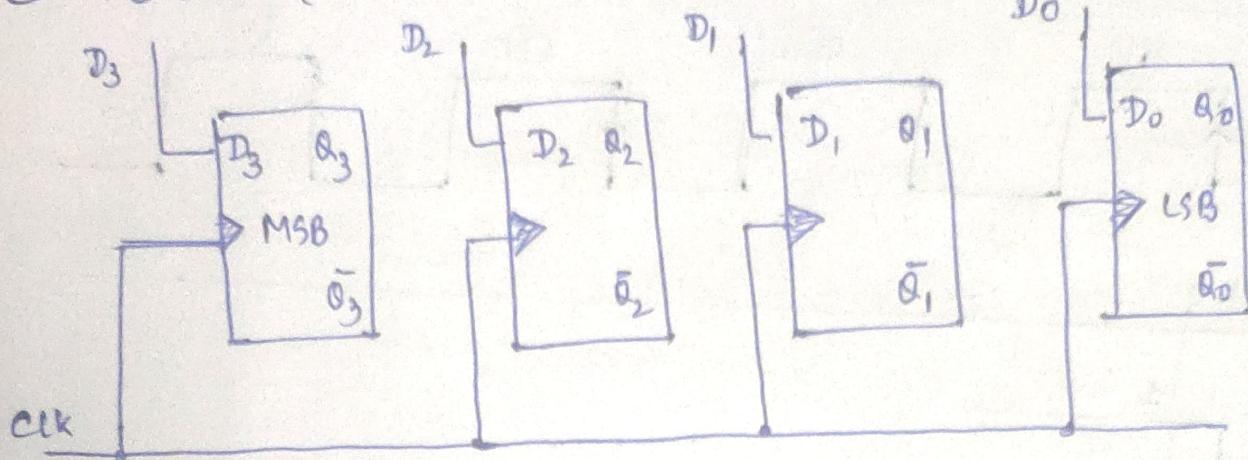
Parallel Data In = single clock pulse.

Parallel Data out = clock pulse.

1 0 1 1 } Data is reached

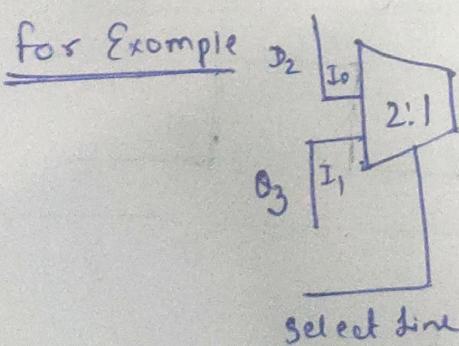
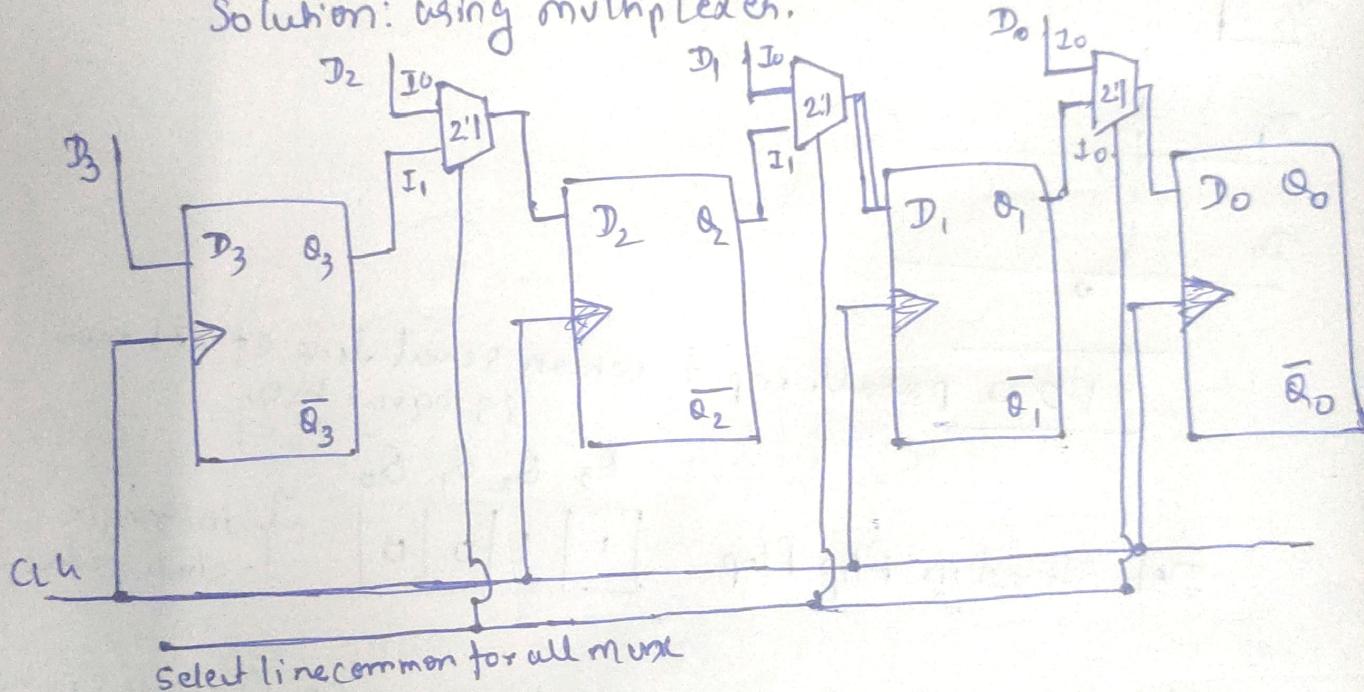
{ in single clock pulse.

# Q) PI SO (Parallel In and Serial Out).



But how to take Serial out from  $Q_3$  to  $D_2$

Solution: using multiplexer.



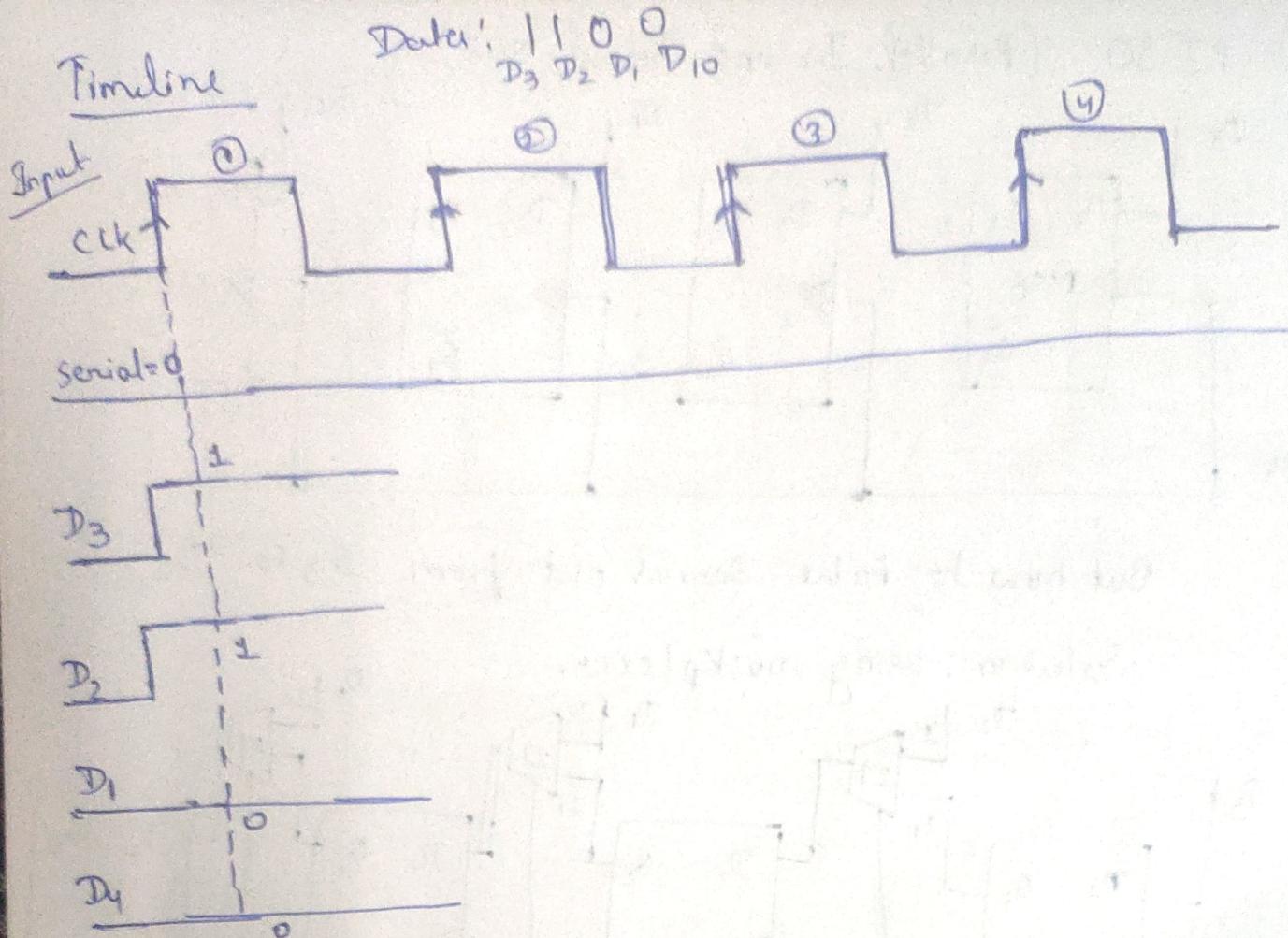
when Select line = 0 (I0 active means)  
(serial In) parallel input

when Select line = 1 (I1 active)  
(Serial out) means serial input  
output

No. of clock pulse required for Input 1

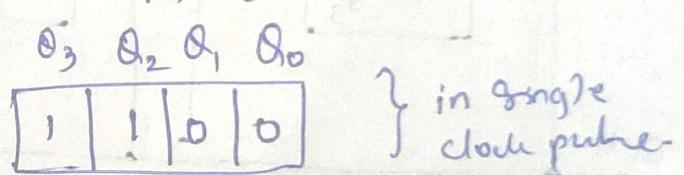
No. of clock pulse Required for output =  $n-1$  (can achieve

by shifting  
from left to right)



1100 parallel input when serial line of 2x1 min is equal to 0.

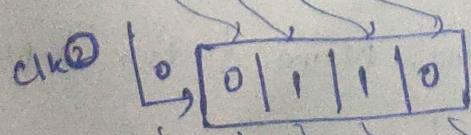
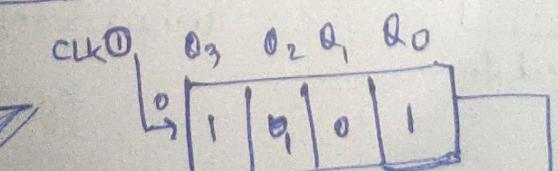
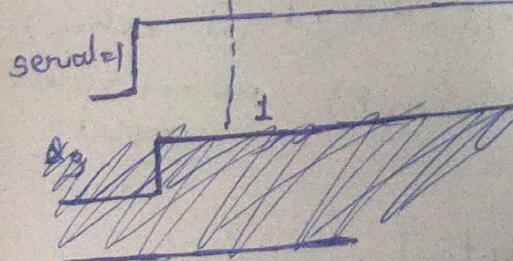
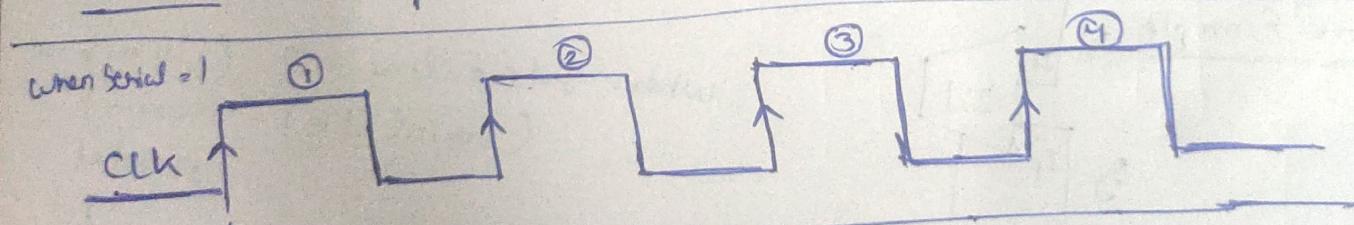
Data stored in Flip-Flop.



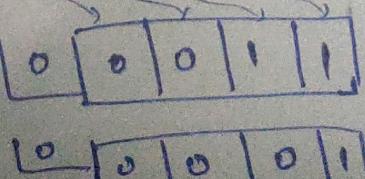
11001

Output

output of can achieve using  $(n-1)$  clocks.pulse



clk③



-1

①1

②0 1 → 0 1 1

# Design & Implement

Mod-10 Decade Asynchronous Count?

M

(N)

No. of states = 10.

Step 1!

$$2^n \geq M$$

$$n = 4$$

No. of flip flop Required.

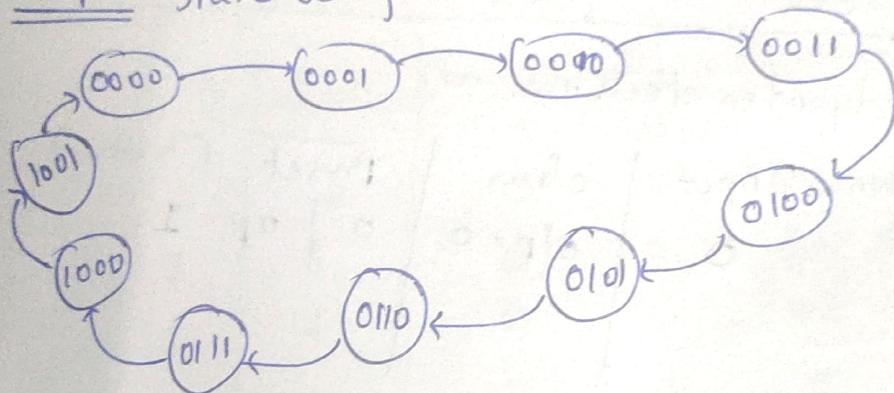
can be become  
from JK, D

asynchronous flip flop.

from state diagram

as well as  
circuit diagram.

Step 2: State diagram



Step 3:  $T = J = k = 1$

because Toggling action is required.

Step 4: from 0000  $\rightarrow$  10 } up sequence by default

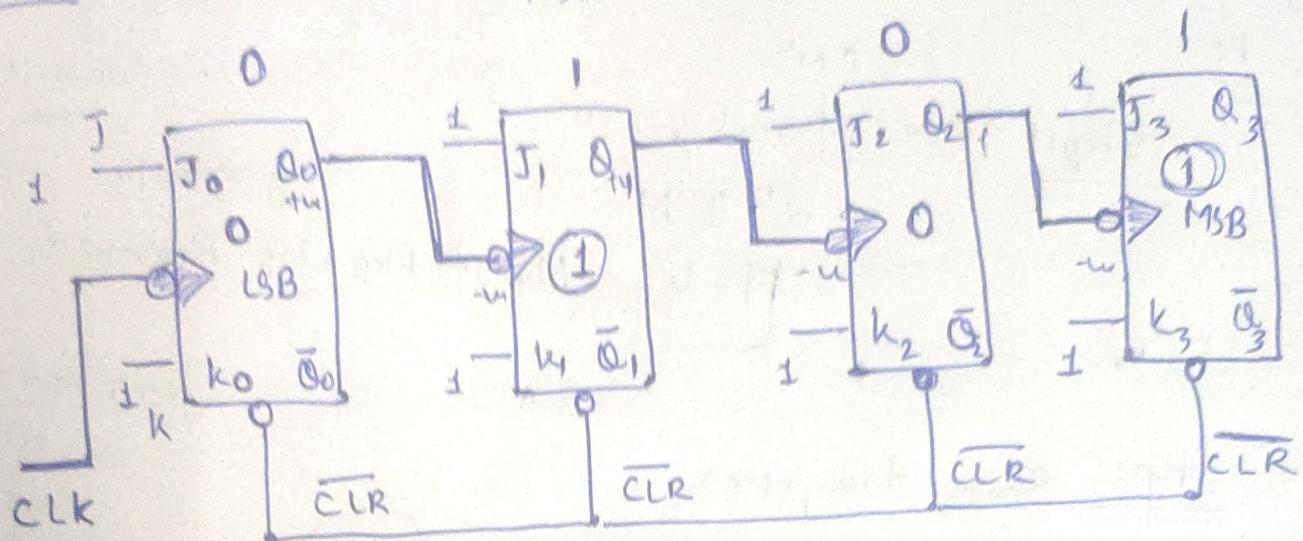
$\bar{Q}$  terminal  $\rightarrow$  one terminal S LK

$\bar{Q}$  terminal  $\rightarrow$  the terminal S LK

In order to design Mod-10 Asynchronous Up Counter.  
we required more two condition

Step 5

(1 0 1 0)<sub>LSB</sub> (reduct) (0 0 0 0)

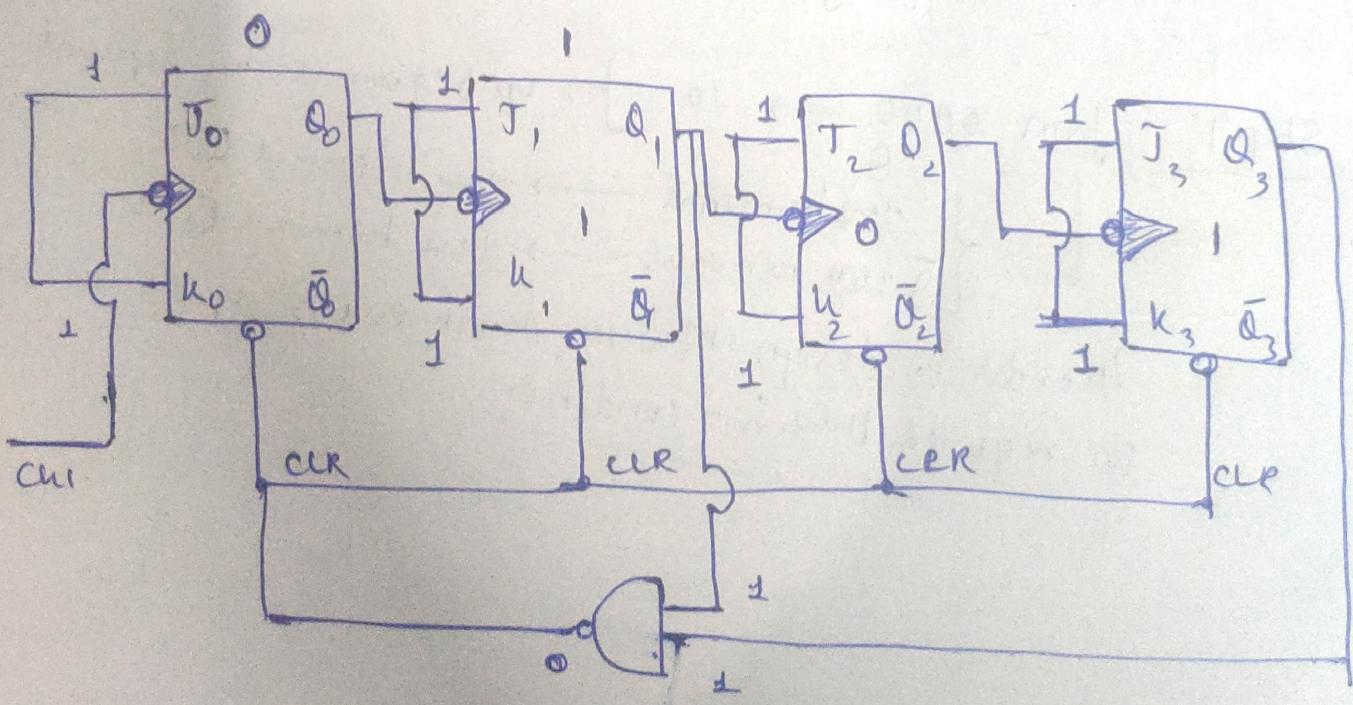


J-K flip-flops depend on clock  $t_{w}$ ,  $-w$ .

but, The Asynchronous Input  
doesn't depend  
on clock  $t_{w}$

0	clear	Preset (active)
	$op = 0$	
0	$op = 1$	

make marked as 0



shortest  $\rightarrow$  MSB always return in left  
LSB always return in Right.

Step - 6  $\overline{Q_1, Q_3}$   $\overline{CLK}$   
on every low signal (i.e.) signal  $Q_0$  toggles.  
 $Q_1$  Toggles when  $Q_0$  change  $1 \rightarrow 0$

$Q_2$  Toggles when  $Q_1$  change  $1 \rightarrow 0$

$Q_3$  Toggles when  $Q_2$  changes  $1 \rightarrow 0$

$\downarrow^1$   
 $0$   
 $1010$

<u>Step - 7</u> $\overline{Q_1, Q_3}$		MSB	LSB			
CLR	CLK	$Q_3$	$Q_2$	$Q_1$	$Q_0$	
0	X	0	0	0	0	$\rightarrow$ on
1	$\downarrow$	0	0	0	1	$\rightarrow$ 1st
1	$\downarrow$	0	0	1	0	$\rightarrow$ 2nd
1	$\downarrow$	0	0	01	1	$\rightarrow$ 3rd
1	$\downarrow$	0	1	0	0	$\rightarrow$ 4th
1	$\downarrow$	0	1	0	1	$\rightarrow$ 5th
1	$\downarrow$	0	1	1	0	$\rightarrow$ 6th
1	$\downarrow$	0	1	1	1	$\rightarrow$ 7th
1	$\downarrow$	0	1	0	0	$\rightarrow$ 8th
1	$\downarrow$	1	0	0	1	$\rightarrow$ 9th
1	$\downarrow$	1	0	1	0	$\rightarrow$ 10th
0	(X)	0	0	0	0	This o/p will Reset

Impat  $\downarrow$

0 became.

two input  
from  $Q_1$  and  $Q_3$   
are 1's.

irrespective  
clock.

of flip flop,  
when clock became all outputs are 0.  
irrespective of clock.

Q1 Design Mod-8 Synchronous counter.

using JK flip flop?

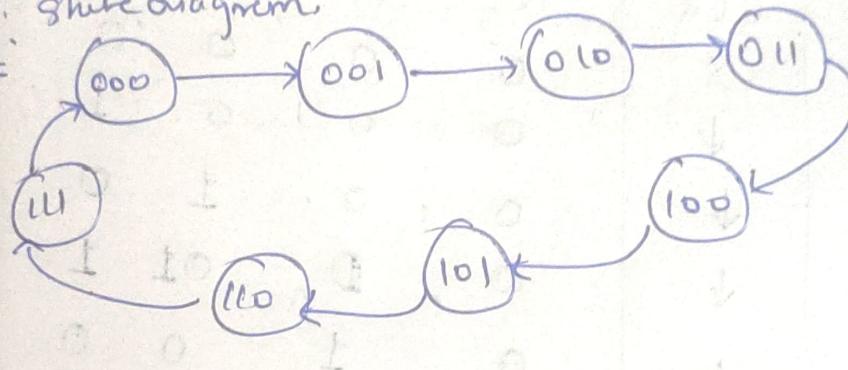
Ans Step1: No. of states = 8

$$2^n > 8$$

$$\boxed{n=3}$$

∴ No. of flip flops = 3.

Step2: State diagram



Step3: Write ET of the flip flop.

S	Qntl	J	K
0	0	0	X
0	01	1	X
1	0	X	1
1	1	X	0

Step 4:			Present			next			output		
$Q_A$	$Q_B$	$Q_C$	$Q_{A+1}$	$Q_{B+1}$	$C_{B+1}$	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
0	0	0	1	0	1	x	0	0	x	1	x
1	0	1	1	1	0	x	0	1	xx	1	
1	1	0	1	1	1	x	0	x	0	1	x
1	1	1	0	0	0	x	1	x	1	x	1

Step 5: obtain 6 Eqs from kmap for.

$$T_A = Q_B Q_C$$

$$K_A = 1$$

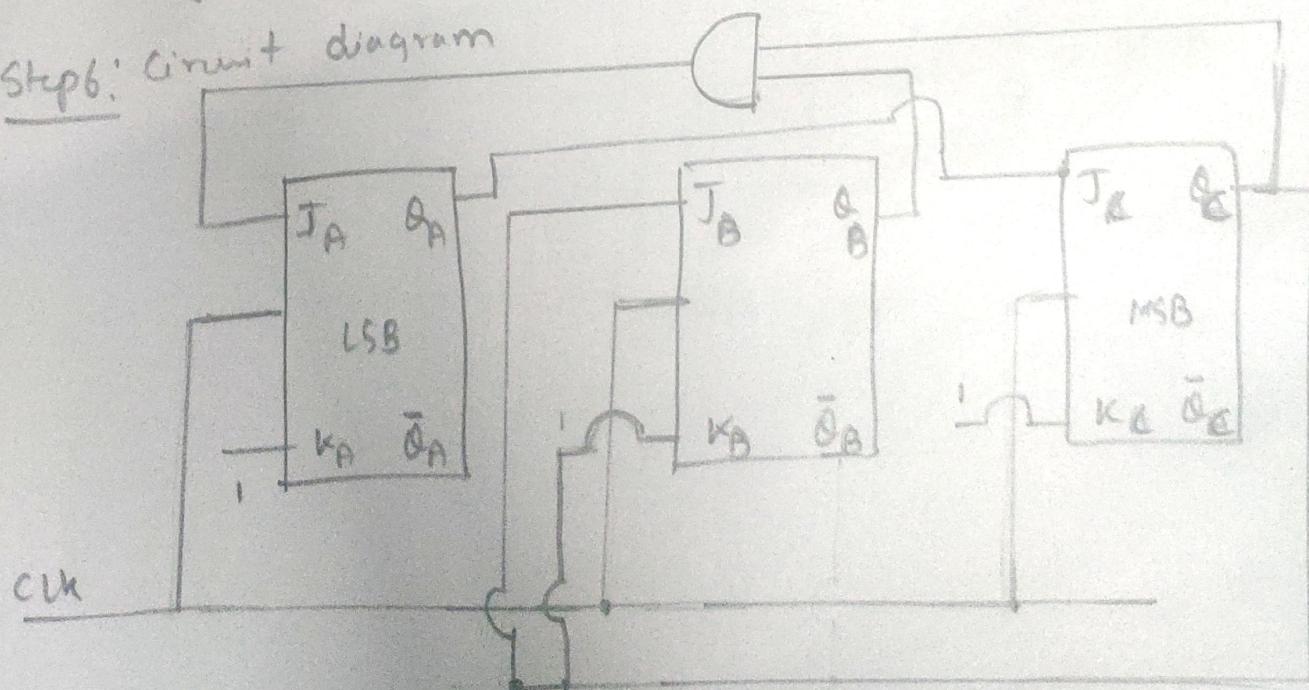
$$T_B = Q_C$$

$$K_B = Q_C$$

$$T_C = Q_A$$

$$K_C = 1$$

Step 6: Circuit diagram



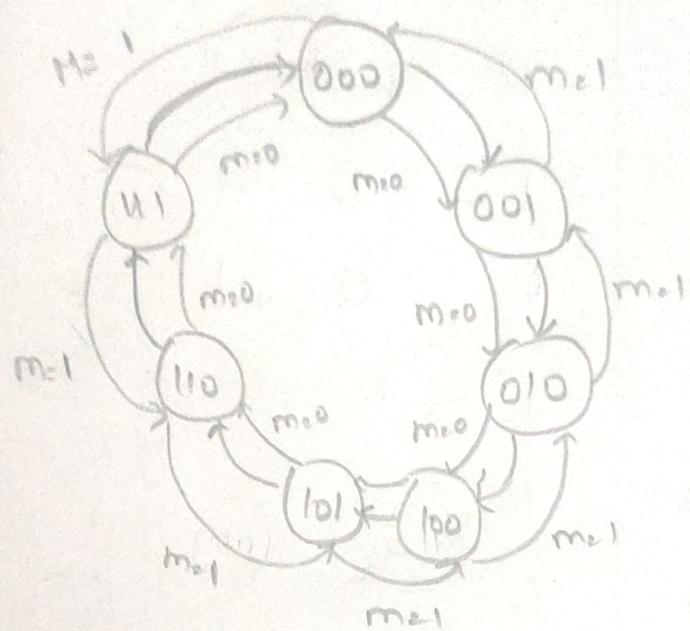
\* Synchronous 3bit - up / down Counter:-

own Mod-8 Synchronous. Using T flip flop.

Step1: Find No. of flip-flops required.  $N = 3$

Step2: State Diagram.

Where  $m$  = Controlling input



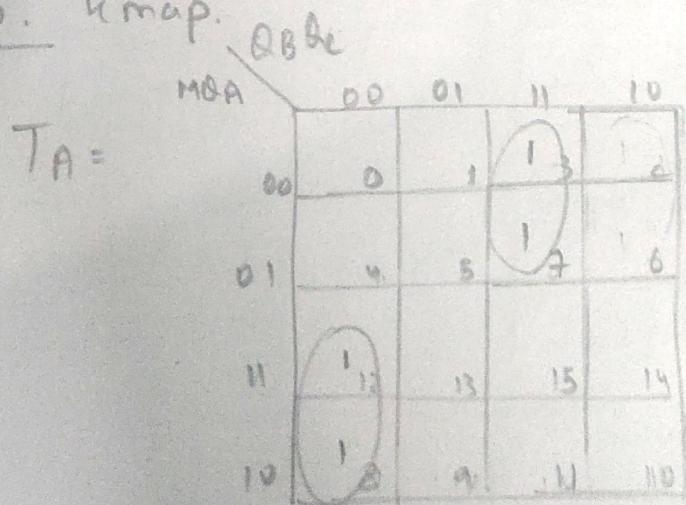
Step3: Excitation table T Flip-Flop.

$Q_n$	$Q_{n+1}$	T	$A_1$	$A_2$
0	0	0		
0	1	1		
1	0	1		
1	1	0		

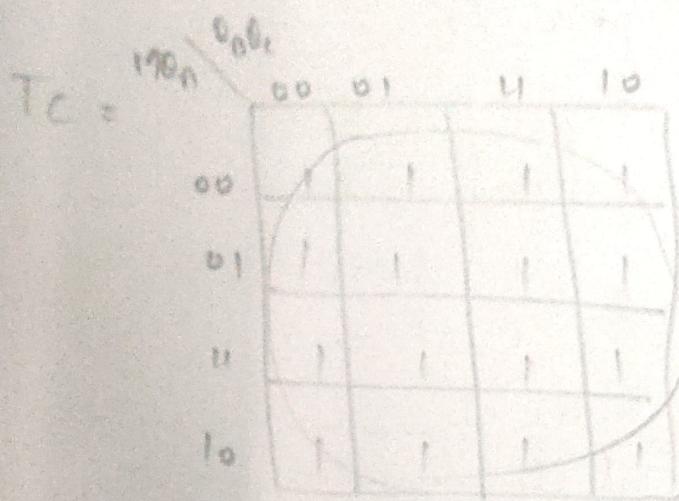
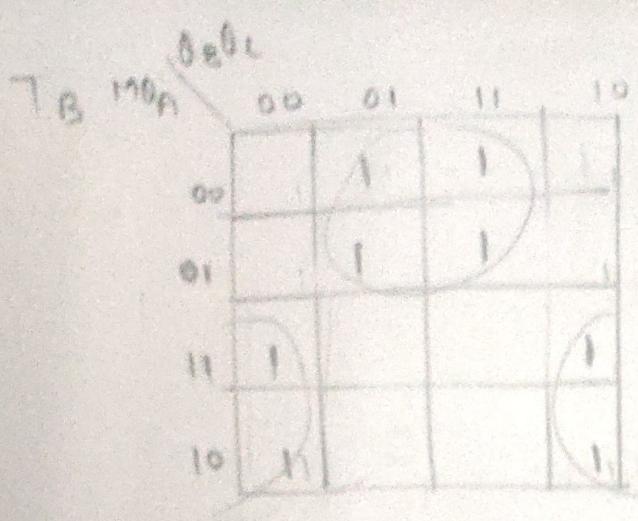
Step4: Counter table

Input M	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>A+1</sub>	Q <sub>B+1</sub>	Q <sub>C+1</sub>	T <sub>A</sub>	T <sub>B</sub>	T <sub>C</sub>
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	1	0	0	1	1	0	0	0
0	0	1	1	1	0	0	1	0	1
0	1	0	0	1	0	1	0	0	0
0	1	0	1	1	1	1	0	0	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1	0	1
1	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	1
1	0	1	1	0	1	1	0	0	0
1	1	0	0	0	0	1	0	1	1
1	1	0	1	1	1	0	0	0	1
1	1	1	0	1	1	0	0	0	0
1	1	1	1	1	1	1	0	0	0

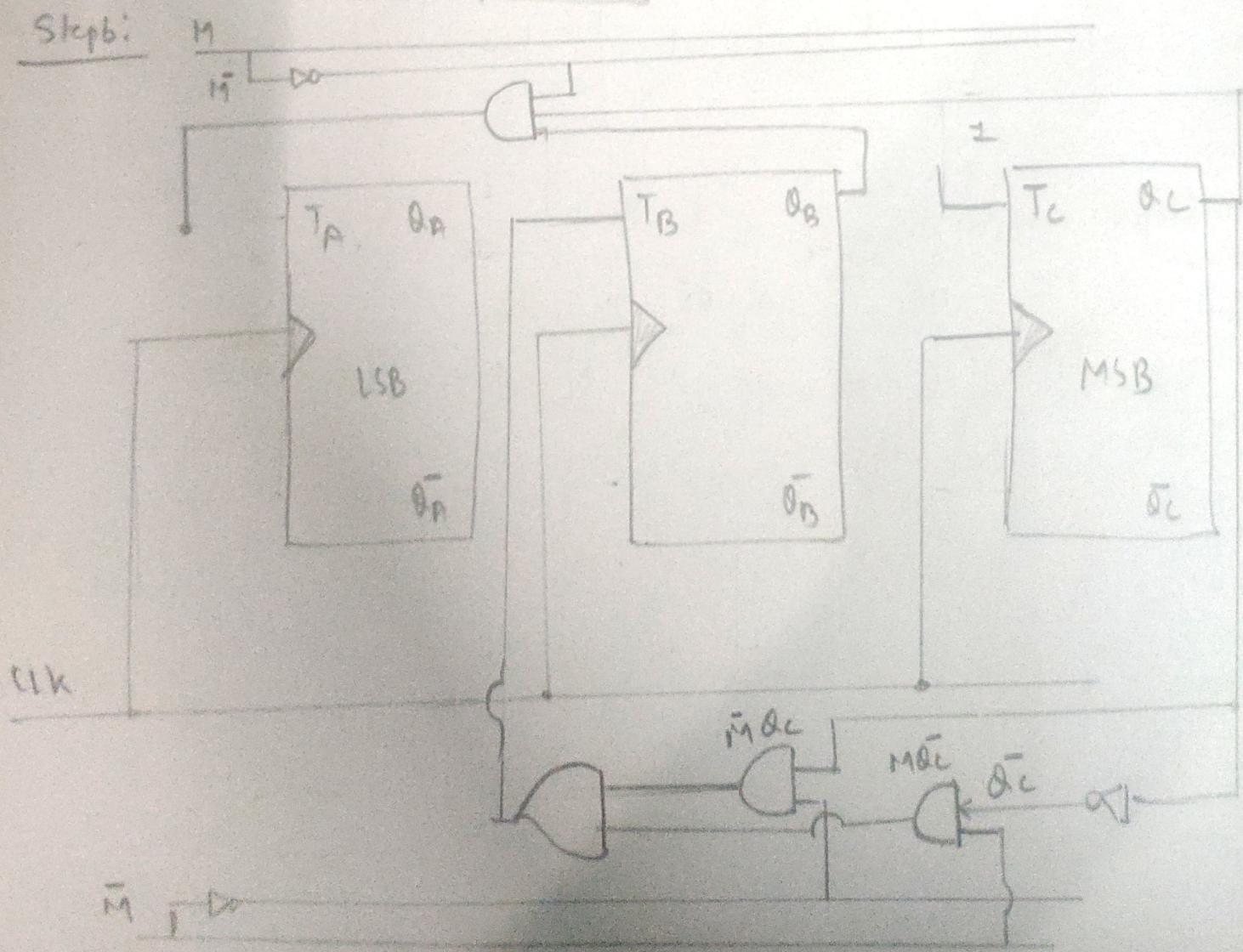
Step 5: K-map.

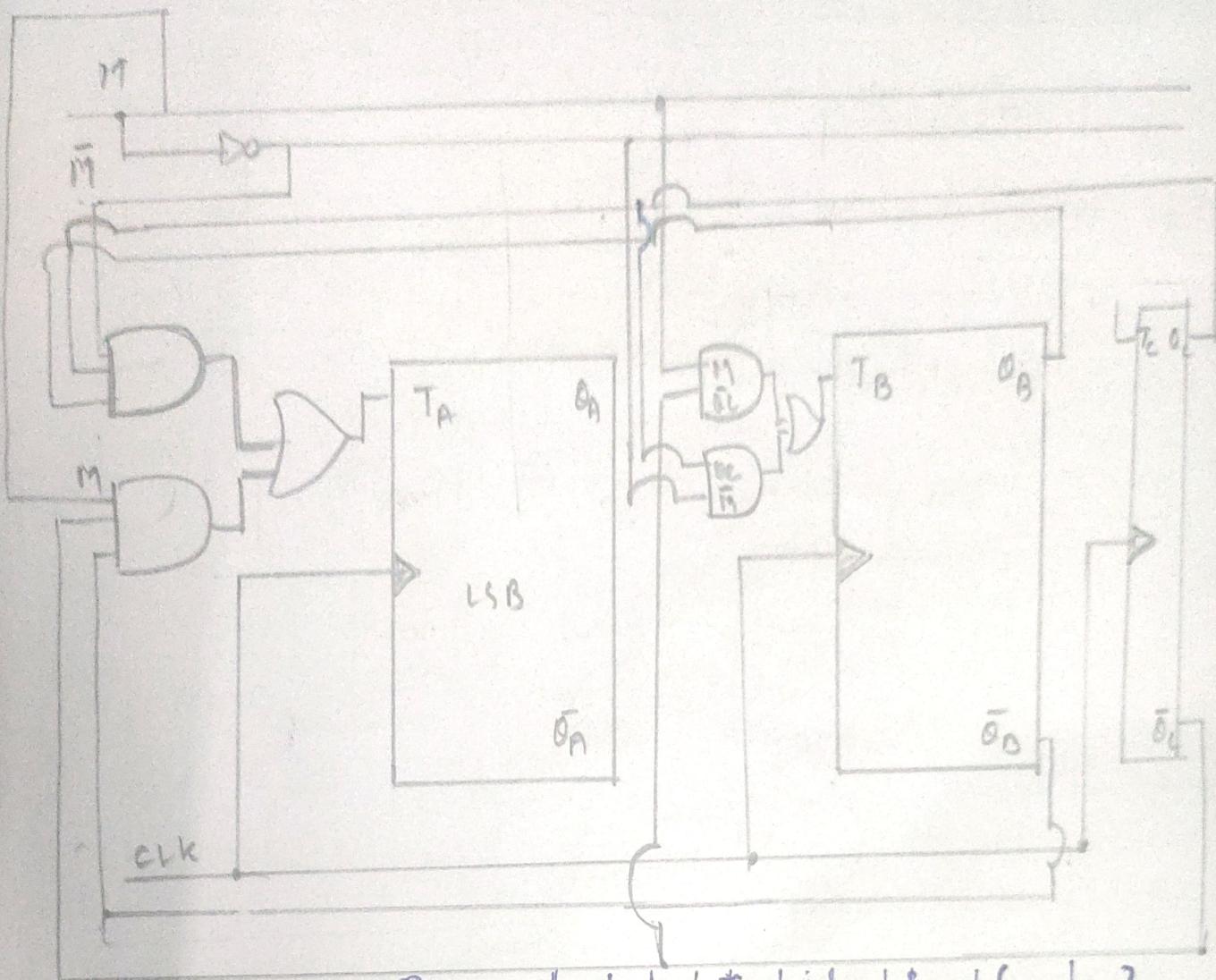


$$T_A = \bar{M} Q_B Q_C + M \bar{Q}_B \bar{Q}_C$$



$$T_C = 1$$





Ques

$$\begin{aligned}
 E &\rightarrow TE^1 \quad (1) \\
 E' &\rightarrow +TE'^1 / \epsilon \quad (2) \\
 T &\rightarrow FT^1 \quad (3) \\
 T' &\rightarrow *FT'^1 / \epsilon \quad (4) \\
 F &\rightarrow id / (CE) \quad (5)
 \end{aligned}$$

	1	+	*	id	+	*	C	L	?
E	-	-	-	1	1	1	1	1	-
E'	-	-	-	1	1	3	+ -	-	3
T	2	-	-	1	1	1	-	-	-
T'	1	-	-	1	1	1	-	-	-
F	8	5	-	1	1	6	-	-	6

NT

First

Follow

E

{id, C}

{\$, )}

E'

{+, ε}

{\$, )}

T

{id, C}

{+, \$, )}

T'

{\*, ε}

{+, \$, )}

F

{id, C}

{\*, +, \$, C}

## \* Mod-8 Asynchronous up-down counter (3bit)

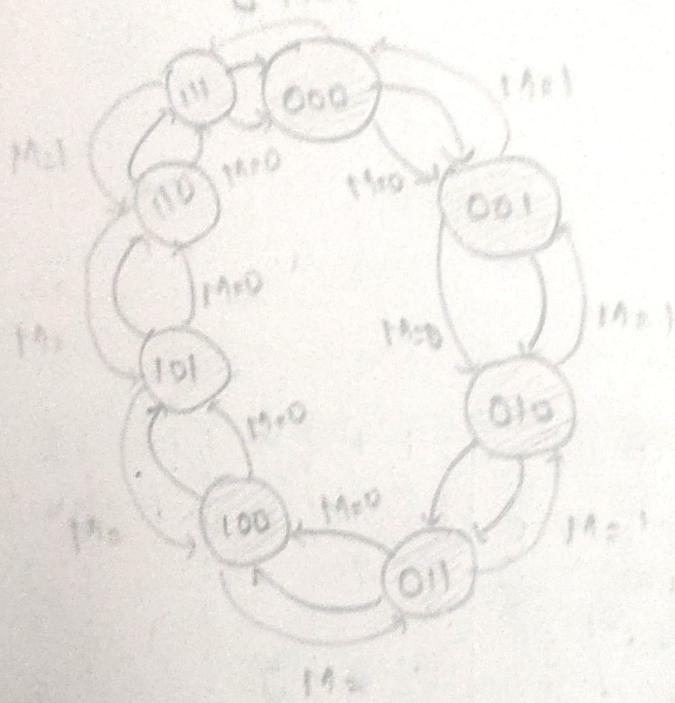
Ques Mod-8 Asynchronous counter.

Ans Step1  $n \geq N$

$$n = 3$$

No. of flip flops required: 3

Step2: Design state diagram for up and down counter using 3 bit

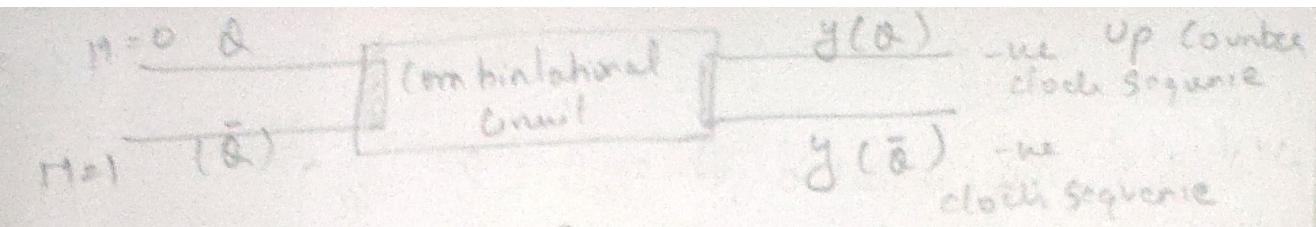


Step3: Asynchronous counter can be made using only J=K=1 if flip flops are used requires toggling action

Step4: for up counter Q must be connected with +ve clock edge ie.  $\rightarrow$  for down Q must connected with -ve clock edge ie.  $\overleftarrow{Q}$

for down Q must connected with -ve clock edge ie.  $\overleftarrow{Q}$

for combination of up & down counter we use both +ve and -ve



$M=0 \& \bar{Q}$

when  $m=0$

$M$	$Q$	$\bar{Q}$	$y$
0	0	0	0 $\rightarrow$ 0
0	0	1	0 $\rightarrow$ 1
0	1	0	1 $\rightarrow$ 2
0	1	1	1 $\rightarrow$ 3
1	0	0	0 $\rightarrow$ 0
1	0	1	1 $\rightarrow$ 1
1	1	0	0 $\rightarrow$ 1
1	1	1	1 $\rightarrow$ 2

→ using this circuit  
 input is flopped b/w  $Q$  and  $\bar{Q}$

$M=0 \& \bar{Q}$

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

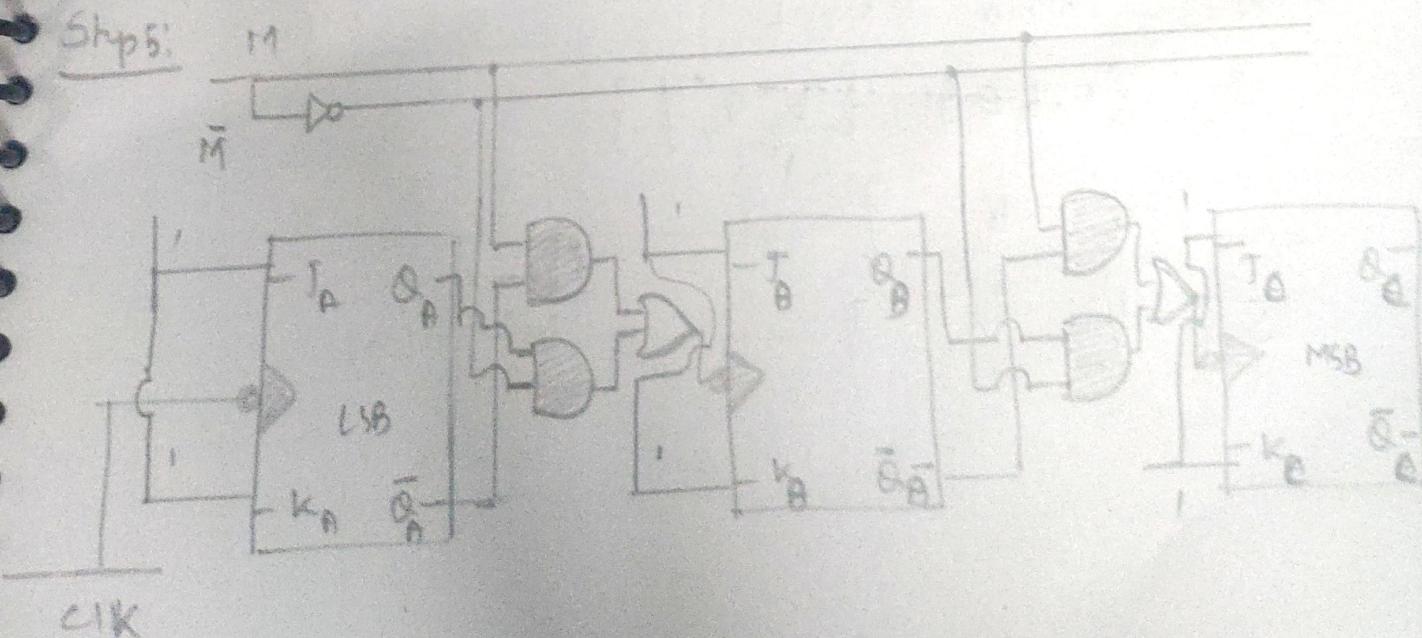
$$y = M\bar{Q} + \bar{M}Q$$

$$y = M \oplus Q$$

$\oplus$  - HD  
 $\ominus$  - NOR

$y = M \oplus Q$  circuit is used to switch the signal b/w  $Q$  &  $\bar{Q}$  for creating up and down asynchronous counter simultaneously.

Step 5:  $M$



Step-6 condition flipping the values.

Step-7 Truth table using flipping values condition.

→ Similar to  $\bar{Q}_0$  changes when five to -ve ( $\frac{1}{0}$ )

$Q_1$  changes when  $\bar{Q}_0$  changes ( $1 \rightarrow 0$ )

$Q_2$  changes when  $Q_1$  changes ( $1 \rightarrow 0$ )

$\bar{Q}_0$  changes when when  $\bar{Q}_0$  (-ve) is connected  
with -ve edge of clock  
( $0 \rightarrow 1$ )

Not given {  $Q_1$  changes when  $\bar{Q}_0$  remain same i.e. ( $0 \rightarrow 0$ )  
 $\bar{Q}_2$  changes when  $\bar{Q}_1$  remain same i.e. ( $0 \rightarrow 0$ )  
check from Sir last 20 pdf Notes.

clk       $Q_3$      $Q_2$      $Q_1$      $\bar{Q}_3$      $\bar{Q}_2$      $\bar{Q}_1$

Q<sub>2</sub>    $S \rightarrow a \underline{S} b S / b \underline{S} a S / \epsilon$    ③  
NT      Follow      First

$S \quad \{a, b, \epsilon\} \quad \{b, a, \$\}$

$\begin{array}{ccccccc} \dots & \frac{1}{1} & a & - & b & - & \$ \\ S & ; & 1/3 & 2/3 & 3 & - & \dots \end{array}$

Not in LL(0)